# DATA FUSION FOR SITUATION MONITORING, INCIDENT DETECTION, ALERT AND RESPONSE MANAGEMENT

**NATO Science Series**

A series presenting the results of scientific meetings supported under the NATO Science Programme.

The series is published by IOS Press and Springer Science and Business Media in conjunction with the NATO Public Diplomacy Division.

*Sub-Series*

| | |
|---|---|
| I. Life and Behavioural Sciences | IOS Press |
| II. Mathematics, Physics and Chemistry | Springer Science and Business Media |
| III. Computer and Systems Sciences | IOS Press |
| IV. Earth and Environmental Sciences | Springer Science and Business Media |
| V. Science and Technology Policy | IOS Press |

The NATO Science Series continues the series of books published formerly as the NATO ASI Series.

The NATO Science Programme offers support for collaboration in civil science between scientists of countries of the Euro-Atlantic Partnership Council. The types of scientific meeting generally supported are "Advanced Study Institutes" and "Advanced Research Workshops", although other types of meeting are supported from time to time. The NATO Science Series collects together the results of these meetings. The meetings are co-organized by scientists from NATO countries and scientists from NATO's Partner countries – countries of the CIS and Central and Eastern Europe.

**Advanced Study Institutes** are high-level tutorial courses offering in-depth study of latest advances in a field.
**Advanced Research Workshops** are expert meetings aimed at critical assessment of a field, and identification of directions for future action.

As a consequence of the restructuring of the NATO Science Programme in 1999, the NATO Science Series has been re-organized and there are currently five sub-series as noted above. Please consult the following web sites for information on previous volumes published in the series, as well as details of earlier sub-series:

http://www.nato.int/science
http://www.springeronline.nl
http://www.iospress.nl
http://www.wtv-books.de/nato_pco.htm

Series III: Computer and Systems Sciences – Vol. 198 ISSN: 1387-6694

# Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management

Edited by

Elisa Shahbazian

*Lockheed Martin Canada, Montreal, Quebec, Canada*

Galina Rogova

*Encompass Consulting, Rochester, New York, USA*

and

Pierre Valin

*Defense Research and Development Canada, Valcartier, Quebec, Canada*

IOS Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

Published in cooperation with NATO Public Diplomacy Division

Proceedings of the NATO Advanced Study Institute on Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management
Yerevan, Armenia
19–29 August 2003



ISBN 1-58603-536-3
Library of Congress Control Number: 2005932880

*Publisher*
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

*Distributor in the UK and Ireland*
Gazelle Books
Falcon House
Queen Square
Lancaster LA1 1RN
United Kingdom
fax: +44 1524 63232

*Distributor in the USA and Canada*
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com



PRINTED IN THE NETHERLANDS

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Preface

An Advanced Study Institute (ASI) "Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management" was held in Yerevan State University's Narek Hotel, 19–29 August 2003 in Tsakhkadzor. This ASI continued the exploration of the relatively young (less than 20 years) discipline called Data Fusion, subsequent to the Multisensor Data Fusion ASI held in Pitlochry, Scotland, UK, 25 Jun 2000 – 7 Jul 2000. This publication is the Proceedings of the Institute.

An ASI is a high-level tutorial activity, one of many types of funded group support mechanisms established by the NATO Science Committee in support of the dissemination of knowledge and the formation of international scientific contacts. The NATO Science Committee was approved at a meeting of the Heads of Government of the Alliance in December 1957, subsequent to the 1956 recommendation of "Three Wise Men" – Foreign Ministers Lange (Norway), Martino (Italy) and Pearson (Canada) on Non-Military Cooperation in NATO. The NATO Science Committee established the NATO Science Programme in 1958 to encourage and support scientific collaboration between individual scientists and to foster scientific development in its member states. In 1999, following the end of the Cold War, the Science Programme was transformed so that support is now devoted to collaboration between Partner-country and NATO-country scientists or to contributing towards research support in Partner countries. Since 2004, the Science Programme was further modified to focus exclusively on NATO Priority Research Topics (i.e. Defense Against Terrorism or Countering Other Threats to Security) and also preferably on a Partner country priority area.

Data Fusion is a very broad interdisciplinary technology domain. It provides techniques and methods for:

1. integrating information from multiple sources and using the complementarity of these sources to derive maximum information about the phenomenon being observed;
2. analyzing and deriving the meaning of this information and predicting possible consequences of the observed state of the environment;
3. selecting the best course of action; and
4. controlling the actions.

The Data Fusion ASI in Pitlochry provided a systematic high-level view of data fusion fundamental theory and the enabling technologies and presented a set of applications in an accessible manner. In that ASI, more emphasis was put on the first, more mature phase of data fusion, namely the detection and identification/classification of phenomena being observed and exploitation of the related methods for Security-Related Civil Science and Technology (SST) applications. The organizers felt that in this ASI it was necessary to:

- expand on the data fusion methodology pertinent to Situation Monitoring, Incident Detection, Alert and Response Management;
- discuss some related Cognitive Engineering and visualization issues;
- provide an insight into the architectures and methodologies for building a data fusion system;

- discuss fusion approaches to image exploitation with emphasis on security applications;
- discuss novel distributed tracking approaches as a necessary step of situation monitoring and incident detection;
- provide examples of real situations, in which data fusion can enhance incident detection, prevention and response capability.

The theme of the Institute was scientific communication and exchange of ideas among academic, industrial, and government laboratory groups having a common interest in the development of information fusion based approaches to detection and prevention of incidents.

The technical program was created to highlight general concepts (Fusion Methodology, Human Computer Interactions and Systems and Architectures) in the first week and applications (Data Fusion for Imagery, Tracking and Sensor Fusion and Applications and Opportunities for Fusion) in the second week, thus ensuring that the attendees were given a logical presentation of the data fusion material. In addition, at the request of some students as well as lecturer participants, a short tutorial on the Introduction to Data Fusion was repeated from the previous Data Fusion ASI in Pitlochry, on one of the evenings during the first week.

The Organizing Committee and the Directors encouraged informal discussion sessions, which were especially useful because of the interdisciplinary nature of the topics discussed as well as the fact that most students did not have a formal education in Data Fusion, since the discipline is not on the curricula of most Universities in the world.

Fifty-nine participants and lecturers representing Armenia, Belgium, Bulgaria, Canada, Czech Republic, France, Norway, Portugal, Israel, Italy, Spain, Russia, Ukraine, and the United States attended the Institute. A distinguished faculty of lecturers was assembled and the technical program was organized with the generous and very capable assistance of the Organizing Committee composed of Prof. Ashot Akhperjanian (Armenia), Dr. Elisa Shahbazian (Canada), Dr. Eloi Bosse (Canada), Dr. Galina Rogova, (USA), and Dr. Pierre Valin (Canada).

The benefits to be gained from any ASI depends on the faculty – the lecturers who devote so much of their time and talents to make an Institute successful. As the reader of these proceedings will see, this ASI was particularly honored with an exceptional group of lecturers to whom the organizers and participants offer their deep appreciation.

Due to the broad interdisciplinary nature of the subject, the editors of this volume were faced with difficult decisions, such as:

- considering the technical continuity more important than keeping lectures by the same author together, since some lecturers chose to talk on very disparate topics;
- accepting some redundancy of discussion between lectures, when the applications are different; and
- accepting some minor differing interpretations of the data fusion model and taxonomy, since these are still debatable topics in the data fusion community;
- choosing to include some lectures on applications related to Situation Monitoring, Incident Detection, Alert and Response Management that did not discuss fusion methodology but presented a great challenge for fusion applications.

We are grateful to a number of organizations for providing the financial assistance that made the Institute possible. Foremost to the NATO Security Through Science Programme, which provided not only important financial support for the Institute, but equally valuable organizational and moral support, both in the beginning, when the ASI was proposed, as well as during the institute, through the participation of the NATO observer Dr. Steinar Hoibraten from Norway. In addition, the following sources made significant contributions: the U.S. Office of Naval Research, Defense Research and Development Canada in Valcartier, Centre of Research in Mathematics of University of Montreal, Canada, and Lockheed Martin Canada.

We would like to thank the management of Yerevan State University for allocating the Narek hotel for the ASI and ensuring that all the requirements of the ASI were fulfilled, and the staff of Narek hotel for a truly enjoyable and memorable two weeks in Tsakhtadzor. We would like to thank the Yerevan Physics Institute for organizing an unforgettable picnic at their facility on the banks of the mountainous Lake Sevan. We would also like to thank our local site manager, Karine Gasparian, for her dedicated efforts to address all communication, housing, transportation and entertainment requirements of the ASI participants, so that the Organizing Committee was able to fully concentrate on the technical program issues. Our special thanks go to our administrative assistants and interpreters Anna Galstyan, Artsvik Gevorkian, Armen Malkhasyan, Gayane Malkhasyan and Alana Mark, whose competence and warm friendliness made all the attendees feel welcomed at the ASI and comfortable in Armenia.

A very special acknowledgement goes to Ani Shahbazian who undertook a very challenging task of English language editing of all the lecturers' manuscripts, reformatting all lectures after the technical editing was complete, and producing a camera-ready document for IOS Press. Thank you for your long hours and hard work.

And, finally, all of our thanks go to the people of Armenia, who certainly displayed, in every way, their warmth and hospitality.

Elisa Shahbazian
Montreal, Canada

Galina Rogova
Rochester, USA

Pierre Valin
Valcartier, Canada

December 2004

This page intentionally left blank

# Contents

Preface v
*Elisa Shahbazian, Galina Rogova and Pierre Valin*

**1. FUSION METHODOLOGY**

Information Fusion for Decision Making 3
*Erik Blasch*

A Gentle Introduction to Fusion of Multiple Pattern Classifiers 23
*Fabio Roli*

Uncertainty Management for Intelligence Analysis 35
*Ronald R. Yager*

Knowledge, Uncertainty and Belief in Information Fusion and Situation Analysis 61
*Éloi Bossé, Anne-Laure Jousselme and Patrick Maupin*

Information Evaluation: A Formalisation of Operational Recommendations 81
*Vincent Nimier*

Fusion Techniques for Airborne and Spaceborne Identification of Objects 91
*Pierre Valin*

Distributed Fusion: Learning in Multi-Agent Systems for Time Critical Decision Making 104
*Galina L. Rogova, Peter Scott and Carlos Lollett*

Active Robotic Sensing as Decision Making with Statistical Methods 129
*Lyudmila Mihaylova, Tine Lefebvre, Herman Bruyninckx and Joris De Schutter*

A New Genetic Algorithm for Global Optimization of Resources in Naval Warfare 143
*David Boily and Hannah Michalska*

Calculus of Variations and Data Fusion 171
*Paolo Corna, Lorella Fatone and Francesco Zirilli*

Reliability in Multiple Hypotheses Testing and Identification Problems 189
*Evgueni Haroutunian*

**2. HUMAN COMPUTER INTERACTION**

Decision Support in Command and Control 205
*Éloi Bossé, Stéphane Paradis and Richard Breton*

Multimodal Input Fusion in Human-Computer Interaction 223
*Andrea Corradini, Manish Mehta, Niels Ole Bernsen, Jean-Claude Martin and Sarkis Abrilian*

Spatio-Temporal Data Visualization and Analysis for Multi-Target Tracking 235
*Kiril Alexiev*

**3. SYSTEMS AND ARCHITECTURES**

Principles of Systems Engineering for Data Fusion Systems 255
*Alan Steinberg*

A Taxonomy of Sensor Processing Architectures 265
*Shivakumar Sastry and S.S. Iyengar*

Knowledge Fusion in the Scalable Infosphere 283
*Alexander Smirnov*

Multi-Agent Data and Information Fusion 308
*Vladimir Gorodetsky, Oleg Karsaev and Vladimir Samoilov*

Architecture Analysis and Demonstration of Distributed Data Fusion 340
*Elisa Shahbazian, Louise Baril and Guy Michaud*

Data Fusion Testbed for Recognized Maritime Picture 353
*Eric Lefebvre*

Comparisons of Track-Level Fusion Results with Tracklets Application Within a Simulation Environment 366
*Eric Ménard, Daniel Turgeon, Guy Michaud and Elisa Shahbazian*

**4. DATA FUSION FOR IMAGERY**

Multisensors and Contextual Information Fusion 381
*Yannick Allard and Alexandre Jouan*

Fusion of Two Imagery Classifiers 394
*Hugues Demers*

Application of Multi-Dimensional Discrete Transforms on Lie Groups for Image Processing 404
*Ashot Akhperjanian, Armen Atoyan, Jiri Patera and Vardan Sahakian*

Application of Continuous Extension of DCT to FLIR Images 417
*Armen Atoyan and Jiri Patera*

Neural Network-Based Fusion of Image and Non-Image Data for Surveillance and Tracking Applications 426
*Malur K. Sundareshan and Yee Chin Wong*

Super-Resolution of Tactical Surveillance and Tracking Data for Fusion of Images 448
*Malur K. Sundareshan and Supratik Bhatacharjee*

Design of Fusion Architectures for Surveillance and Tracking Using Information Value Mappings 464
*Malur K. Sundareshan and Timothy J. Peterson*

Intelligent Fusion of Visual, Radio and Heterogeneous Embedded Sensor Information Within Cooperative and Distributed Smart Spaces 480
*Carlo S. Regazzoni, Reetu Singh and Stefano Piva*

Multimodal Cooperative Modulation Estimation and Terminal Location for Multisource Sensor Networks 498
*Marco Guainazzo, Matteo Gandetto, Maristella Musso, Carlo S. Regazzoni and Pramod K. Varshney*

Issues in Multicamera Dynamic Metadata Information Extraction and Interpretation for Ambient Intelligence 520
*Luca Marchesotti, Giuliano Scotti and Carlo Regazzoni*

An Expert System for Surveillance Picture Understanding 542
*Helman Stern, Uri Kartoun and Armin Shmilovici*

**5. TRACKING AND SENSOR FUSION**

Group Tracking 555
*Erik Blasch*

Robust Modification of the EM-Algorithm for Parametric Multitrajectory Estimation in Noise and Clutter 575
*Mikhail B. Malyutov and M. Lu*

Non-Parametric Multi-Trajectory Estimation 587
*M.B. Malyutov, Alexandre B. Tsybakov and Ion Grama*

Integrated Estimation and Guidance for Target Interception 597
*Dany Dionne and Hannah Michalska*

An Adaptive, Variable Structure, Multiple Model Estimator 613
*Hannah Michalska and Dany Dionne*

A Distributed Sensor Network for Real-Time Multisensor Surveillance Against Possible Terrorist Actions 627
*Gian Luca Foresti and Lauro Snidaro*

Advantages and Drawbacks of Multisite Radar Systems 640
*Victor S. Chernyak*

**6. APPLICATIONS AND OPPORTUNITIES FOR FUSION**

Monitoring Search and Rescue Operations in Large-Scale Disasters 659
*Alessandro Farinelli, Luca Iocchi and Daniele Nardi*

Fusion of Various Methods for Resolving the Shakespeare Controversy 671
*Mikhail B. Malyutov*

A Game Model for Effective Counteraction Against Computer Attacks in Intrusion Detection Systems 685
*Edward Pogossian and Arsen Javadyan*

The Use of G-Networks Models for the Assessment of Computer Networks Performance 708
*Khanik Kerobyan, Khatchik Vardanyan and N. Kostanyan*

Resource Sharing Performance Models for Estimating Internet Service Characteristic in View of Restrictions on Buffer Capacity 729
*Khanik Kerobyan and Khatchik Vardanyan*

Considerations of Possible Security-Related Nuclear Power Plants Incidents in Romania 740
*Bogdan Constantinescu*

Blast Impact Assessment in the Iron Ore Mining Region of Ukraine 749
*Mykola Kharytonov, Alexandr Zberovsky and Andriy Babiy*

Information and Spatial Data Processing in Risk Management 760
*Urbano Fra Paleo*

Forest Fire Detection by Infrared Data Processing 774
*Luis Vergara and Ignacio Bosch*

**APPENDICES**

Appendix A: Organizing Committee 787

Appendix B: Lecturers and Authors 788

Subject Index 797

Author Index 819

# 1. FUSION METHODOLOGY

This page intentionally left blank

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Information Fusion for Decision Making

## *Designing Realizable Information Fusion Systems*

Erik BLASCH
*Air Force Research Laboratory, Dayton, Ohio 45433*

**Abstract.** Humans explore their world by selecting sensor observations, fusing data, and deciding how to act. For the most part, the goal of sensory data fusion is to increase confidence in entity identification and location. Since the 4th century B.C., the fusion process has been acknowledged through sensed data association. However, in the past century, probability and statistics formalized the association problem with mathematical correlation. Together, association and correlation aid in reducing uncertainty for decision making. Fusion can have a variety of meanings, such as data fusion, sensor fusion, and information fusion. Thus, it is important to explore terminology. Data fusion is the correlation of raw data whereas sensor fusion is the multimodal integration of transduced data into a common perception. Information fusion (IF) is the collection of sensory and knowledge data to create a unique understanding upon which a user performs response management. IF for decision making includes: the situational context, sensor control, and the user-machine interaction. In this chapter, we will highlight many of the key aspects of fusion research to give the reader a flavor of what IF can do and cannot do to aid a user. This chapter will discuss the fundamentals of information fusion: (1) who: decision makers and machines; (2) what: data, sensor, and information; (3) where: applications; (4) when: appropriate situations; (5) why: to reduce uncertainty and increase confidence; and (6) how: techniques. What you gain from reading this chapter is a perspective of how to integrate the fusion-machine and the cognitive-user, a taxonomy of information fusion algorithms, and an appreciation of the benefits and limitations of fusion designs. Sensor and data fusion books are useful for the reader who is unfamiliar with basic concepts. This chapter is not intended to repeat the explorations of these texts, but to summarize important aspects of IF relative to "cognitive fusion" which integrates the user with the fusion machine for decision making.

**Keywords.** Fusion, decision making, user-fusion, tracking, cognitive fusion

### 1. Introduction

There are many ways to describe information fusion (IF), such as the combination of data, algorithms, and knowledge representations. Many books [2,19,22,31,32,40,63] explain techniques to combine physical data to reduce uncertainty. However, the main concept that is typically overlooked in previous books, papers, and explorations of fusion is that the purpose for a fusion system is to help the user do his job [15]. The human has a defined task to accomplish and would like to utilize a machine to (a) extend their sensory capabilities, and (b) to reduce data complexity and uncertainty. Thus, we will use the term "*user*" to represent the human and his associated task. Each user operates with contextual information and has a mission to accomplish, such as a physician performing a diagnosis, an economist looking at the financial health of an organization,

or a security specialist looking for intruders. In each case, the user does not care what algorithms or data is combined to produce information, but assesses fusion performance based on whether it helps him do his job. Thus, data, sensor, and information fusion are employed for human decision making and represents the models, processes, and techniques used to combine data to enhance knowledge understanding.

In the rest of the chapter, we will explore the basic concepts of fusion from the standpoint of the user requirements for a decision-making operation. If the reader keeps in mind his own application in developing "fusion" systems, he will better appreciate the need, benefits, and limitations of fusion systems that help him do his task. The concepts developed in the chapter are related to the new perspective of viewing a "fusion system" as a tool. Many references related to IF point the reader to a large body of literature that utilizes the concepts of fusion systems with such applications as robotics [1], military [16], economics [9], medical [20], and industrial [30]. In 1998, application and theoretical fusionists formed a society, "The International Society of Information Fusion," dedicated to fusion methods of which the chapter author was a founding member [33, www.inforfusion.org].

This chapter addresses ontological concerns for data fusion with a bias towards the highest level of data fusion – User Refinement. Fundamentals of IF include: terminology, history, models, techniques, and capabilities. Section 2 briefly discusses fusion terminology and historical developments. Section 3 presents models and Section 4 lists appropriate techniques at each level of the User-Fusion model. Section 5 tabulates the advantages and disadvantages of fusion as it pertains to User Refinement. Section 6 discusses User Refinement and Section 7 draws conclusions.

## 2. What Is Fusion?

The idea of fusion has been around since the beginning of man. For instance, Aristotle used the word fusion to mean the combination of sensory data (i.e. two visual images) [12]. However, fusion might also mean integration (e.g., of visual and auditory data). *Integration* is the combination of information, whereas *fusion* is the combination of information that produces something that is new or different than the data representation used to create the information.

- fusion: "merging of different elements into a union" [64]
- integration: "make whole by bringing the parts together – unify."

The word fusion has been used in many domains, such as computer science, physics, and psychology. IF researchers leverage these three domains; however, there are cases which do not pertain. For instance, *nuclear fusion* is the combination of nucleus particles such that the neutrons and protons of one nucleus combine with another to make a more massive nucleus. Also, *medical fusion* can be used to describe bone growth where the bone is fused – meaning that the cells grow together. These examples demonstrate that we are interested in "information fusion" and not just "fusion."

Fusion is the process "to make whole," and there are many antonyms:

- con-Fusion: unclear, assemble without order, disturbance;
- dif-Fusion: spread about, scatter (like entropy), random thermal motion;
- ef-Fusion: the pouring forth of gas or fluid, or brought into a cavity;
- pro-Fusion: copious, abundance, outpour, too much [64].

There are various types of fusion, which relate to whether the machine or user processes the facts. Machine Fusion is the computer integration of data from multiple sources to achieve a unified, complete, and accurate representation about an entity. Transforming data to give it meaning results in information. For example, signal processing of sensor measurements for object identification results in information [22]. Data Fusion is a dynamic process involving the association, correlation, and combination of data from multiple sources resulting in a fused product, which is more complete and accurate than any of the separate data elements [61]. Information Fusion is an adaptive multiple level process involving aggregating, data mining, and integration of information from multiple sources resulting in a fused information product of greater value than any of its parts.

Cognitive fusion is a user process for integrating data and combining context to produce knowledge, which ultimately affords user decision making [8]. It can be referred to as grey-matter fusion. Humans cannot reason over a plethora of data, so data is fused into a refined set of information necessary for reasoning. Cognitive fusion can be separated into many constructs, but two that are necessary for human decision making include *sensor* and *data fusion*. Human sensors process the world in analog signals, which are converted to a neuronal all-or-none response. Sensing over different physical entities includes visual and audio signals, as well as non-human sensors including radar, x-ray, and ultrasonics [39] that are presented to the user. Data fusion for human consumption includes text, numbers, pixels, etc. To help a human make decisions, fusionists must answer IF questions such as choice of sensors, mathematical models, input data, and output representations to afford decision making.

Formally, IF integrates the machine and cognitive fusion functions:

> "Information fusion is a formal mathematics, techniques, and processes to combine, integrate, and align data originating from different sources through association and correlation for decision making. The construct entails obtaining greater quality, decreased dimensionality, and reduced uncertainty for the user's application of interest."

- data Fusion: "Combining & organizing numerical entities for analysis;"
- sensor Fusion: "Combining devices responding to a stimulus;"
- information Fusion: "Combining data to create knowledge;"
- display Fusion: "Presenting data simultaneously to support human operations;"
- cognitive Fusion: "Integrating information in the user's mental model."

Display fusion is an example of integrating cognitive and machine fusion. Users are required to associate machine correlated data from individual systems displayed on an interface. Figure 1 shows a team of people coordinating decisions derived from fusion displays. Since we desire to utilize IF where appropriate; we assume that combining different sets of data from multiple sources would allow for a performance gain (i.e. increased quality and reduced uncertainty). Uncertainty about world information can be reduced through systematic and appropriate combinations of data. Since the world is complex (based on lots of information, users, and machines) we desire timely responses of extended dimensionality over different operating conditions [57]. The computer or fusion system is adept at number crunching while the user can reason over displayed information, so the effective fusion of physical, process, and cognitive models is needed. The strategy of "how to fuse" requires pragmatic qualities of effective

and efficient methods to *decompose* user information needs into signal, feature, and decision fusion tasks [27]. The decomposition requires prioritizing data combinations to employ a knowledge gain. The decomposition of tasks requires a fusion model or architecture, which is developed from user information needs. Information needs are based on user requirements that include object information and fusion performance expectations. To control fusion performance, sensor management directs machine or database resources based on observations, assessments, and mission constraints, which can be accepted, adapted, or changed by the user. Today, with a variety of users and many ways to fuse information, standards are evolving for effective and efficient fusion techniques.

**Figure 1.** Information fusion of displays.

### *2.1. Information Fusion Research*

*It is not what researchers say Information Fusion (IF) is, it is what they are actually doing*. There are many applications of IF including (a) tracking and target recognition for defense systems [6], traffic control [3], and site monitoring [44]; (b) remote sensing applications for agriculture, geoscience [17], and environmental monitoring [17]; (c) intelligent systems such as robotics [1], data bases, and web pages; (d) health diagnosis medical systems; and (e) decision tools such as cockpit displays [26]. In each of these cases, the decision desired for system effectiveness includes *dimensionality reduction* (reduce amount of information), *precision and certainty* (increase confidence in results), and *robustness* (correct for errors). Examples of other fusion applications are:

*Financial* – Stock prices
*Robotics* – Encoder data
*Maintenance* – Material properties
*Remote Sensing* – Forest Content
*Medical Diagnostics* – MRI information
*Biological* – Audio and visual data
*Military Radar* – Signal return
*Military* – Location of forces
*Environmental Information* – Weather data
*Sensor Diagnostics* – Motor vibration

Research areas concerned with cognitive fusion (CF) include: 1) **Military** (e.g., UAVs) [11]; **Aviation** (ATC, Pilots) [25]; **Civil** – Fire, Weather Surveillance, Decision Aiding, Response; **Medicine** – Diagnosis, Surgery, Prostheses; **Industrial** – Training, Repairs [30]; **Educational** – Distance Education, Disability Support [7]; and **Robotics** [40]. CF is based on two perspectives: a) micro, which is tactical, and b) macro, which

can be theater or strategic [15]. Micro systems include one user, while macro systems could be a team of people with a common representation of data. Both levels of CF are critical in the design and process of achieving IF systems where the supervisory control by a human-in-the-loop (HIL) needs decision-making information [9]. Fusion models must incorporate workload, attention, and trust constraints to aid cognitive fusion, design and implementation [15].

To process data for CF, IF systems employ Mathematical Tools. The mathematical tools typically used in IF designs include probability, evidential theory, and set theory. A probability analysis includes frequency interpretation [45]. An example is Bayesian mathematics which assumes known *a priori* information and an exhaustive data set that supports a single answer. Evidence theory includes Dempster-Shafer, DSmT [24], pignistic probability [56], and confidence factor methods which handle conflicting information; however, the set size can grow quickly for large problems. A third type of tool includes: Fuzzy Set, Possibility, Expert System, Random Sets [60], or Neural Networks [29,47]. Each of the set techniques weighs the belief of the data in proportion to the other possibilities; however, solutions are brittle to a specific situation.

### *2.2. Historical Developments in Fusion*

Since the Greeks, man has explored many techniques of knowledge abstraction. For the fusionist, we are concerned with developments that extend sensing and utilize mathematics to formalize a knowledge gain. Some historical developments utilized by fusionists include:

- Plato: Use of Propositions, Beliefs, Evidence, through proofs;
- Aristotle: Tri-partate soul: 1) nutritive, 2) sensitive, & 3) rational with a Common sense;
- 1500's: Descartes. Mind and Body. Body integrates matter and motion;
- 1600's: Kepler (motion), Galileo (material properties), Newton (dynamic interaction);
- 1700's: Bayes, belief propagation of a finite set of possibilities;
- 1800's: James (Psychologist), experimental methods to study the integration of senses;
- 1960: Kalman, propagating observations over time by filtering, estimation and prediction;
- 1990: Waltz, Hall, first set of books dedicated to sensor and data fusion techniques [63].

Human reasoning is information association, which can be modeled mathematically. Mathematical knowledge abstraction and information simulation association culminates in decision making theory. Human decision making is the highest taxonomy of IF systems. The major paradigms established since the beginning of man's existence follow observation processing to seek knowledge. For example, Fitts and Posner [28] modeled the human motor functions, while Rasmussen [46] modeled the tasks humans perform. Some key taxonomies for seeking purposeful knowledge include:

**Table 1.** The reasoning hierarchy

| process | B.C. | 1950's | 1980's | 1990's | fusion function |
|---|---|---|---|---|---|
| survival | nutritive | automatic | skills | l 0 – 1 | correlation/est. |
| observation | sensitive | associative | rules | l 2 – 4 | association |
| thought | rational | cognitive | knowledge | level 5 | reasoning |
| | Aristotle | Fitts & Posner | Rasmussen | Blasch | |

IF exists in many forms, such as integrating audio and visual data, plans, or surveillance requirements. It is important to remember that the goal of any IF system is to extend user sensing capabilities and increase decision confidence. Models help illustrate the functions of effective data input, observation fusion, and information output for human decision making and action.

## 3. Information Fusion Models

There are three types of models: (1) physical, (2) mathematical, and (3) process. A *Physical Model* is a model whose parameters resemble the object characteristics being described and is empirically derived. Physical models include the object geometries, dynamics, and composition. A *Mathematical Model* is a symbolic model whose values are expressed in causal symbols and relationships. An example of a mathematical model is the quantification of the physical system's parameters as they change and can be used to support simulation studies. Typical fusion documents refer only to physical and mathematical models. What is needed is a strategy to employ the correct model at the correct time, which can only be achieved with a decision maker. *Process models* are designed to replicate steps in a sequence of events. Process models define system work flows, machine dynamics, and user functions for a given task. In the next section, we explore IF process models which utilize physical and mathematical models for real-world tests.

### *3.1. Data Fusion Models*

Two fusion models that are machine-focused include the Data-Feature-Decision (DFD) [23] model and the revised Joint Directors of Labs (JDL) model [55], shown in Figure 2 and 3 respectively. Dasarathy introduced the DFD model to guide a machine to make decisions. The JDL model consists of five modules with a human computer interface. While the JDL model has been the standard for fusion processing discussions, it should be adapted to reflect the importance of human information exploitation. Information exploitation includes control, decision, and action. In the JDL model, Level 4 process refinement might include both the user and sensor control functions as feedback for refining the fusion process. However, this is the result of display fusion and not cognitive fusion.

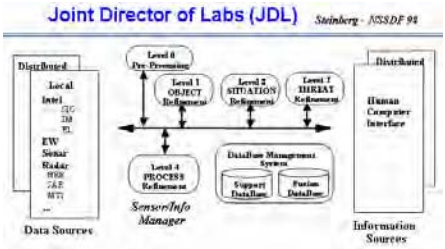



**Figure 2.** DFD model.

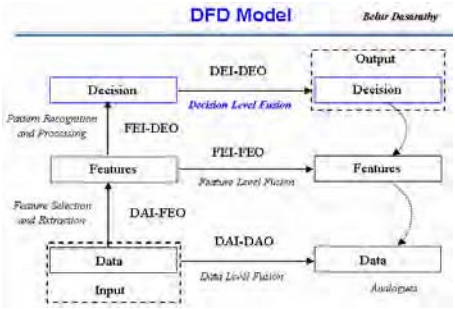



**Figure 3.** JDL revised model.

The Omnibus model [5], shown in Figure 4, is an extension of the observe-orient-decide-act OODA control loop (sometimes referred to as the Boyd control loop). The Omnibus model describes machine fusion based on human reasoning. Machine fusion models traditionally follow human reasoning because engineers have designed IF systems based on the way they think. Thus, the DFD, JDL, and Omnibus models all mimic human reasoning.

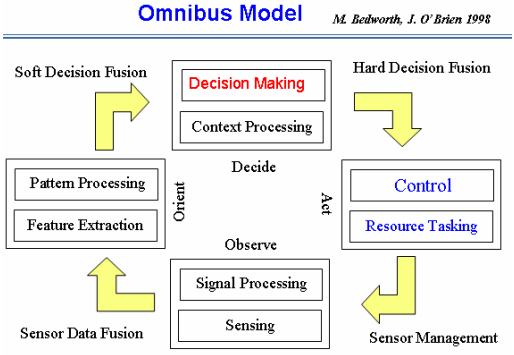



**Figure 4.** Omnibus model.

### *3.2. User Models*

Human in the Loop (HIL) models of semi-automated systems seek to provide the user with adequate situation awareness (SA). A pioneer and continued leader in the SA literature, Mica Endsley, stated that "Situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" [26]. This now-classic model translates into 3 levels:

- Level 1 SA: Perception of elements in the environment.
- Level 2 SA: Comprehension of the current situation.
- Level 3 SA: Projection of future states.

Operators of dynamic systems use their SA in determining their actions. To optimize decision making, the SA provided by a machine should be as precise as possible

about objects in the environment (Level 1 SA). A SA approach should present a fused representation of the data (Level 2 SA) and provide support for the operator's projection needs (Level 3 SA) in order to facilitate the operator's goals. From the SA model presented in Figure 5, attention and workload are key elements that affect not only SA, but also the user's decision-making and action abilities.

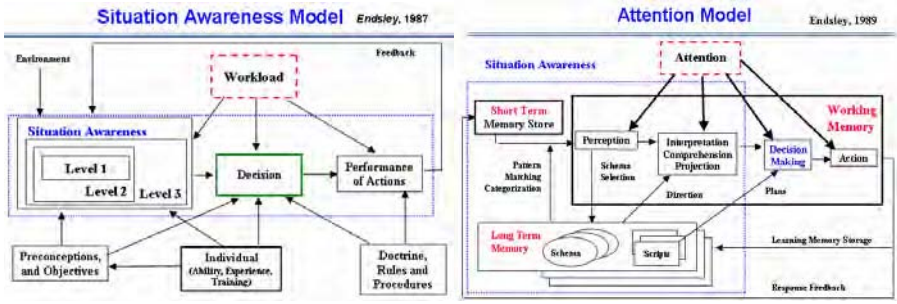



**Figure 5.** Endsley's Attention and Situation Awareness Models [25,26].

Both machine and user models can be utilized together with an integration of the top-down (user) and the bottom-up approaches (machine). The model that captures the simultaneous fusion processes is the User-Fusion model, shown in Figure 6. The key to the user-fusion process is data association, metric comparison, and process control. Users associate information to gain knowledge, while machines associate data to produce information. The difficulties of using only machine data association are that sensor data can be unreliable and the transfer process may corrupt data. User fusion may infer inappropriate data relations due to limited memory, inadequate training, and be limited in sensing (i.e. outside line-of-sight). By assessing the data (level 0 and 1) with information (level 2 and 3), process and user refinement (level 4 and 5) can compensate for these machine-user errors.

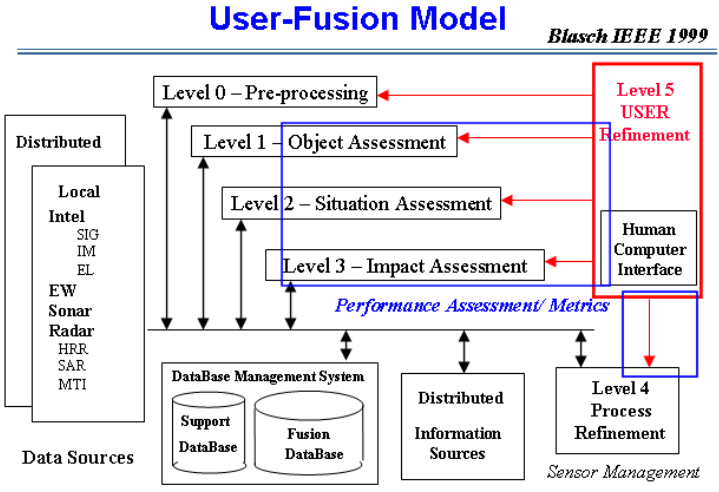



**Figure 6.** User-Fusion model [15].

The User-Fusion model activities are:

- Level 0: Sub-Object Data Assessment: estimation and prediction of signal/object observable states on the basis of pixel/signal level data association (e.g. information systems collections);
- Level 1: Object Assessment: estimation and prediction of entity states on the basis of observation-to-track association, continuous state estimation and discrete state estimation (e.g. data processing);
- Level 2: Situation Assessment: estimation and prediction of relations among entities, to include force structure and force relations, communications, etc. (e.g. information processing, FDP, FL);
- Level 3: Impact Assessment: estimation and prediction of effects on situations of planned or estimated actions by the participants; to include interactions between action plans of multiple players (e.g. assessing threat actions to planned actions and mission requirements, DM, PE);
- Level 4: Process Refinement (an element of Resource Management): adaptive data acquisition and processing to support mission objectives (e.g. sensor management and information systems dissemination, IO, $C^2$);
- Level 5: User Refinement (an element of Knowledge Management): adaptive determination of who queries information and who has access to information (e.g. information operation), and adaptive data retrieved and displayed to support cognitive decision making and action (e.g. altering the sensor display).

*Process refinement* of the machine controls data flow. *User refinement* guides information collection, region of coverage, situation and context assessment as well as process refinement of sensor selections. While each level needs fused information from lower levels, the refinement of fused information is a function of the user. Two important issues are *control* of the information and the *planning* of actions. There are many social issues related to centralized and distributed control such as team management and individual survivability. For example, a display, shown in Figure 1, might be globally distributed, but centralized for a single commander for local operations. Thus, the information displayed on a human computer interface should not only reflect the information received from lower levels, but also afford analysis and distribution of commander actions, directives, and missions. Once actions are taken, new observable data should be presented as operational feedback. Likewise, the analysis should include the local and global operational changes and the confidence of new information. Finally, execution should include updates to the distributed system of orders, plans, and actions to people carrying out mission directives. Thus, process refinement is not the fusing of signature, image, track, and situational data, but that of *decision information fusion* (DEC-IF) for functionality.

**DEC-IF** is a globally and locally refined assessment of fused observational information. DEC-IF is a result on which a user can plan and act. Additionally, human data can be gathered from other people to determine what information to collect, analyze, and distribute to others. One way to facilitate the receipt, analysis, and distribution of actions is that of the OODA loop [54]. The OODA loop requires an interactive display for functional cognitive decision-making. In this case, the display of information should orient a user in determining the situation based on the observed information (i.e. SA). These actions should be assessed from the user's ability to deploy his assets against the resources, constraints, and opportunities of the environment.

Extended information gathering can be labeled as *action information fusion* (ACT-IF) because an assessment of possible/plausible actions can be considered. The goal of any fusion system is to provide the user with a set of refined information for functional action. Taken together, the user-fusion system is actually a *functional sensor*, whereas the traditional fusion model is just an observational sensor. The user refinement block not only is the determination of *who* wants the data and whether they can refine the information, but *how* they process the information for knowledge. Bottom-up processing of information can take the form of physical sensed quantities that support the higher-level functions. To include machine automation as well as IF, it is important to afford an interface with which the user can interact with the system (i.e. sensors). Specifically, the top-down user can query necessary information to plan and decide while the automated system can work in the background to update the changing environment. Finally, the functional fusion system is concerned with the entire situation while a user is typically interested in only a subsection of the environment. The computer must try many combinations, but the human is adept at cognitive fusion and is sensitive to the changes in situational context.

Entity identity (ID) and localization information is the desired result of either real or perceived data from a time and space representation. The difficulty with presenting the complete set of real data is the sheer amount available. For example, a person monitoring many regions of interest (ROIs) waits for some event or *threat* to come into the sensor domain. In continuous collection modes, most sensor data is entirely useless. Useful data occurs when a sensed *event* interrupts the monitoring system and initiates a threat update. Decision-makers only need a subset of the entire data to perform successful SA and impact assessment to respond to a threat. Specifically, the user needs a time-event update on the object of interest. Thus, data reduction must be performed. The advantage of reducing the incoming data can be realized by using a fusion strategy over time as well as space. Entity data can be fused into a single common operating picture for enhanced CF.

An inherent difficulty resides in the fusion of only two forms of data. If one sensor perceives the correct entity and the other does not, there is a *conflict*. Sensor conflicts increase the cognitive workload and delay action. However, if time, space, and spectral events from different sensors are fused, conflicts can be resolved with an emergent process to allow for better decisions over time. The strategy is to employ multiple sensors from multiple perspectives with user refinement. Additionally, the human-computer interface (HCI) can be used to give the user a global and local CF to guide attention, reduce workload, increase trust, and afford action [15].

Fusion models are based on processing information for decision making. The User-Fusion model highlights the fact that fusion systems are designed for human consumption. The top-down approach explores an information needs pull by queries to the IF system (*cognitive fusion*), while a bottom-up approach pushes data combinations to the user from the fusion system (*display fusion*). The main issues for user-fusion interaction is the ontology for queries, metrics for conveying successful fusion opportunities, and techniques for uncertainty and dimensionality reduction. In the next section, we highlight the main mathematical techniques for fusion based on the User-Fusion model for decision making.

## 4. Fusion Techniques

Both Hall [31] and Waltz [63] present a taxonomy of mathematical fusion techniques. Many texts and papers show mathematical ways to conduct IF. Essentially, all the methods are based on one simple idea: uncertainty reduction. Formal methods for uncertainty calculations are based on a combination of the state, X, and an uncertainty, P, representation:

$$X_F = X_1 \otimes X_2 \otimes \ldots \otimes X_N \quad (1)$$

$$(1 / P_F) = (1 / P_1) \oplus (1 / P_2) \oplus \ldots \oplus (1 / P_N) \quad (2)$$

where $\otimes$ is the state-fusion operation, $\oplus$ is the covariance-fusion operation, and X and P are the state and covariance respectively. X and P can be represented as scalars or matrices. Examples of state-fusion techniques include: Bayes, Dempster-Shafer, [11] DSmT [24], and Fuzzy/Expert analysis [14]. Examples of covariance-fusion are Kalman Filter, decision theory, and statistical analysis. While space limits an exhaustive description of the methods, techniques are presented in the context of the User-Fusion model and the reader should inquire for himself the associated mathematics of each. The basic notion of these techniques is to reduce uncertainty, decrease dimensionality, and increase confidence. To achieve these results, repeated measurements are processed by observing the world and determining how to combine multi-modal information. In the context of the User-fusion model, let's explore the techniques used at each level.

The first is the support algorithms of the pre-processing step, as shown in Figure 8. Key elements include data alignment (or registration [Fig. 7]), the man-machine interface (display), the data-base (storage of data and information), and the associated numerical libraries of routines (such as differential equations solvers).

To perform state (X) and covariance (P) representations, Level 1, object assessment, is divided into position and identity fusion methods, shown in Figure 9 and 10 respectively. Blasch presented premier work that integrated these position and identity fusion methods in 1999 with a "Belief filter for Simultaneous tracking and ID" [10]. Positional fusion is the processing of the entity target location and is conducted with tracking techniques [21,4].

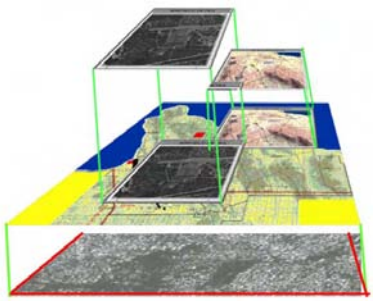


**Figure 7.** Example of registration.

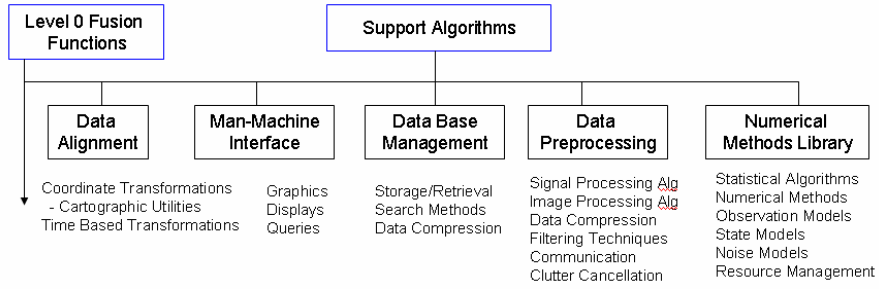



**Figure 8.** Support algorithms Level 0 fusion.

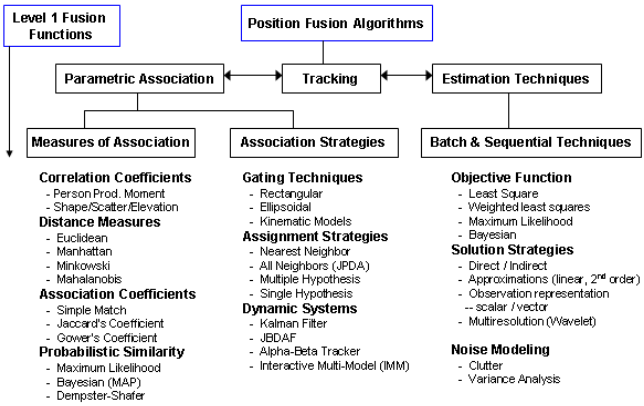



**Figure 9.** Positional fusion in Level 1.

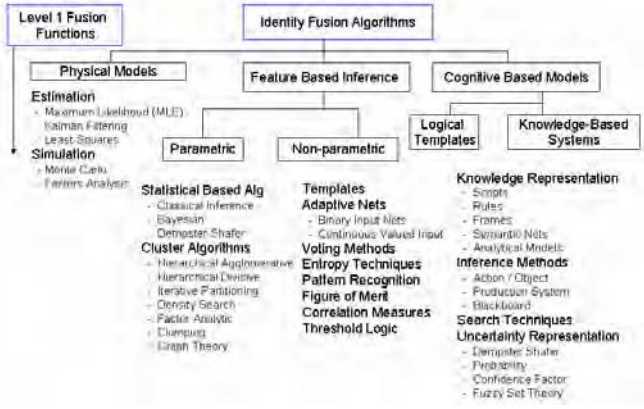



**Figure 10.** Identity fusion in Level 1.

Identity fusion is feature analysis of physical attributes for discrete ID. Cognitive or knowledge-based methods try to mimic the way humans identify objects. Identifying objects can be individual algorithms [47] or a fusion of classifiers [42,48,58].

Level 2, situational assessment, functions include aggregating Level 1 inferences for knowledge and situational awareness. Using context information, the location and identity of objects are mapped to knowledge coordinates (e.g. lat, long, elev.). The mapping is either in a spatial-temporal representation that matches the user cognitive model or correlated with a machine database support function. Many techniques, shown in Figure 11, are explored in the information aggregation to determine a multi-perspective information analysis for users [38]. Schubert [52,53] presented a case of force aggregation as a Level 2 function.

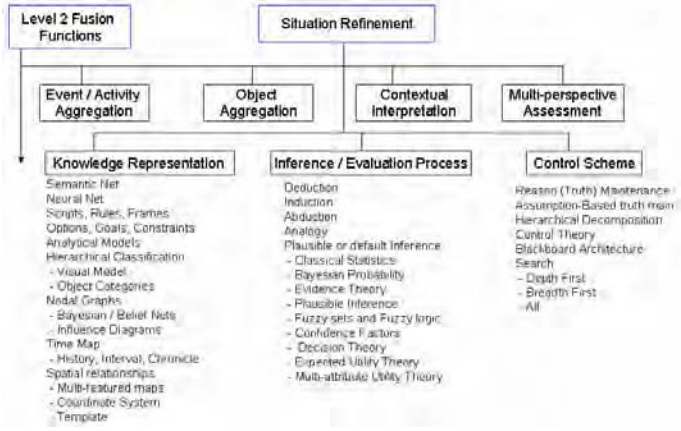



**Figure 11.** Situational assessment techniques.

Level 3 functions of impact assessment are similar to the techniques explored in Level 2, except that in the case of Level 2, the information is aggregated, and in Level 3 the salient information is retained through filtering. Filtering of pertinent information is required to determine the highest priority or useful set of information. Level 2 and 3 functions are similar, as shown in Figure 12.

Shown in Figure 13 is a risk analysis for impact assessment [49,50]. Level 3 functions filter past information, estimate current status, or predict future outcomes. An example is a risk analysis, which uses utility theory to determine the potential impact of a situational state.

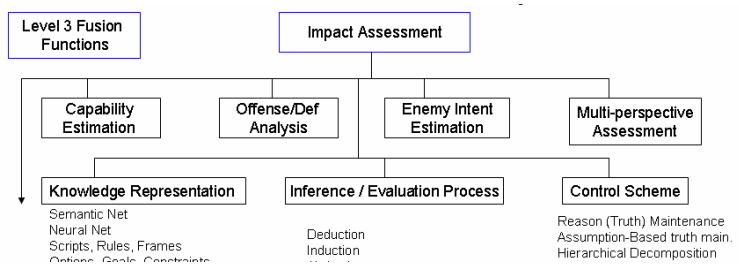



**Figure 12.** Level 3 functions.

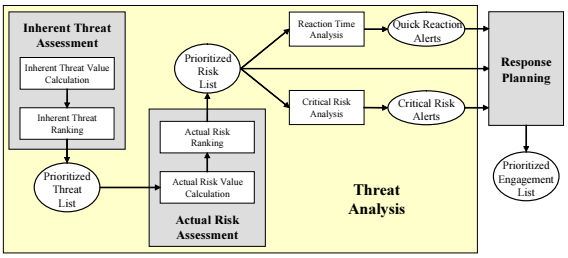



**Figure 13.** Level 3 functions, threat assessment [50].

Level 4, Process Refinement, includes sensor management control of sensors and information. In order to utilize sensors effectively, the IF system must explore service priorities, search methods (breadth or depth), and determine the scheduling and monitoring of tasks. Scheduling is a control function that relies on the aggregated state-position, state-identity and uncertainty information for knowledge reasoning [65]. Typical methods used are (1) objective cost function for optimization, (2) dynamic programming (such as NN methods and reinforcement-learning based on a goal), (3) greedy information-theoretic approaches [43], or (4) Bayes net which aggregates probabilities. Whichever method is used, the main idea is to reduce the uncertainty. Figure 14 shows the many possibilities of techniques for Level 4 processes; however, the user must agree to the strategy embedded in the sensor management control function.


**Figure 14.** Level 4 Sensor Management functions.

Level 5, User Refinement, functions include the (1) selection of models, techniques, and data, (2) determining the metrics for decision making, and (3) performing higher-level reasoning over the information based on the user's needs as shown in Figure 15.

The user's goal is to perform a task or a mission. The user has preconceived expectations and utilizes the level 0–4 capabilities of a machine to aggregate data for decision making. To perform a mission, a user pays engineers to design a machine to augment his work [62]. Each engineer, as a vicarious user, imparts decisions into the algorithm design. However, since the machine was designed by engineers, the "user" might control the fusion system in a different way than was intended. For example, the user plans ahead (forward reasoning) while the machine is reasoning over collected data

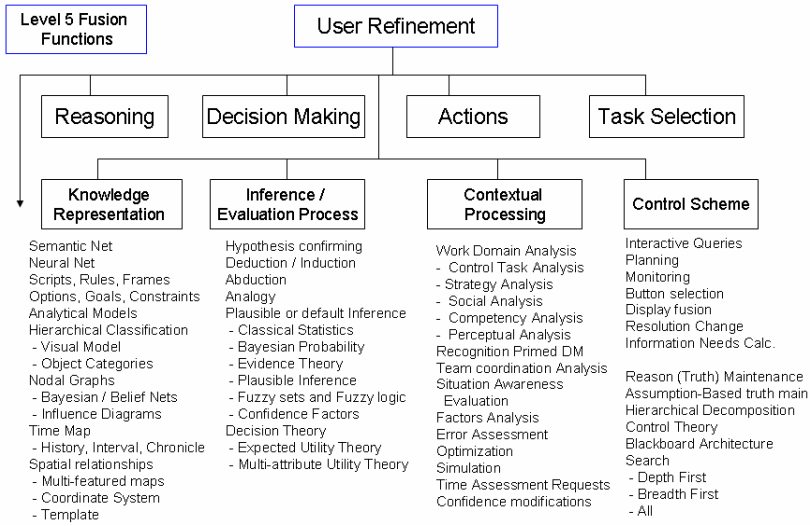



**Figure 15.** Level 5 – User Refinement functions.

(backward reasoning). If a delay exists in the IF reasoning, a user might deem it useless for planning ahead.

Cognitive reasoning by the user includes incorporating contextual information, sometimes unquantifiable, adaptively changing priorities based on task needs, and being sensitive to mission needs. Decision making includes evaluation theory, statistical decision theory, and utility analysis methods. There are models that capture the user's cognitive decision making such as the recognition-primed decision making (RPD) model [35] and the cognitive work analysis (CWA) model [61]. Based on cognitive task modeling, the user does hypothesis confirming, conflict resolution, and prioritizing; all of which can be supported by IF results.

## 5. Fusion Benefits and Limitations

Before designing IF systems for decision making, it is important to understand what the IF limitations are, or stated another way, the misapplications of sensor and data fusion can lead to erroneous results. In this section the fusion advantages and disadvantages are briefly summarized.

Table 2 shows examples of IF limitations. For instance, one difficulty is trying to fuse unregistered data. Before performing the fusion process, the data needs to be aligned so that similar information can be combined. For the case of fusing two medical images (i.e. PET and MRI), all that is needed is to determine the reference coordinate system and then match the image coordinates to the reference frame. For the case of non-IMINT data, a global reference frame is needed to align the non-IMINT to IMINT data. If data is not registered, the fusion process is hampered and operational performance is limited. A case might be the object ID information. The object ID is dependent on the object location. If an ID is evaluated, but not registered to an object in space, a miscorrelation would result. Thus a mapping from ID space to the spatial-temporal reference is needed to fuse position and ID information. If data is misaligned

or inappropriately combined, the user will make an incorrect analysis. Finally, if fusion quality is low, information validity, reliability, and trust is even lower.

Using the techniques described in Section 4 fusion CAN DO:

- **uncertainty reduction**: gain knowledge from entropy info. reduction;
- **detection**: get a signal out of noise (signal detection theory);
- **association**: heterogeneous (across) & homogenous (within) fusion;
- **correlation**: covariance Error Analysis;
- **estimation**: filtering techniques (Kalman Filter) for prediction;
- **combine data**: add data for information such as data overlay;
- **control**: determine where to place sensors in real time;
- **dimension reduction**: reduce plethora of information to a smaller set.

**Table 2.** Fusion advantages and disadvantages

| Advantages | Disadvantages |
|---|---|
| 1. **robust operational performance** is provided because any one sensor has the potential to contribute information while others are unavailable. | 1. can increase the complexity of observations; too much data can be confusing, especially if the data was not collected coincidently. |
| 2. **extended spatial coverage** is provided because one sensor can look where another sensor cannot. | 2. improper results from misaligned data; non-overlapping regions can confuse the system. |
| 3. **extended temporal coverage** is also provided because one sensor can detect or measure an event at times that others cannot. | 3. cannot combine multimodal data that is not consistent in space and time with no physical feature correlation. |
| 4. **increased confidence** accrues when multiple independent measurements are made on the same event or target. | 4. over-reliance on the fused estimate sometimes the result is just a number (i.e. p(id) = 90%) |
| 5. **reduced ambiguity** in measured information is achieved when the information provided by multiple sensors reduces the set of hypotheses. | 5. fusing bad data with good data; when new data is measured, it is always fused with the current estimate, which could increase uncertainty. |
| 6. **improved detection performance** results from effective integration of separate measurements of the same event or target. | 6. fusing data that is not independent such as double counting measurements (i.e. i see the same person again) |
| 7. **enhanced spatial resolution** is provided when multiple sensors can geometrically form a synthetic sensor aperture. | 7. can increase time to collect data on a dynamic object delaying association and increasing miscorrelation. |
| 8. **improved system operational reliability** may result from the inherent redundancy of a multisensor suite. | 8. not understanding the assumptions of the algorithms can allow for misapplication. |
| 9. **reduced dimensionality of the measurement space** (i.e., different sensors measuring) reduces vulnerability of any single portion of the measurement space. | 9. can combine data into one measurement that represents neither separately which results in a fused estimate that means nothing. |
| 10. **increase the validity of information –** can increase current understanding of data into a realizable set through signal detection. | 10. can integrate multimodal data that is not uniquely associated to the context of the situation. |

## 6. Human Refinement in Decision Making

To reason actively over a complex collection of data, a usable set of information must be available on which the user can act [27]. Ontology, *the seeking of knowledge* [12], indicates that the goal must be defined, such as reducing uncertainty. One issue related to an ontology of human refinement is the ability to address uncertainties. The user has to deal with many aspects of uncertainty, such as sensor bias, communication delays, and noise. Heisenberg uncertainty exemplifies the challenge of observation and accuracy [17]. Uncertainty is a measure of doubt [59], which can prevent a user from making a decision. The user can supplement a machine-fusion system to account for deficiencies and uncertainties. Jousselme [34] presented a useful representation of the different types of uncertainty. A state of ignorance can result in an error. An error between expected and measured results from distortion of the data (misregistered) or incomplete data (not enough coverage). With insufficient sensing capabilities, lack of context for situational assessment, and undefined priorities, IF systems will be plagued by uncertainty. Some of the definitions of uncertainty are:

- **vague:** not having a precise meaning or clear sense;
- **probability:** belief in element of the truth;
- **ambiguity:** cannot decide between two results [Webster].

If a machine presents results that conflict with the user's expectations, the user would experience cognitive dissonance. The user needs reliable information that is accurate, timely, and confident. Figure 16 shows that for decision making there are technological operational conditions that affect the user's ability to make informed decisions [36]. The IF system must produce quality results for effective and efficient decision making by increasing fused output quality. An IF design should be robust to object, data, and environment model variations [51]. Using high fidelity models would increase the quality inputs to the IF system which would enhance user-fusion performance capabilities.

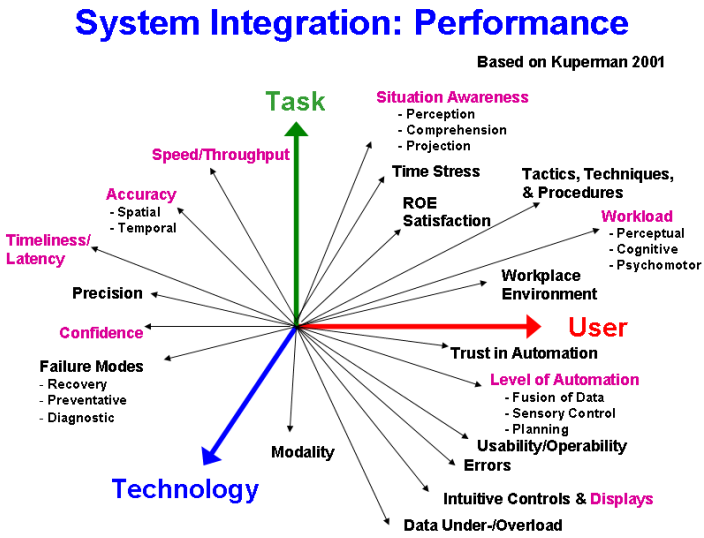



**Figure 16.** Machine-User interaction Performance. [37]

## 7. Conclusions

The key purpose of information fusion system design is to aid the user's decision-making process. The user can passively monitor a fusion display or engage in interactive cognitive fusion provided there is adequate situational awareness and quality information-fusion performance metrics. There is a variety of types of information fusion, all of which facilitate decision making, but the key element of fusion (or key equation) is uncertainty reduction, shown in Figure 17. By recognizing the benefits and limitations of IF, an engineer can appropriately design IF systems to extend user sensing by decreasing dimensionality, increasing confidence, and reducing uncertainty.

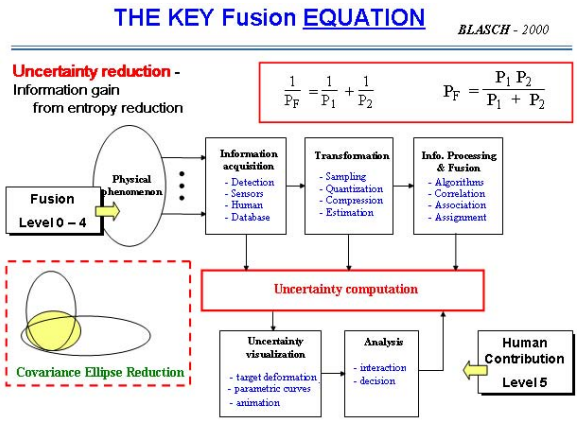



**Figure 17.** The key to fusion is reducing uncertainty.

## References

[1] Abidi, M. and R. Gonzalez (ed.) *Data Fusion In Robotics and Machine Intelligence*, Academic Press, 1992.
[2] Antony, R.T. *Principles of Data Fusion Automation*, Artech House, 1995.
[3] Bar-Shalom, Y., and Li, X.-R. *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, 1998.
[4] Bar-Shalom, Y. & W. Dale Blair (editors), *Multitarget-Multisensor Tracking: Applications and Advances, vol. III*, Artech House, 2000.
[5] Bedworth, M. and J. O'Brien, The Omnibus Model: A New model of data fusion? *IEEE AES Magazine,* April 2000.
[6] Blackman, S. *Multiple-Target Tracking with Radar Applications*, Artech House, Norwood MA, 1986.
[7] Blasch, B., Wiener, W., & Welsh, R. (Eds.). (1997). Foundations of orientation and mobility, (2nd ed.). NY: American Foundation for the Blind Press, Inc.
[8] Blasch, E. Assembling an Information-fused Human-Computer Cognitive Decision Making Tool, *IEEE AES Magazine,* June 2000, pp. 11–17.
[9] Blasch, E. Decision Making for Multi-Fiscal and Multi-Monetary Policy Measurements, *Fusion 98,* Las Vegas, NV, July 6–9, 1998, pp. 285–292.
[10] Blasch, E. *Derivation of a Belief Filter for Simultaneous HRR Tracking and ID,* Ph.D. Thesis, Wright State University, 1999.
[11] Blasch, E. Learning Attributes For Situational Awareness in the Landing of an Autonomous Aircraft, *DASC*, San Diego, CA, October 1997, pp. 5.3.1–5.3.8.
[12] Blasch, E., Ontological Issues in Higher Levels of Information Fusion: User Refinement of the Fusion Process, *Fusion 03*, July 2003.
[13] Blasch, E. Sensor Cognition Display of fused sensory information, *National Sensor and Data Fusion Conference*, San Antonio, TX, 2000.

[14] Blasch, E. and S. Huang, Multilevel Feature-based fuzzy fusion for target recognition, *SPIE, Conf 4051*, April 2000, pp. 279–288.
[15] Blasch, E. and S. Plano, JDL Level 5 Fusion model 'user refinement' issues and applications in group tracking, *SPIE Aerosense 4729,* April 2002.
[16] Blasch, E. & L. Hong. Sensor Fusion Cognition using belief filtering for tracking and identification, *SPIE, Vol. 3719,* April, pp. 250–259, 1999.
[17] Blasch, E. & J Schmitz, Uncertainty Issues in Data Fusion, *NSSDF 02,* August 2002.
[18] Blasch, K., T. Ferré, A. Christensen & J. Hoffmann, New Field Method to Determine Streamflow Timing Using Electrical Resistance Sensors, *Vadose Zone Journal* 1:289–299, 2002.
[19] Brooks, R. & S. Iyengar, *Multisensor Fusion: Fundamentals and Applications with Software*, Prentice Hall 1998.
[20] Cantoni, V., Di Gesu, V., and Setti, A. (Eds.) *Human and Machine Perception – Information Fusion*, Plenum Press / Kluwer, 1996.
[21] Chang, K.C., Z. Tian, S. Mori, & C.-Y. Chong, MAP Track Fusion Performance Evaluation, *Fusion02*, 2002.
[22] Clark J. and A. Yuille, *Data Fusion for Sensory Information Processing Systems*, Kluwer Academic Pub., 1990.
[23] Dasarathy, B.V. *Decision Fusion*, IEEE Computer Society Pr., Los Alamitos, CA, 1994.
[24] Dezert, J. and F. Smarandache, Partial Ordering of hyper-power and matrix representation of belief functions within DSmT, *Fusion03*, 2003.
[25] Endsley, M.R. Predictive utility of an objective measure of Situation awareness. In *Proc HFS 34th Ann Mtg*. Santa Monica, CA: HFES, 1990.
[26] Endsley, M.R. Design and evaluation for situational awareness. *Proceedings of the Human Factors Society Meeting.* Santa Monica, CA: Human Factors Society, 1988.
[27] Fabian, W., Jr, & E. Blasch, "Information Architecture for Actionable Information Production, (for bridging Fusion and ISR Management)," *NSSDF 02,* Aug 2002.
[28] Fitts P. and M. Posner, *Human Performance*. Belmont, CA; Brooke's Cole. 1967.
[29] Goodman, I.R; Mahler, R.P.S., and Nguyen, H.T. *Mathematics of Data Fusion*, Theory and Decision Library. Series B, Mathematical and Statistical Methods, Vol. 37.
[30] Gros, X. *NDT Data Fusion*, Arnold Publishing, 1997.
[31] Hall, D.L. *Mathematical Techniques in Multisensor Fusion*, Artech House, Boston, London, 1992, Norwood, MA.
[32] Hall, D. & Llinas, J. (Eds.) *Handbook of Multisensor Data Fusion*, CRC Press, 2001.
[33] International Society of Information Fusion, *Fusion 1998–2003* proceedings.
[34] Jousselme, A.-L., P. Maupin, E. Bosse, Formalization of Uncertainty in Situation Analysis, DSTO conference, 2003.
[35] Klein, G., Calderwood, R., & MacGregor, D. "Critical decision method for eliciting knowledge." *IEEE Trans on SMC,* 19, 1989, 462–472.
[36] Kuperman, G. Human Systems Interface (HIS) Issues in Assisted Target Recognition (ASTR), *NAECON97*, pp. 37–48, 1997.
[37] Kuperman, G. Personal Communication, 2001.
[38] Lambert, D. Situations for Situational Analysis, *Fusion01*, 2001.
[39] Leonard, J. & H. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer, 1992.
[40] Lou, R. & M. Kay, *Multisensor Integration and Fusion for Machines*, Ablex, 1995.
[41] Manyika, J. & H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*, Ellis Horwood Series in Elec. Eng.
[42] Magnus, A. & M. Oxley, Fusing and Filtering Arrogant Classifiers, Fusion02, 2002.
[43] Musick, S. & R. Malhotra, Chasing the Elusive Sensor Manager, *IEEE NAECON*, 1994.
[44] Osadciw, L., P. Varhsney, & K. Veeramacheneni, Improving Personal Identification Accuracy Using Multisensor Fusion for Building Access Control, *Fusion02*, 2002.
[45] Papoulis, A. & S. Pillai, *Probability and Random Variables and Stochastic Processes*, McGraw Hill, 2001.
[46] Rasmussen, J., Skills, rules and knowledge; signals, signs and symbols; and other distinctions in human performance models, *IEEE Trans. Syst., Man, Cybern*., vol. SMC-13, pp. 257–266, Jan. 1983.
[47] Rogova. G., Combing the results of several NN classifiers, *Neural Networks*, 7(5): 777–781, 1994.
[48] Roli, F. & Kittler (Eds.) *Multiple Classifier Systems*, Springer-Verlag, Lecture notes in Computer Science, Vol. 2364.
[49] Roy, J., From Data Fusion to Situational Analysis, *Fusion01*, 2001.
[50] Roy, J., S. Paradis and M. Allouche, Threat evaluation for impact assessment in situation analysis systems, Vol. 4729, SPIE 2002.
[51] Ross, T. & M. Minardi, Performance Assessment of Detector Reported Confidences, 5427–45, SPIE03.

[52] Schubert, J. Robust Report Level Cluster-to-Track Fusion, *Fusion02*, 2002.
[53] Schubert, J. Evidential Force Aggregation, *Fusion03*, 2003.
[54] Shahbazian, E., D. Blodgett, and P. Labbe, The Extended OODA Model for Data Fusion Systems, *Fusion01*, 2001.
[55] Steinberg, A., C. Bowman, and F. White, Revisions to the JDL Data Fusion model, *NSSDF*, 1999, pp. 235–251.
[56] Sudano J., Pignistic Probability Transforms for Mixes of Low- and High- Probability Events, *Fusion01*, 01.
[57] Torrez, W., Information Assurance Considerations for a Fully Netted Force for Strategic Intrusion Assessment and Cyber Command and Control, *Fusion01*, 2001.
[58] Tremble, C. and P. Valin, Experiments of Individual Classifiers and on Fusion of Set Classifiers, *Fusion02*, 2002.
[59] Tversky, K. and D. Kahneman, *Judgment Under Uncertainty*, Utility, Probability, & Decision Making, Wendt and Vlek (Eds.), D. Reidel Pub, Boston, 1973.
[60] Valin, P., Random Sets and Unification, *NATO ASI on Multisensor and Sensor Data Fusion*, Pitlochry, Scotland, United Kingdom, June 2000, Kluwer publishers, in press.
[61] Varshney, P., Multisensor Data Fusion, *Electronics and Communication Engineering Journal*, vol. 9. pp. 245–253, Dec. 1997.
[62] Vicente, K., *Cognitive Work Analysis, Toward Safe, Productive, and, Healthy Computer-Based Work*, Lawrence Erlbaum, 1999.
[63] Waltz, E. and J. Llinas*, Multisensor Data Fusion*, Artech House, Inc. 1990.
[64] *Webster's New Collegiate Dictionary*, Merriam-Webster Pub. 1995
[65] Xiong, N. and P. Svensson, "Multisensor Management for information fusion: issues and approaches," *Information Fusion*, 3, 2002, 163–186.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# A Gentle Introduction to Fusion of Multiple Pattern Classifiers

Fabio ROLI
*Department of Electrical and Electronic Engineering – University of Cagliari, Piazza d'Armi – 09123 Cagliari (Italy), roli@diee.unica.it*

**Abstract.** In the field of pattern recognition, fusion of multiple classifiers is currently used for solving difficult recognition tasks and designing high performance systems. This chapter is aimed at providing the reader with a gentle introduction to this fertile area of research. We open the chapter with a discussion about motivations for the use of classifier fusion, and outline basic concepts on multiple classifier systems. Main concepts about methods and algorithms for creating and fusing multiple pattern classifiers are reviewed. The chapter closes with a critical discussion of current achievements and open issues.

**Keywords.** Information fusion, pattern classification, fusion of multiple classifiers, multiple classifier systems, ensemble learning

## 1. Motivations for Multiple Classifier Fusion and Basic Concepts

### *1.1. Introduction*

During the last thirty years, the scientific discipline of pattern classification has developed into a rich research area characterized by three main branches of theoretical and experimental investigation: the so called statistical, structural, and syntactic approaches. Along the way, important contributions to such main pattern classification research lines were provided by the machine learning, neural networks, artificial intelligence, information fusion, and statistics disciplines.

Despite the encouraging results provided by the three traditional approaches, it was recognized very early that the key to pattern recognition problems cannot lie on a single technique. As Kanal wrote in his 1974 paper: "… no single technique is applicable to all problems….what we have is a bag of tools and a bag of problems…" [1]. Researchers of the AI community used similar arguments for supporting the importance of "hybrid" models in knowledge representation and processing [2]. The general idea of combining multiple techniques or merging multiple experts for solving difficult decision-making tasks is really very old [3–5]. In the pattern recognition field, after some early works [6,7], this idea strongly emerged under many different names during the 1990's: multiple experts, multiple classifier systems, hybrid systems, decision fusion, and other names [8]. This ambitious research trend was also motivated by empirical observations about the complementarity of different pattern classifier designs, natural requirements of sensor fusion applications, and intrinsic difficulties of the optimal choice of some classifier design parameters, such as the architecture and the initial weights for a neural network. Although the idea of combining multiple pattern classifi-

ers naturally points to the concept of hybrid systems [9,10], most of the research conducted in the pattern recognition field has dealt with the combination of classifiers of the same type (e.g., the combination of statistical classifiers) [11]. Some works on general methods for designing hybrid systems appeared in the AI literature [9,10,12]. However, due to the inherent complexity of this general issue, methods for designing pattern recognition systems made up of real hybrid modules (e.g., statistical and structural classifiers, or statistical and symbolic algorithms) were quite specific to the particular application considered [11]. Therefore, clear methodologies for designing hybrid systems, namely, methods for combining structural and statistical methods or integrating connectionist and symbolic techniques, are still beyond the state of the art.

In the following section, we critically review the main motivations for, and the rationale behind, the use of multiple pattern classifier fusion.

### *1.2. Motivations*

The traditional approach to pattern classifier design is based on the so called "classifier evaluation and selection" approach [13]. In a few words, the performances of a set of different classification algorithms (or different "versions" of the same algorithm) are assessed against a representative pattern sample, and the best classifier is selected. The classifier evaluation and selection approach to classifier design basically works as follows:

- design a set of N classifiers $C_1$, $C_2$,…,$C_N$;
- assess classifier errors $E_1<E_2<E_3<\ldots<E_N$ (with related confidence intervals, if possible) using a validation set, which must be independent from the training set;
- select the best classifier $C_1$, and consider it the "optimal" one.

It is well known that this traditional design approach works well when a large and representative data set is available ("large" sample size cases), so that estimated errors allow to select the best classifier for unknown samples [13–15]. However, in many small sample-size real cases, the validation set often only provides "apparent" errors which can substantially differ from true errors (i.e., from errors on the unknown test data). This is the well known "generalization" error, which can make impossible the selection of the optimal, if any, classifier, and, in the worst case, one could select the worst classifier on the basis of performances assessed on the validation set.

In the small sample size case, it is quite intuitive that one could avoid the selection of the worst classifier by, for example, "averaging" over the individual classifiers. Recently, Dietterich provided a paradigmatic example of this situation [16]. Assume that you have few training data with respect to the size of the feature space, and the N classifiers you designed provide the same accuracy on your validation data. It is easy to see that you can avoid selecting the worst classifier by averaging over the individual classifiers. I call this the "worst" case motivation for the use of multiple pattern classifier fusion, as it points out that fusion can protect you against the selection of the worst classifier.

Besides avoiding the selection of the worst classifier, under particular hypotheses, fusion of multiple classifiers can improve the performance of the best individual classifier and, in some special cases, provides the optimal Bayes classifier. This is possible if individual classifiers make "complementary" errors. Fortunately, we have many experimental evidences about that in practical applications [17–20]. We also have some

theoretical supports about the benefits of classifier complementarity for some classes of fusers (e.g., linear combiners and majority voting). For example, Tumer and Ghosh, Roli and Fumera, showed that averaging outputs of individual classifiers with unbiased and uncorrelated errors can improve the performance of the best individual classifier and, for an infinite number of classifiers, provides the optimal Bayes classifier [21,22]. I call this the "best" case motivation for the use of multiple pattern classifier fusion, as it points out that fusion can improve the performance of the best individual classifier.

There are other motivations for the use of multiple pattern classifier fusion:

- computational motivations. For example, many learning algorithms, such as back-propagation in neural nets, suffer from the problem of local minima. Finding the best classifier can be difficult even with enough training data. Fusion of multiple classifiers constructed by running the learning algorithm from different starting "points" (e.g., different initial random weights in the case of neural nets) can help to avoid local minima, thus improving performance;
- the motivation that I call the "curse" of the pattern classifier designer. This motivation comes from: a) the need to avoid having to make a meaningful choice of some arbitrary initial condition, such as the initial weights for a neural network; b) the intrinsic difficulty of choosing appropriate design parameters (e.g., the number of hidden neurons, or the value of the "k" parameter for the k-nearest neighbours classifier); c) the "saturation" of design improvement. In real cases, during the design phase, the performance of any individual classifier often reaches a "plateau," despite any effort in further varying the design parameters;
- in multi sensor applications, fusion of multiple classifiers is naturally motivated by the application requirements;
- monolithic vs. modular classifier systems. Different classifiers can have different domains of competence. Fusion is the natural way to exploit such complementarity.

### *1.3. Basic Concepts on Multiple Classifier Systems*

In the field of pattern recognition, research areas dealing with the fusion of multiple classifiers are known as "multiple classifier systems" [8,16]. Hereafter, we also use this term to illustrate the basic concepts on fusion of multiple classifiers.

Basically, the multiple classifier system (MCS) consists of an ensemble of different classification algorithms and a "function" to fuse classifier outputs. More precisely, the different types of MCSs can be characterized by:

- the fusion architecture: parallel, serial, hybrid. Parallel architectures: multiple classifiers operate in parallel. A single combination function merges the outputs of the individual classifiers. Serial architectures: classifiers are applied in succession, with each classifier producing a reduced set of possible classes. A primary classifier can be used. When it rejects a pattern, a secondary classifier is used, and so on. Hybrid architectures can merge, in various ways, parallel and serial topologies;
- the classifier ensemble: type and number of combined classifiers. The ensemble can be subdivided into subsets in the case of non parallel architectures;

– the fuser: that is, the fusion function used. So far, two main types of fusers are used. Integration (fusion) functions: for each pattern, all the classifiers contribute to the final decision. Integration assumes competitive classifiers. Selection functions: for each pattern, just one classifier, or a subset, is responsible for the final decision. Selection assumes complementary classifiers. Integration and selection can be "merged" for designing a hybrid fuser. Multiple functions may be necessary for non parallel architectures.

So far, research on MCSs has focused on parallel architectures. General methodologies and clear foundations are mostly available only for parallel architectures. MCSs based on other architectures (serial, hierarchical, hybrid, etc.) were highly specific to the particular application considered. In the remainder of this chapter, we focus on parallel architectures. Many of the basic concepts we will discuss also hold for different architectures.

Now, let us briefly discuss the critical relationship between the classifier ensemble and the combination function (the "fuser").

Classifier fusion is obviously useful only if the combined classifiers are mutually complementary. Ideally, classifiers should exhibit high accuracy and high diversity. It is easy to see that the required degree of error diversity depends on the fuser complexity. For example, if we are using majority voting as fuser, errors can be tolerated, supposing that the majority is always correct. A. Sharkey defines four error-diversity levels [23]:

– Level 1: no more than one classifier is wrong for each pattern;
– Level 2: the majority is always correct;
– Level 3: at least one classifier is correct for each pattern;
– Level 4: all the classifiers are wrong for some patterns.

As the degree of error diversity among classifiers is so crucial for fusion, researchers proposed various measures to assess how diverse (in other words, "complementary") two classifier are [24]. As an example, Kuncheva proposed the use of Q statistics to assess how diverse the classifiers $c_i$ and $c_k$ are [24]:

$$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{1}$$

where $N^{11}$ is the number of patterns correctly classified by both classifiers, $N^{00}$ is the number of patterns wrongly classified by both classifiers, and so on.

The Q statistics varies between –1 and 1. Classifiers which tend to classify the same patterns correctly will have values of Q close to 1, and those which commit errors on different patterns will render Q negative.

So far, measures of error diversity in classifier ensembles are a matter of on-going research, as no measure showed a clear relation with MCS performance. In my opinion, the only clear achievement is that the required degree of error diversity depends on the fuser complexity:

– simple fusers can be used for classifiers that exhibit a high degree of error diversity (i.e., a high degree of complementarity);

– complex fusers, for example, a dynamic selector, are necessary for classifiers with a low degree of error diversity.

To sum up, the design of MCS involves two main phases: the design of the classifier ensemble, and the design of the fuser. The design of the classifier ensemble is aimed at creating a set of complementary/diverse classifiers. The design of the combination function/fuser is aimed at creating a fusion mechanism that can exploit the complementarity/diversity of classifiers and optimally combine them. These two design phases are obviously linked [25].

In the following sections, we provide the reader with a short overview of methods for creating and fusing multiple classifiers. We will refer the reader to appropriate references for details.

## 2. Methods for Creating Multiple Classiers

The effectiveness of MCS relies on combining diverse/complementary classifiers. Several approaches have been proposed to design ensembles made up of complementary classifiers.

### *2.1. Using Problem and Designer Knowledge*

When problem or designer knowledge is available, "complementary" classification algorithms can often be designed quite easily. This is the case of:

– applications with multiple sensors [26];
– applications where complementary representations of patterns are possible (e.g., statistical and structural representations) [27];
– when designer knowledge allows varying the classifier type, architecture, or parameters to create complementary classifiers [23].

It is easy to see that these are heuristic approaches, which perform as well as the problem/designer knowledge allows designing complementary classifiers.

### *2.2. Injecting Randomness*

Simple design methods are based on injecting randomness in the classification/training algorithm [23]. For example:

– neural networks: the back-propagation algorithm is often run several times using different (random) starting points (initial weights);
– decision trees: the test at each internal node can be chosen randomly between the top *n* best tests.

Again, these are basically heuristic approaches. We can only hope that they produce complementary classifiers.

### *2.3. Manipulating Training Data*

These methods are based on training N classifiers with N different training sets.

*Data splitting:* training data are randomly subdivided into N disjoint subsets. Each classifier is trained on a different subset (infeasible for small training sets).

*Cross-validated committees:* training data are randomly subdivided into N disjoint subsets. N overlapping training sets are constructed by dropping out a different one of the N subsets.

*Bagging:* the method proposed by Breiman [28] for constructing multiple classifiers by training data manipulation. Bagging is based on obtaining different training sets of equal size as the original one, by using a statistical technique named bootstrap. The bootstrap technique is based on the concepts of bootstrap sample and bootstrap replication:

- bootstrap replication: a classifier trained with a bootstrap sample;
- bootstrap sample: $x^*=(x^*_1,\ldots,x^*_n)$, random sample of size n drawn with replacement from the original sample $x=(x_1,\ldots,x_n)$. Each sample in x can appear in x* zero times, once, twice, etc.

The resulting training sets $L_i$, i=1,…,N, contain usually small changes with respect to the original training set L. Figure 1 illustrates the Bagging method for creating and fusing N classifiers.

The rationale behind Bagging is that instances of an "unstable" classifier constructed on different bootstrap samples can exhibit significant differences [28]. A classifier is named "unstable" if small changes in the training set cause substantial changes in its outputs.

Bagging is a method for constructing multiple classifiers, not a fusion rule. In principle, any fuser can be used. Usually, simple fusers are used, such as simple averaging or majority vote. Reported experimental results show that bagging is effective when used with simple combining rules. However, the use of complex rules should be further investigated.

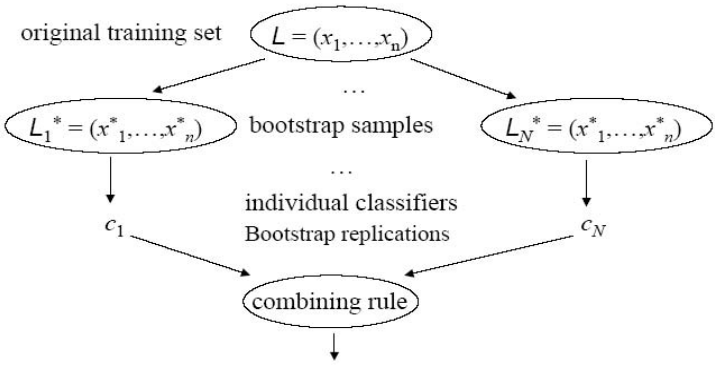



**Figure 1.** Scheme of Bagging. The N classifiers $c_1, c_2,\ldots,c_N$ are trained on the bootstrap samples generated by the original training set.

*AdaBoost:* the AdaBoost algorithm [29] is aimed at producing highly accurate ("strong") classifiers by combining "weak" instances of a given base classifier. AdaBoost iteratively constructs an ensemble of N complementary classifiers. Additional weak classifiers are introduced iteratively if necessary, and they are trained on samples that previous classifiers have misclassified. The resulting classifiers are com-

bined by weighted voting. AdaBoost is an ensemble learning method, not a general purpose method for constructing multiple classifiers like Bagging.

### *2.4. Manipulating Input Features*

Manual or automatic feature selection/extraction can be used for generating diverse classifiers using different feature sets. For example, subsets related to different sensors, or subsets of features computed with different algorithms. Different feature sets can be generated using different feature extraction algorithms applied to the original feature set. The "hope" is that classifiers using different features are complementary. Manual or automatic selection can work with sets of redundant/irrelevant features.

A successful example of this approach is the Random Subspace Method by Ho which consists in random selection of a certain number of subspaces from the original feature space, and training a classifier on each subspace [30].

### *2.5. Manipulating Output Features*

Another interesting idea is building complementary classifiers by partitioning the set of classes in different ways. Each component classifier is trained to solve a subset of the N class problem. For instance, each classifier could solve a two class problem (e.g., One vs. All strategy). A suitable combination method able to "recover" the original N class problem is necessary. Dietterich and Bakiri proposed a technique called Error-Correcting Output Coding (ECOC) [31]. ECOC works well for a large number of classes. But it could be applied to subclasses within a smaller number of classes.

## 3. Methods for Fusing Multiple Classifiers

Methods for fusing multiple classifiers can be classified according to the type of information produced by the individual classifiers [32,33]:

- abstract-level fusion rules: rules used when each classifier outputs a unique class label for each input pattern;
- rank-level fusion rules: rules used when each classifier outputs a list of possible classes, with ranking, for each input pattern;
- measurement-level fusion rules: rules used when each classifier outputs class "confidence" levels for each input pattern.

For each of the above categories, methods can be further subdivided into: integration vs. selection rules, and fixed rules vs. trained rules.

### *3.1. Abstract-Level Fusion Rules*

*Majority voting:* let us consider the N abstract ("crisp") classifiers outputs S(1),…,S(N) associated to the pattern x. Majority fusion rule: class label $c_i$ is assigned to the pattern x if $c_i$ is the most frequent label in the crisp classifiers outputs. Usually N is odd. The frequency of the winner class must be at least N/2. If the N classifiers make independent errors and they have the same error probability, lower than 0.5, then it can be shown that the error of the majority voting rule is monotonically decreasing in N [34].

Clearly, the performances of the majority vote quickly decrease for dependent classifiers.

*Behavior Knowledge Space:* in the Behavior Knowledge Space (BKS) method, every possible combination of abstract-level classifiers outputs is regarded as a cell in a look-up table [35,36]. Each cell contains the number of samples of the validation set characterized by a particular value of class labels. Reject option by a threshold is used to limit error due to "ambiguous" cells. A simple example of BKS operation is given in Figure 2 for a two-class problem and three classifiers

*Classifiers outputs $S^{(1)}$, $S^{(2)}$, $S^{(3)}$*

| Class | 0,0,0 | 0,0,1 | **0,1,0** | 0,1,1 | 1,0,0 | 1,0,1 | 1,1,0 | 1,1,1 |
|---|---|---|---|---|---|---|---|---|
| 0 | 100 | 50 | **76** | 89 | 54 | 78 | 87 | 5 |
| 1 | 8 | 88 | **17** | 95 | 20 | 90 | 95 | 100 |

$$P(c=0 \mid S^{(1)}=0, S^{(2)}=1, S^{(3)}=0) = \frac{76}{76+17} = 0.82 \geq th$$

**Figure 2.** Example of Behavior Knowledge Space operation for a two-class problem and three classifiers. Every possible combination of abstract-level classifiers outputs is regarded as a cell in the look-up table. The probability that the input pattern belongs to class "0" when the three classifiers assign it to the classes 0,1,0, respectively, is given.

Abstract level methods are the most general fusion rules. They can be applied to any ensemble of classifiers, even to classifiers of different types. The majority voting rule is one of the simplest fusers. This made way for theoretical analyses [34]. When prior performance is not considered, the requirements of time and memory are negligible. As we proceed from simple rules to adaptive (weighted voting) and trained (BKS) rules the demands on time and memory quickly increase. Trained rules, such as BKS, impose heavy demands on the quality and size of data set.

### *3.2. Rank-Level Fusion Rules*

Some classifiers provide class "scores," or some sort of class probabilities. This information can be used to "rank" each class. In general, if C ={$c_1$,…,$c_k$} is the set of classes, these classifiers can provide an "ordered" (ranked) list of class labels (Figure 3).

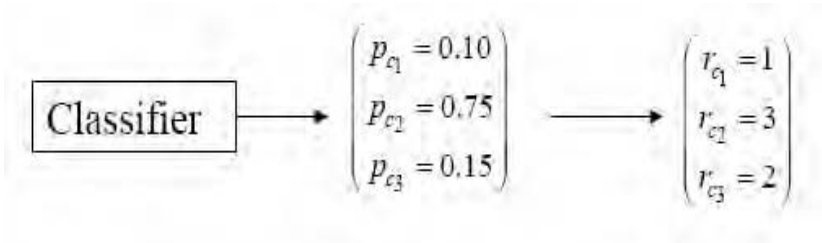



**Figure 3.** Example of ranked list of class labels for a three-class problem. The class with the highest value of posterior probability, i.e., class "2," has the highest rank $r_{c2}$=3, and so on.

The most commonly used rank-level fusion rule is the Borda Count method. This method can be explained with a simple example. Assume you have a problem with N=3 classifiers and k=4 classes C={a, b, c, d}. In addition, assume that, for a given pattern, the ranked outputs of the three classifiers are as follows:

**Table 1.** Ranked output of classifiers

| Rank value | Classifier 1 | Classifier 2 | Classifier 3 |
|---|---|---|---|
| 4 | c | a | b |
| 3 | b | b | a |
| 2 | d | d | c |
| 1 | a | c | d |

For this example, it is easy to see that the winner-class is "b" because it has the maximum overall rank.

Rank-level fusion rules are suitable in problems with many classes, where the correct class may often appear near the top of the list, although not *at* the top (e.g., word recognition with sizeable lexicon). On the other hand, rank-level rules are not supported by clear theoretical underpinnings, and results depend on the scale of numbers assigned to the choices.

### *3.3. Measurement-Level Fusion Rules*

This kind of fusion rule can be used when each classifier outputs class "confidence" levels for each input pattern (Figure 4). It should be noted that normalization of classifier outputs is not a trivial task when combining classifiers with different output ranges and different output types (e.g., distances vs. membership values).

Different rules can be used for measurement-level fusion: linear combination of classifier outputs, product of classifier outputs, order-statistics operators (max, min, med), etc. In addition, the soft outputs of the N individual classifiers can be considered as features of a new classification problem (classifier-output feature space). In other words, classifiers can be regarded as feature extractors. Therefore, another classifier can be used as fuser. This is the so-called "stacked" approach [37], or "meta-classification" [38], or "brute-force" approach [24]. In order to train the meta-classifier, the outputs of the N individual classifiers on an independent validation set must be used.

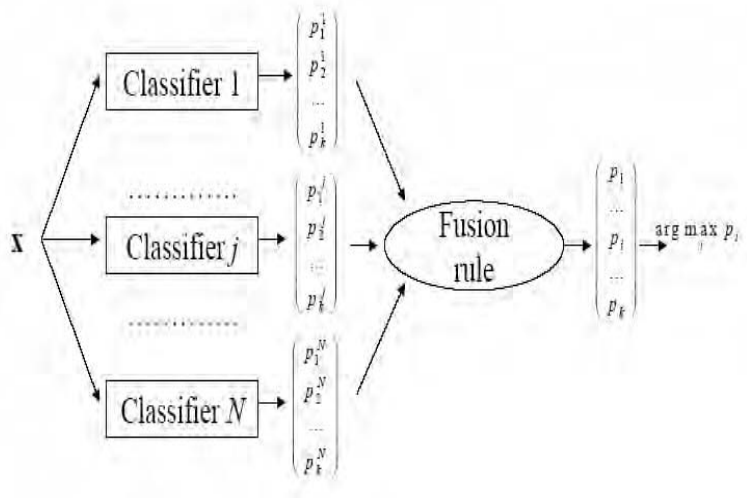



**Figure 4.** Architecture of measurement-level fusion rules.

## 4. Applications, Achievements and Open Issues

As discussed in the series of the International Workshops on Multiple classifier systems (www.diee.unica.it/mcs), classifier fusion can play an important role for all applications where decision-level fusion is required. Let me recall here the following question posed to the participants of the Int. School on Ensemble Methods for Learning Machines (Vietri, Italy, Sept. 2002): "Identify an application/problem for which fusion of multiple classifiers surely does not work (i.e., does not provide any benefit)." No application was identified clearly! My personal position was (and still is): for cases where multiple classifiers cannot improve performance, they can nonetheless increase reliability. Therefore, the use of MCS is just a matter of cost vs. benefit.

MCSs are currently used in many applications; among others, sensor fusion, remote sensing, biometrics, documents analysis and OCR, data mining and KDD.

However, one should ask:

- after more than a decade of research in the MCS field, where do we stand?
- what are the main results?
- today, what does a designer of pattern classification systems have in her/his hands that was not available ten years ago?

After more than a decade of research, we have two main approaches to pattern classifiers fusion [11]:

- coverage optimization methods: a simple combination function is given. The goal is to create a set of complementary classifiers that can be combined optimally (Bagging, Random Subspace, etc.);
- decision optimization methods: a set of carefully designed and optimized classifiers is given and unchangeable, the goal is to optimize the combination function.

For both coverage and decision optimization methods, we have many empirical evidences of their effectiveness, and some theoretical supports. However, for a given task, the choice of the most appropriate fuser still lies on the "old" paradigm of model evaluation and selection, and a unifying framework is clearly beyond the state of the art.

But if the choice of the most appropriate combination method lies again on the "old" paradigm of model evaluation and selection, then, one could say that MCS research simply moved the original problem to a different level. Instead of looking for the best classifier, now we look for the best combination rule. Have we fallen into a vicious circle? Are we looking for the best set of combination rules too soon [11]?

Despite the above unsolved issues, we have some good news:

- MCS provides solutions for cases where the selection of the best classifiers is very difficult, as the estimates of generalization error are optimistically biased ("apparent" errors);
- MCS provides solutions for cases where the optimisation of an individual classifier is very difficult. Increases in design effort provide very small improvements. In such cases, coverage optimisation methods can be a solution;
- MCS can avoid the choice of arbitrary initial conditions, or the tuning of difficult design parameters;
- MCS provides solutions for cases where a single classifier that is optimal on the whole feature space does not exist. Different classifiers can be optimal in different regions of the feature space ("Modular" MCS).

Therefore, I think that there is room for optimism about the future of research on the fusion of multiple classifiers. In particular, while theoretical, experimental and engineering research progress, I think that the number of practical tasks for which we will be able to choose (in a probabilistic sense) the best single or combined system will increase. For each task, the size of the set of single or combined systems that we will have to assess and compare for selecting the best one will decrease.

## References

[1] L. Kanal, Patterns in pattern recognition, IEEE Trans. Information Theory, IT-20, 1974, pp. 697–722.
[2] M. Minsky, Logical versus analogical or symbolic versus connectionist or neat versus scruffy, AI Magazine, 12, 1991, pp. 34–51.
[3] N.C. de Condorcet, Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix, Imprimerie Royale, Paris, 1785.
[4] P.S. de Laplace, Deuxième supplément a la théorie analytique des probabilités. Networks, 1818. Reprinted (1847) in Oeuvres Complete de Laplace, vol. 7 (Paris, Gauthier-Villars) 531–580.
[5] J. Von Neumann, Probabilistic logics and the synthesis of reliable organisms from unreliable components, In C.E. Shannon and J.McCarthy Eds., Automata Studies, 43–98, Princeton Univ. press, 1956.
[6] B.V. Dasarathy, B.V. Sheela, A composite classifier system design: concepts and methodology, Proceedings of the IEEE, 67, 5, May 1979, pp. 708–713.
[7] L.A. Rastring, and R.H. Erenstein, Method of Collective Recognition, Energoidzdat, Moscow, 1981 (in Russian). In English: ISSN0038-5328/81/0006-0022§7.50/0, Scripta Publishing Co., 1982.
[8] Multiple Classifier Systems, Springer-Verlag, Lecture Notes in Computer Science, Vol. 2364, 2002, F. Roli, and J. Kittler Eds.
[9] Hybrid Architectures for Intelligent Systems, CRC Press, 1992, A. Kandel, G. Langholz Eds.
[10] Hybrid Methods in Pattern Recognition, World Scientific Publishing, 2002, H. Bunke and A. Kandel Eds.

[11] T.K. Ho, Multiple Classifier Combination: Lessons and Next Steps, in H. Bunke and A. Kandel (Eds.), Hybrid Methods in Pattern Recognition, World Scientific Publishing, 2002.
[12] ECAI Workshop on Foundations of connectionist-symbolic integration: representation, paradigms, and algorithms, Berlin, Germany, 2000.
[13] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, John Wiley&Sons, 2001.
[14] S. Raudys, Statistical and neural classifiers: an integrated approach to design, Springer-Verlag, 2001.
[15] L. Devroye, L. Gyorfi, G. Lugosi, A probabilistic theory of pattern recognition, Springer-Verlag, 1996.
[16] J. Kittler and F. Roli (Eds.) Multiple Classifier Systems, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1857, 2000.
[17] G. Giacinto, F. Roli, and L. Bruzzone. Combination of Neural and Statistical Algorithms for Supervised Classification of Remote-Sensing Images. Pattern Recognition Letters, May 2000, vol. 21, no. 5, pp. 385–397.
[18] G.L. Marcialis and F. Roli, Experimental results on fusion of multiple fingerprint matchers, Proc. of the 4th Int. Conf. on Audio- and Video-Based Person Authentication, June, 9–11, 2003, Guildford, U.K., J. Kittler and M.S. Nixon Eds., LNCS 2688, pp. 814–820.
[19] G. Giacinto, F. Roli, L. Didaci, Fusion of multiple classifiers for intrusion detection in computer networks, *Pattern Recognition Letters*, 24(12), 2003, pp. 1795–1803.
[20] Roli, F., Fumera, G., Kittler, J., 2002. Fixed and Trained Combiners for Fusion of Unbalanced Pattern Classifiers, Proc. Fifth Int. Conference on Information Fusion, pp. 278–284.
[21] Tumer, K., Ghosh, J.: Linear and Order Statistics Combiners for Pattern Classification. In: Sharkey, A.J.C. (ed.): Combining Artificial Neural Nets. Springer, 1999, 127–161.
[22] G. Fumera, F. Roli, Analysis of error-reject trade-off in linearly combined multiple classifiers, Pattern Recognition, 2004, in press.
[23] Sharkey, A., Multi-Net Systems, Combining Artificial Neural Nets, Ensemble and Modular Multi-Net Systems, Springer-Verlag, 1999, pp. 1–27.
[24] Kuncheva, L.I., Combinations of multiple classifiers using fuzzy sets, in Fuzzy Classifier Design, 233–267 Springer-Verlag, 2000.
[25] F. Roli, G. Giacinto, Design of Multiple Classifier Systems, in H. Bunke and A. Kandel (Eds.), Hybrid Methods in Pattern Recognition, World Scientific Publishing, 2002.
[26] B.V. Dasarathy, Decision Fusion, IEEE Computer Society Press, 1994.
[27] Y. Yao, G.L. Marcialis, M. Pontil, P. Frasconi, F. Roli, Combining Flat and Structural Representations for Fingerprint Classification with Recursive Neural Networks and Support Vector Machine, Pattern Recognition, Vol. 36 (2), 2003, pp. 397–406.
[28] Breiman, L., Bagging Predictors, Machine Learning, 24, 123–140, 1996.
[29] Freund Y., R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Tech. Rep. AT&T Bell Labs, Murray Hill, NJ, 1995.
[30] Ho, T.K., The random subspace method for constructing decision forests, IEEE Trans. On Pattern Analysis and Machine Intelligence 20, 832–844, 1998.
[31] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research, 2, 263–286, 1995.
[32] L. Xu, A. Krzyzak, and C.Y. Suen, Methods for combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. on Systems, Man, and Cyb., Vol. 22, No. 3, May/June 1992, pp. 418–435.
[33] C.Y. Suen, L. Lam. Multiple classifier combination methodologies for different output levels, Proc. First International Workshop on Multiple Classifier Systems (MCS 2000). Springer-Verlag Pub., Lecture Notes in Computer Science, Vol. 1857, J. Kittler and F. Roli Eds., (2000) pp. 52–66.
[34] Lam, L., Suen, C.Y., Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance. IEEE Trans. on Systems, Man and Cybernetics – Part A 27, 1997, pp. 553–568.
[35] Huang, Y.S. and Suen C.Y., A method of combining multiple experts for the recognition of unconstrained handwritten numerals, IEEE Trans. on Pattern Analysis and Machine Intelligence 17, 1995, pp. 90–94.
[36] S. Raudys, F. Roli, The Behavior Knowledge Space Fusion Method: Analysis of Generalization Error and Strategies for Performance Improvement, *4th Int. Workshop on Multiple Classifier Systems (MCS 2003)*, Guildford, United Kingdom, June 11–13 2003, T. Windeatt and F. Roli Eds., LNCS 2709, pp. 55–64.
[37] Wolpert, D.H., Stacked generalisation, Neural Networks 5, 241–259, 1992.
[38] G. Giacinto, and F. Roli, Ensembles of Neural Networks for Soft Classification of Remote Sensing Images, Proc. of the European Symposium on Intelligent Techniques, March 20–21, 1997, Bari, Italy, 1997.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Uncertainty Management for Intelligence Analysis

Ronald R. YAGER
*Machine Intelligence Institute, Iona College, New Rochelle, NY 10801*
*yager@panix.com*

**Abstract.** Considerable concern has arisen regarding the quality of intelligence analysis. This has been in large part motivated by the task, prior to the Iraq war, of determining whether Iraq had weapons of mass destruction. One problem that made this analysis difficult was the uncertainty in much of the information available to the intelligence analysts. In this work we introduce some tools that can be of use to intelligence analysts for representing and processing uncertain information. We make considerable use of technologies based on fuzzy sets and related disciplines such as approximate reasoning.

**Keywords.** Intelligence analysis, fuzzy sets, information fusion, uncertainty

## 1. Introduction

In his report to Congress [1], David A. Kay, who led the US government's efforts to find evidence of Iraq's illicit weapons programs, reported that the current intelligence systems dealing with weapons of mass destruction are increasingly based on limited information. In light of this situation, he indicated that modern intelligence analysis systems need a way for an analyst to say, "I don't have enough information to a make a judgment," a capacity that he felt the current intelligence systems do not possess. Central to attaining this capability is the ability to deal with uncertain and imprecise information. We believe that fuzzy logic with its focus on uncertainty can help. It has the ability to simultaneously exploit both precise formal measurements of the type obtained from state of the art electronic and mechanical monitoring devices as well the type of imprecise information obtained from human sources which is often perception based and expressed in linguistic terms. Here, we begin to look at the possibilities of using fuzzy logic [2] and related soft computing technologies to provide the tools necessary to supply this capability to intelligence analysts. As we shall subsequently see, the dual measures of possibility and certainty [3] provide a useful way of formalizing the concept of not knowing with certainty.

## 2. Variables and Question Answering

By a variable we shall mean an attribute associated with some specific object. Thus, if V is a variable then V ≡ attribute (object). John's age and the number of nuclear de-

vices possessed by North Korea are examples of variables. In the first case, the attribute is age and the object is John. In the second case, the attribute is the quantity of nuclear devices and the object is North Korea. Typically with a variable, we assume it has a domain, X, consisting of the set of possible values. In many situations, a task of great interest is the answering of some question about a variable. For example, is John over 65? Another closely related task is that of making a decision in which knowledge about a variable is central to the decision. For example, a bartender deciding whether to serve John a drink must ascertain that his age is at least 21.

We emphasize the distinction between the task of finding the value of a variable and that of answering a question about a variable. Clearly, although knowing the value of a variable can help in answering a question, it is not always necessary. That is, there can be some uncertainty and still we can answer a question about a variable with certainty.

In order to be able to answer a question about the value of a variable, we must draw upon all our sources of information about the variable. Figure 1 illustrates this situation. The information provided by these sources may be related to the variable of interest in a number of different ways. It may be information directly about the value of the variable of interest, an observation on the age of John. For example, a birth certificate. It may be about the attribute without specific reference to John. Human beings typically live no more than about 85 years. It may be information about the value of another attribute associated with John, "the color of John's hair is grey." It may be information relating the variable of interest to other attributes or variables, "John is five years younger than Mary." Furthermore, each of these pieces of information may have different degrees of credibility. In addition, the information from the sources may be obtained from precise measurement or may be based upon perceptions and observations. It may be expressed formally or in linguistic terms.

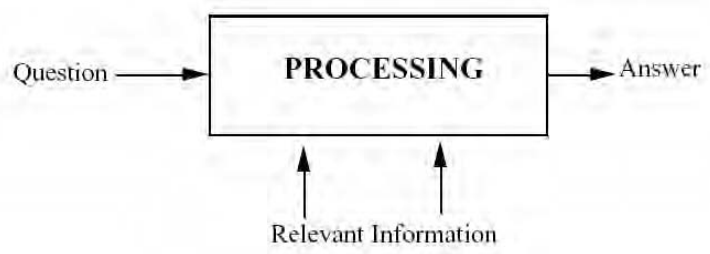



**Figure 1.** Task of answering a question.

The process of answering a question about the attribute involves a combining of this information. In some cases, this process may involve a fusing of the available information to obtain an effective value for the variable. Then an answer to the question of interest is obtained with respect to this fused information. In some cases, the answer to the question may be obtained using a process that doesn't depend upon obtaining an effective value.

## 3. Basic Knowledge Representation Using Fuzzy Sets

Among the central tasks involved in providing an answer to a question is the representation of the relevant information in a manner that allows formal manipulation. The representational language should be rich enough to allow the modeling of different types of information. Fuzzy subsets provide the basis for a very expressive framework for the representation of a wide body of knowledge. This knowledge can be either precise or imprecise. It can be used to represent knowledge expressed using linguistic values. Here, we shall briefly discuss this representational capability, however, we note the extensive literature on this, especially the work of Zadeh under his paradigm of computing with words [4,5] and the related theory of approximate reasoning [6–10].

Within the framework provided by fuzzy sets, knowledge about the value of a variable V is expressed using a statement V *is* A where A is a fuzzy subset of the domain X. The use of this type of representation can be seen as a generalization of the idea of imposing a constraint on the value of V, such as saying that V lies in the subset B, when B is a crisp subset of X. An example of this is saying John's age is between 25 and 35. The use of fuzzy subsets allows for a grading of this concept of V lying in the set B. Hence, the statement V *is* A manifests a constraint on the value of the variable V. The assignment of a fuzzy subset A to the variable induces a possibility distribution on X such that A(x) indicates the possibility that x is the value of V.

These types of fuzzy assignments can arise in, although are not restricted to, situations in which the information about the value of the variable is initially expressed in linguistic terms. An example of this would be the observation that John is middle aged. In this case, the fuzzy subset is the representation of the linguistic term middle-age. Here, the definition of the fuzzy subset A is such that for $x \in X$ the membership grade A(x) is the compatibility of the age x with the concept being represented, middle-age. We should note that while the use of a crisp subset allows for a representation of uncertainty of value but the use of fuzzy subsets allows for a more sophisticated representation as it makes more than just a simple distinction between those values that are possible and those that are impossible – it allows a grading of possibility.

We note that, in the case where A = {x}, then the statement V *is* A is equivalent to saying that V = x. Another special case is when A = X. Here, the statement V *is* X is equivalent to saying that we don't know. If B is some crisp subset of X, then the statement V *is* B is equivalent to saying the value of V lies in B. The situation when $A = \varnothing$, the null set, corresponds to the case where we are saying our knowledge is that V is not in X. This situation indicates a complete conflict with our assumption that V must take its value in X. More generally, if A is such that $\text{Max}_X A(x) < 1$ then we have some degree of conflict with the assumption that V has X as its domain. We shall say a fuzzy subset is normal if there exists at least one $x \in X$ so that A(x) = 1. If $\text{Max}_X A(x) < 1$ we say A is subnormal.

Consider the situation where we have the knowledge that V lies in B, V *is* B, where B is the crisp subset X. From this, we can naturally infer that V lies in E where $B \subseteq E$, here E is any set containing B. Thus, knowing that John's age is between 25 and 35 allows us to infer that John's age is between 10 and 50. In the fuzzy framework, that generalizes to what is called **the entailment principle** [11]. This principle states that, from the knowledge that V *is* A, we can infer V *is* F where $A \subseteq F$. We recall that for fuzzy subsets $A \subseteq F$ if $A(x) \leq F(x)$ for all x.

Clearly, the knowledge that V is contained in [25,35] is more informative and less uncertain than the knowledge that V is contained in [10,50]. Furthermore the statement that V is 25 is even more informative than either of the preceding as it contains no uncertainty. In [12–14], we introduced the concept of specificity to measure the amount of information contained in a fuzzy proportion V *is* A. Specificity is inversely related to the idea of uncertainty, the more specific the more certain our knowledge.

**Definition[1]:** Assume A is a fuzzy subset over X. Let $x^*$ be such that $A(x^*) = \text{Max}_X[A(x)]$, it is an element having the maximal membership grade in A. Let $\hat{A}$ be the average membership grade of A over the space $X - \{x^*\}$, it is the average over all elements except $x^*$. The specificity of A, denoted Sp(A) is defined as $Sp(A) = A(x^*) - \hat{A}$, it is the difference between the highest membership grade and the average of all the other elements.

**Note:** If more than one element attains the highest membership grade then all except one of these are used to find the average.

**Note:** We note the specificity of the statement V *is* A is equal to Sp(A). Thus we use the terms Sp(V *is* A) interchangeably with Sp(A).

We can observe some properties of Sp(A):

1. it lies in unit interval $0 \leq Sp(A) \leq 1$;
2. $Sp(A) = 1$ iff there exists one element $x^*$ such that $A(x^*) = 1$ and all other elements have $A(x) = 0$;
3. if $A(x) = c$ for all x, then $Sp(A) = 0$;
4. if A and B are two normal fuzzy subsets, they have one element with membership grade and $A \supseteq B$ then $Sp(B) \geq Sp(A)$. Thus, containment in the case of normality means an increase of specificity.

**Note:** Essentially, specificity measures the degree to which V *is* A points to one and only one element as the value of V.

As we shall subsequently see, the measure of specificity can play an important role in the processing of information. Consider the statement V *is* A where A is a normal fuzzy subset, that is there exists at least one element that has full possibility of having the value of V. We earlier noted that if $B_1$ is such that $B_1 \subset A$ as well as remaining normal then V *is* $B_1$ provides more information about the value of V than the original statement V *is* A. Essentially in this case we introduced some degree of clarity, we reduced the uncertainty by reducing the possibility of some elements while still leaving the possibility of finding a solution. On the other hand, if $B_2 \supset A$ then V *is* $B_2$ provides less information than V *is* A. In this case, we have reduced our certainty because we have added more possibilities. A third situation is where we have V *is* $B_3$ but with $B_3 \subset A$ but with $B_3$ subnormal $\text{Max}_X[B_3(x)] < 1$. We don't have a solution completely compatible with the assumption that V lies in X. In this case, we can possibly have less information than the original statement V *is* A, $Sp(B_3) \leq Sp(A)$. More generally, we observe that a reduction of specificity (certainty) in our knowledge can come about from two sources, one being increased possibility and the other being an increase in conflict with the assumption that its value lies in the given domain.

## 4. On the Measures of Possibility and Certainty

As we earlier noted, a task of great interest is the answering of a question about some variable. Here, we shall concern ourselves with this issue when our knowledge about the variable as well as the question is represented using the preceding representation. Here then given the knowledge that V *is* A our task is the determination of the validity of the statement V *is* B.

In order to build our intuition, we shall initially consider the case in which the sets A and B are crisp sets. There are two situations regarding our knowledge of A. In the first, we have no uncertainty regarding our knowledge of V, $V = x_1$, here $A = \{x_1\}$. In this situation, we can very clearly answer our question about the truth of the statement V *is* B. If $x_1 \in B$ then the answer is yes, if $x_1 \notin B$ then the answer is no. This exact information with respect to the value of V leads to precise answers.

The second case is where A is not a singleton, there exists some uncertainty about the value of V. This is the more typical situation, as noted by Kay [1], in intelligence analysis. The uncertainty associated with the knowledge that V *is* A makes the clear determination of whether another statement V *is* B is true or false not always attainable. Using figure 2 can help us understand the situation when A is uncertain.

We see in case 1, knowing that V *is* A assures us that V *is* B is valid. In case 2, knowing that V *is* A assures us that V *is* B is not true. Finally, in case 3, we can't tell. Thus we observe from this crisp environment that we have the following rules regarding the determination of truth of the statement V *is* B given V *is* A:

If $A \subseteq B$ then the answer is yes
If $A \cap B = \emptyset$ then the answer is no
If $A \cap B \emptyset$ and $A \not\subset B$ then the answer is I don't know

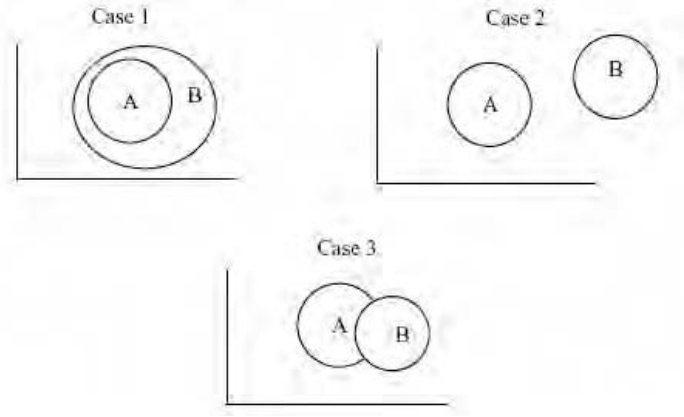



**Figure 2.** Different relations between knowledge and question.

Thus the attainment of a clear answer to questions, in the face of uncertainty in our knowledge, is not always attainable. We note this is a situation that holds even in the special case when B is a singleton. We see that asking if V = 30 if we only know that $V \in [25,40]$ can't be answered by yes or no, the appropriate answer is *I don't know*.

In the fuzzy set environment more sophisticated tools are needed to address this problem. Two measures have been introduced by Zadeh [3] to help. These are the

measures of possibility and certainty. We note that Dubois and Prade [15,16] refer to the measure of certainty as the measure of necessity. In the following, we shall, unless otherwise stated, assume A and B are normal.

The possibility that V *is* B given V *is* A is denoted by Poss[V *is* B/V *is* A] and is defined as Poss[V *is* B / V *is* A] = $\text{Max}_X[D(x)]$ where D(x) = Min[A(x), B(x)]. Thus

$$\text{Poss}[V \textit{ is } B / V \textit{ is } A] = \text{Max}_X[A(x) \wedge B(x)]$$

Since D = A $\cap$ B, we see that Poss[V *is* B / V *is* A] is the maximum degree of intersection between A and B.

The second measure introduced by Zadeh is the measure of certainty. We define this as

$$\text{Cert}[V \textit{ is } B / V \textit{ is } A] = 1 - \text{Poss}[V \textit{ is not } B / V \textit{ is } A] = 1 - \text{Max}_X[A(x) \wedge \overline{B}(x)]$$

With some manipulation we attain

$$\text{Cert}[V \textit{ is } B / V \textit{ is } A] = \text{Min}_X[\overline{A}(x) \vee B(x)]$$

We observe that Cert[V *is* B / V *is* A] is indicating the degree to which A is contained in B. That is if A is contained in B the knowledge that V is in A assures us that it is in B.

These measures of possibility and certainty can be seen as respectively providing upper and lower (optimistic and pessimistic) bounds, on the answer to the question of whether V *is* B is true given we know that V *is* A.

We note that if A is a normal fuzzy subset, there exists an $x^*$ such that $A(x^*) = 1$, then Poss[V *is* B / V *is* A] $\geq B(x^*)$ and Cert[V *is* B / V *is* A] $\leq B(x^*)$. Thus we see when A is normal we have Cert[V *is* B/V *is* A] $\leq$ Poss[V *is* B / V *is* A].

Let us look at these measures for some special cases of A and B. We first consider the case when A and B are crisp. In this case, Cert[B/A] and Poss[B/A] must be either one of zero. We see that if Cert[B/A] = 1 then Poss[B/A] = 1 and this corresponds to the case where V *is* B is true. If Poss[B/A] = 0 then Cert[B/A] = 0 and we know that V *is* B is false. If Cert[B/A] = 0 while Poss[B/A] = 1 then we are in the situation in which the answer is unknown.

Consider the situation where B is a crisp subset and A can be fuzzy. Here we have that

$$\text{Poss}[V \textit{ is } B / V \textit{ is } A] = \text{Max}_{X \in B}[A(x)]$$

$$\text{Cert}[V \textit{ is } B / V \textit{ is } A] = \text{Min}_{X \notin B}[\overline{A}(x)] = 1 - \text{Max}_{X \notin B}[A(x)]$$

An important special case of this is where B = $\{x^*\}$, here we are interested in determining whether V is equal to some particular value. In this case we see that Poss[V *is* B / V *is* A] = $A(x^*)$ and Cert[V *is* B / V *is* A] = $\text{Min}_{X \neq X^*}[\overline{A}(x)] = 1 - \text{Max}_{X \neq X^*}[A(x)]$. The certainty is the negation of largest possibility of value not equal

to x*. We also observe that if $A(x^*) \neq 1$ then we must have Cert[V *is* $x^*$ / V *is* A] = 0. This follows since with normal there exists some element $x_1 \neq x^*$ with $A(x_1) = 1$ and hence $1 - \text{Max}_{X \neq X^*}[A(x)] = 0$.

We also observe in the case where $X = \{x_1, x_2\}$ if we ask is $V = x_1$, we see that Cert[V *is* $x_1$ / V *is* A] = $1 - A(x_2)$. It is simply the negation of the possibility of the other element.

Consider now the special case where A is a crisp set. Here

$$\text{Poss}[V \textit{ is } B / V \textit{ is } A] = \text{Max}_{x \in A}[B(x)]$$

$$\text{Cert}[V \textit{ is } B / V \textit{ is } A] = \text{Min}_x[\overline{A}(x) \vee B(x)] = \text{Min}_{x \in A}[B(x)].$$

If additionally we assume that $A = \{x_1\}$, the value of V is exactly known, then

$$\text{Poss}[V \textit{ is } B / V \textit{ is } A] = B(x_1) \text{ and } \text{Cert}[V \textit{ is } B / V \textit{ is } A] = B(x_1)$$

Here then $B(x_1)$ is the validity of the statement that V *is* B.

Some clarification may be useful. What we have shown is that generally when our information about a variable, V *is* A, has some uncertainty, the answer to any question about the truth of the statement V *is* B lies in some interval. Thus if A is not a singleton the truth of V *is* B lies in the interval [l, u] where l is the certainty of V *is* B and u is the possibility that V *is* B. Here [l, u] is a subset of the unit interval. On the other hand if A is a singleton then the truth of V *is* B is a precise value b, in the unit interval. If additionally in the case when A is a singleton we have that B is a crisp set then b equals one or zero. The important point here is that there are two manifestations of lack of discreteness. One being as a result of our lack of certainty regarding the knowledge of A, it is granular and it is not a singleton, this generally results in an interval for our truth value, granular truth value. The second issue is related to a lack of crispness, the sets involved are fuzzy, this generally introduces aspects of multi-values logic, the values l, u and b are not necessarily one or zero but can be anywhere in the unit interval.

In cases where the decision process requires a more precise determination of the validity of the proposition V *is* B then provided by the interval [u, l] we must provide some means around this difficulty. However, we must emphasize that the actual processing of the information about the variable V has left us with some uncertainty. In some cases, we may be able to draw upon techniques from decision-making under uncertainty [17] to help make decisions in this kind of environment. First we want to make a clear distinction between the analyst, such as an intelligence analyst, and the want we shall call executive. It is the executive who makes the decision using as **some** of his input the information provided the analyst. While it is not our purpose here to go into great detail about the executive task of decision making, as we are more interested in the analysis task, we shall make a few comments.

In making a decision, such as whether we should preemptively strike an adversary, in addition to the information provided by the analyst about the state of V which may be uncertain, such as whether they have weapons of mass destruction, an executive generally draws upon two other types of information [18]. The first type of information is related to the costs or payoffs associated with the choice of an action and possible states of the uncertain variable V. Formally this is often expressed using a payoff matrix. The second type of information is related to what we call the decision makers'

attitudinal character [19]. This component of the decision process has an extremely subjective nature. It is here that people can have strong differences of opinion, which are purely value and preference driven. Thus one executive in the face of an uncertainty regarding the relevant variable may decide to act in a way that defends against the worst possibility, the so-called Max-Min decision maker [20]. Given an appropriate use of this with respect to the available knowledge of the possible states outcomes this type of decision cannot be said to be right or wrong. The point we want to make here is that uncertainty in our knowledge provides space for the inclusion of subject choices by the executive making the decision. A simple example of this may involve preparing for a party in which we are not sure whether 20 or 500 people are coming. Clearly if we prepare for 500 and only 20 show up then we wasted a lot of money. On the other hand, if we prepare for 20 and 500 show up we have some embarrassment. The choice of how many people to prepare for is based on the subjective preferences of the party giver, the executive, with regard to being embarrassed or wasting money, there is no right or wrong.

In the preceding, we assumed normality with respect to all the sets involved, all sets were assumed to have at least one element with membership grade 1. Here, we shall make some comments about the situation with respect to sub-normality, $\text{Max}_X[A(x)] < 1$. First, we note sub-normality is generally a reflection of some conflict. Sub-normality usually arises from the combination of information from different sources when there is some conflict between the observations of the sources. A second way it can arise is when the information provided by an individual source is in conflict with the assumption about the domain of a variable. This type of situation occurs less frequently. Thus we shall assume that our primary information supplied by the individual sources is normal.

In formal reasoning systems based on logic, the appearance of conflicting statements results in a situation in which we can infer anything, we conclude that everything is true. Our system has a similar property. Assume V *is* A and $A = \emptyset$ then for any statement V *is* B we have $\text{Cert}[V \textit{ is } B / V \textit{ is } A] = \text{Min}_X[\tilde{A}(x) \vee B(x)] = 1$. Thus, in the face of complete conflict everything is certain. On the other hand, with $A = \emptyset$, $\text{Poss}[V \textit{ is } B / V \textit{ is } A] = \text{Max}[A(x) \wedge B(x)] = 0$. Thus nothing is possible but everything is certain. In order to avoid this difficulty of having the certainty greater then the possibility we shall use as our definition of certainty

$$\text{Cert}[V \textit{ is } B / V \textit{ is } A] = (\text{Min}_X[\tilde{A}(x) \vee B(x)]) \wedge (\text{Max}_X[A(x)])$$

If A is normal this just is the definition for certainty we previously used $\text{Cert}[V \textit{ is } B/V \textit{ is } A] = \text{Min}_X[\tilde{A}(x) \vee B(x)]$. While if $A = \emptyset$ we get $\text{Cert}[V \textit{ is } B / V \textit{ is } A] = 0$. More generally, using this definition we always get $\text{Cert}[V \textit{ is } B / V \textit{ is } A] \leq \text{Poss}[V \textit{ is } B / V \textit{ is } A]$.

One further comment is in order with respect to normality. Previously, we defined the entailment principle as saying from V *is* A we can infer V *is* B where $A \subseteq B$. This must be modified to say that B must satisfy the $\text{Max}_X[B(x)] \leq \text{Max}_X[A(x)]$. Thus if A is normal no additional restriction exists on B, on the other hand if $\text{Max}_X[A(x)] = a$ then any statement V *is* B inferred from this must satisfy both $A \subseteq B$ and $\text{Max}_X[B(x)] \leq a$. The inferred set B can't be more possible than the original set A.

## 5. Hedging on Our Data

In the preceding, we introduced V *is* A as a structure for representing uncertain knowledge where A is a fuzzy subset of the domain X of V. We indicated that this generalized the idea of knowing that V lies in some subset. More generally, this formulation imposes some constraint on the value that V can assume. One question we considered was determining whether the proposition V *is* $x^*$ is valid given the knowledge V *is* A. We showed that, with uncertainty in our knowledge about V, the best we could do was to put some bounds on the truth of the hypothesis that V *is* $x^*$. In particular, Poss[V *is* $x^*$ / V *is* A] = A($x^*$) provided an upper bound and Cert[V *is* $x^*$ / V *is* A] = $\text{Min}_{X \neq x^*}[\overline{A}\ (x)] = 1 - \text{Max}_{X \neq x^*}[A(x)]$ provided a lower bound. If we let B = {$x^*$} then **not x*** is $\overline{B}$=X – {$x^*$} and we see that Cert[V *is* $x^*$ / V *is* A] = 1 – Poss[V *is* <u>not</u> $x^*$ / V *is* A].

We now consider the situation where we want to hedge on the knowledge that V *is* A. We let $\alpha \in [0, 1]$ indicate the degree of confidence we attribute to the proposition V *is* A, that is our knowledge V *is* A is $\alpha$ certain. In [21] it was suggested that one can express this hedged knowledge as a proposition V *is* F where F(x) = $\text{Max}[A(x), \overline{\alpha}] = A(x) \vee \overline{\alpha}$. Since $\overline{\alpha} = 1 - \alpha$ we see if $\alpha = 1$ then F(x) = A(x) and we get our original unhedged proposition. If $\alpha = 0$ then $\overline{\alpha} = 1$ and F(x) = 1 for all x. Here, our statement V *is* F effectively carries no information. Essentially this hedging loosens the constraint on the variable V.

In the following we shall let $A^*$ denote the fuzzy set such that $A^*(x) = 1$ if $x = x^*$ and $A^*(x) = 0$ if $x \neq x^*$.

Let us see what happens to our measures of possibility and certainty in this hedged situation

$$\text{Poss}[V \textit{ is } x^* / V \textit{ is } A \text{ is } \alpha\text{–cert}] = \text{Max}_X[(A(x) \vee \overline{\alpha}]) \wedge A^*(x)] = A(x^*) \vee \overline{\alpha}$$

In the case of certainty we have

$$\text{Cert}[V \textit{ is } x^* / V \textit{ is } A \text{ is } \alpha\text{–cert}] = 1 - \text{Max}_{X \neq x^*}[F(x)] = \text{Min}_{X \neq x^*}[\overline{F}\,(x)].$$

Since $F(x) = \overline{\alpha} \vee A(x)$ then Cert[V *is* $x^*$ / V *is* A is $\alpha$–cert] = $1 - \text{Max}_{X \neq x^*}[\overline{\alpha} \vee A(x)] = 1 - \overline{\alpha} \vee \text{Max}_{X \neq x^*}[A(x)] = \alpha \wedge \text{Min}_{X \neq x^*}[\overline{A}\,(x)] = \text{Min}[\alpha, \text{Min}_{X \neq x^*}[\overline{A}\,(x)]]$. More intuitively we see that Cert[V *is* $x^*$ / V *is* A is $\alpha$–cert] = Min[$\alpha$, Cert[V *is* $x^*$ / V *is* A]], it is the smaller of $\alpha$ and the certainty of the unhedged situation.

In anticipation of what we shall do in the following, we shall refer to these as optimistic and pessimistic measures

$$\textbf{Opt}(V \textit{ is } x^* / V \textit{ is } A \text{ is } \alpha\text{–cert}) = \text{Poss}[V \textit{ is } x^* / V \textit{ is } A \text{ is } \alpha\text{–cert}] = A(x^*) \vee \overline{\alpha}$$

$$\textbf{Pess}(V \textit{ is } x^* / V \textit{ is } A \text{ is } \alpha\text{–cert}) = \text{Cert}[V \textit{ is } x^* / V \textit{ is } A \text{ is } \alpha\text{–cert}]$$
$$= \alpha \wedge \text{Min}_{X = x^*}[\overline{A}(x)]$$
$$= \alpha \wedge (1 - \text{Poss}[V \textit{ is not } x^* / V \textit{ is } A])$$

We now consider an alternative method for representing a certainty quantified statements using the Dempster-Shafer belief structure [22]. Here we represent the statement V *is* A is α-cert by the proposition V *is* m where m is a D-S belief structure with two focal elements, $B_1 = A$ and $B_2 = X$ having $m(B_1) = \alpha$ and $m(B_2) = 1 - \alpha$ . In this framework, we use the plausibility and belief measure to obtain our optimistic and pessimistic bounds on the validity of the statement V *is* x*. We recall the plausibility and belief measures are respectively the expected possibility and expected certainty.

$$\text{Pl}[V \textit{ is } x^* / V \textit{ is } m] = \sum_{i=1}^{2} m(B_i)\, \text{Poss}[V \textit{ is } x^* / V \textit{ is } B_i]$$
$$= \alpha\, \text{Poss}[V \textit{ is } x^* / V \textit{ is } A] + \overline{\alpha}\ \text{Poss}[V \textit{ is } x^* / V \textit{ is } X]$$
$$= \alpha\, A(x^*) + \overline{\alpha} = 1 - \alpha\, \overline{A}(x^*)$$

$$\text{Bel}[V \textit{ is } x^* / V \textit{ is } m] = \sum_{i=1}^{2} m(B_i)\text{Cert}[V \textit{ is } x^* / V \textit{ is } B_i]$$
$$= \alpha\, \text{Cert}[V \textit{ is } x^* / V \textit{ is } A] + \overline{\alpha}\, \text{Cert}\,[V \textit{ is } x^* / V \textit{ is } X]$$
$$= \alpha\, \text{Min}_{X \neq x^*}[\overline{A}(x)] + \overline{\alpha}\ 0 = \alpha\, \text{Min}_{X \neq x^*}[\overline{A}(x)]$$

We observe that the more generally pessimistic measures can be generalized using a t-norm [23]. Thus if T is any t-norm then

$$\textbf{Pess}[V \textit{ is } x^* / V \textit{ is } A \text{ is } \alpha\text{–cert}] = T[\alpha, \text{Min}_{X \neq x^*}[\overline{A}(x)].$$

The optimistic measure can be generalized using a t-conorm [23]. Thus if S is any t-conorm then

$$\textbf{Opt}[V \textit{ is } x^* / V \textit{ is } A \text{ is } \alpha\text{–cert}] = S(\overline{\alpha}, A(x^*))$$

We shall not here investigate the issues involved in selecting among these possibilities.

## 6. Multi-Source Information Fusion

We now turn to the issue of aggregation of information from multiple sources, **M**ulti-**S**ource **I**nformation **F**usion (M-SIF).

If V *is* A and V *is* B are two pieces of information then their conjunction (fusion) is V *is* D where $D = A \cap B$, that is $D(x) = \text{Min}[A(x), B(x)]$. More generally, if V *is* $A_i$, for i = 1 to q, are a collection of propositions[2] from multiple sources then their conjunc-

tion is V *is* D where $D = \bigcap_{i=1}^{q} A_i$ here $D(x) = \text{Min}_i[A_i(x)]$. We observe one fundamental feature of this conjunction process. For all x, $D(x) \leq A_i(x)$ that is $D \subseteq A_i$. More generally, if $D = \bigcap_{i=1}^{q} A_i$ and $E = D \cap A_{q+1}$ then $E \subseteq D$, $E(x) \leq D(x)$ for all x. Thus we see the more information we get the smaller the fuzzy subsets.

In using multiple sources of information, usually, our objective is to increase the amount of information we have about the variable of interest. We desire to increase the specificity. We observe that if $D = \bigcap_{i=1}^{q} A_i$, is normal then $Sp(D) \geq Sp(A_i)$ for any i and we have gained information. Thus, here if the information is not conflicting then fusing the information supplied by the multiple sources is a process which can't decrease the information we have from any of the individual sources. Normally, in this case, D usually is more informative than any of the individual sources.

However, if some of the source information is conflicting, this may result in a situation in which D is subnormal, $\text{Max}_x D(x) \leq 1$. In this case, the fusion of the sources may provide us with a situation in which we are more confused and our informativeness, specificity, has decreased. In general it is difficult dealing with situations in which we have conflicting source information. One approach to addressing this situation is not to use all the information. That is, we selectively choose which information to use and fuse. This requires adjudicating between the information supplied by the different sources. Often the choice of the appropriate manner of adjudication requires the use of subjective considerations on the part of the person ultimately responsible for fusing the information. In the following, we shall suggest one approach to addressing this problem. We should note that other approaches are possible.

As we shall subsequently see, this process generally involves a tradeoff between selecting a subset of the available information that is not conflicting and yet large enough to provide a credible fusion of the available information. The technique we shall suggest will make use of the concept of a credibility measure to help in this process.

Let $P_i$ denote V *is* $A_i$, a be piece of data about the variable V. We refer to the collection of these as $\mathbf{P} = \{P_1, ..., P_q\}$. We associate with $\mathbf{P}$ a measure $\mu: 2^{\mathbf{P}} \rightarrow [0, 1]$ such that for each subset B of $\mathbf{P}$, $\mu(B)$ indicates the credibility of using as our fused knowledge the conjunction of the data in B. We shall call $\mu$ the credibility measure. We can associate with $\mu$ some basic properties: $\mu(\varnothing) = 0$ and $\mu(\mathbf{P}) = 1$. Additionally $\mu$ must be monotonic, if $B_1 \subset B_2$ then $\mu(B_2) \geq \mu(B_1)$.

Assume B is a subset of $\mathbf{P}$. Let $D_B = \bigcap_{P_i \in B} A_i$, it is the fusion of the knowledge in B. We observe using the subset B leads to the statement V *is* $D_B$. However, any statement obtained by using only the information in B only has a credibility of $\mu(B)$.

In order to determine the quality of the knowledge obtained by using the subset B we must consider two criteria. One criteria is that the knowledge provided fusing the data in B is informative and the other criterion is that B is credible. The degree of satisfaction to the criteria of informativeness, Inf(B), can be obtained using the measure of

specificity, thus Inf(B) = Sp($D_B$). We recall Sp($D_B$) = $D_B(x^*) - \underset{X-\{x^*\}}{Ave}(D_B)$ where $x^*$ is any element having maximal membership grade in $D_B$. The credibility of using the subset B, Cred(B), can be measured by μ(B). Since our measure of quality is an *anding* of these criteria we can define the measure of the quality of the result obtained using the subset B as Qual(B) = Inf(B) Cred(B), thus Qual(B) = Sp($D_B$) μ(B).

An interesting alternative view of our measure Qual(B) can be obtained. In the preceding, we indicated that a statement such as V *is* $D_B$ is μ(B)-cert as being translated into V *is* F where $F(x) = D_B(x) \vee (1 - \mu(B))$. We note that ∨ is an example of a t-conorm. It is the Max t-conorm, S(a, b) = Max(a, b) = a ∨ b. Let us consider the use of another t-conorm. A particularly interesting one is S(a, b) = a + b – ab, this is called the bounded sum [24]. If we use this instead of the Max we get $F_B(x) = D_B(x) + (1 - \mu(B)) - (D_B(x)(1 - \mu(B)))$. After a little algebra we get

$$F_B(x) = (1 - \mu(B)) + \mu(B)D_B(x)$$

We note that it is monotonic with respect to $D_B(x)$, if $D_B(x_1) \geq D_B(x_2)$ then $F_B(x_1) \geq F_B(x_2)$.

Consider now the measure of specificity. We first recall Sp($D_B$) = $D_B(x^*) - \underset{X-\{x^*\}}{Ave}(D_B)$. Consider now Sp($F_B$) here since $x^*$ still provides the largest membership in $F_B$ then

$$Sp(F_B) = F_B(x^*) - \underset{X-\{x^*\}}{Ave}(F_B)$$

$$Sp(F_B) = ((1 - \mu(B)) + \mu(B)D_B(x^*)) - \underset{x \neq x^*}{Ave}[(1 - \mu(B)) + \mu(B)D_B(x)]$$

The nature of Ave is such that

$$\underset{x \neq x^*}{Ave}[(1 - \mu(B)) + \mu(B)D_B(x)] = (1 - \mu(B)) + \mu(B)\underset{X-\{x^*\}}{Ave}(D_B)$$

Thus here

$$Sp(F_B) = ((1 - \mu(B)) + \mu(B)D_B(x^*)) - (1 - \mu(B)) - \mu(B)\underset{X-\{x^*\}}{Ave}(D_B)$$

$$Sp(F_B) = \mu(B)(D_B(x^*) - \underset{X-\{x^*\}}{Ave}(D_B))$$

$$Sp(F_B) = \alpha Sp(D_B)$$

Thus using this definition for certainty qualification leads to a very nice result for the relationship between the specificities of $D_B$ and $F_B$. This can be of great use in finding the best solution to the fusion problem.

We make some observations about this process of multi source fusion. First, observe that if the whole collection of data $\mathbf{P}$ is such that $D_{\mathbf{P}}$ is normal then for all B since $D_{\mathbf{P}} \subseteq D_B$ then $D_B$ is also normal. Hence, in this case, $Sp(D_{\mathbf{P}}) \geq Sp(D_B)$. Furthermore, since $\mu(\mathbf{P}) = 1 \geq \mu(D_B)$ then

$$Qual(\mathbf{P}) = Sp(D_{\mathbf{P}}) \geq \mu(D_B)\, Sp(D_B) \geq Qual(B).$$

Thus, in the case where the fusion of the data from all the sources doesn't induce any conflict, the most informative thing to do is to use fusion of all the data in $\mathbf{P}$.

More generally, we make the following observation.

**Observation:** If $B_1$ is a subset of $\mathbf{P}$ such that $D_{B_1}$ is normal then for all subsets $B_2$ of $\mathbf{P}$ such that $B_2 \subset B_1$ then $Qual(B_1) \geq Qual(B_2)$.

**Justification:** Since $B_2 \subset B_1$ then $D_{B_1} \subseteq D_{B_2}$ and $\mu(B_1) \geq \mu(B_2)$. Since both are normal it follows that $Sp(D_{B_1}) \geq Sp(D_{B_1})$ and hence $Qual(B_1) \geq Qual(B_2)$.

**Definition:** We shall call a subset B where $D_B$ is a normal, non-conflicting subset. Furthermore, we call a subset B **maximally non-conflicting** if B is non-conflicting and the addition of any other piece of data to B results in sub-normal fusion.

**Observation:** Any subset B of data containing a maximally non-conflicting subset can't provide the best fusion.

We shall now consider some examples of the credibility measure. One special class of credibility measure are those we call cardinality based measures. For these measures no distinction is made between credibility in the different pieces of data, $\mu(B)$ just depends on how many pieces of data are in B, the cardinality of B. We can define a cardinality-based measure using a function $h:[0, 1] \rightarrow [0, 1]$ that satisfies $h(0) = 0$, $h(1) = 1$ and is monotonic, $h(r_1) \geq h(r_2)$ if $r_1 > r_2$. Using h we can define $\mu(B) = h(\frac{|B|}{|\mathbf{P}|})$. These types of functions are often obtained as a representation of some linguistic quantifier such as *most*, "*at least about half.*"

Another class of credibility measures are those that are completely additive. Here we associate with each piece of data $P_i$ a value $\alpha_i \in [0, 1]$ and assume $\sum_{i=1}^{q} \alpha_i = 1$. In this case $\mu(B) = \sum_{j \in B} \alpha_j$.

Let $G_k$, k = 1 to g, be a collection of subsets of $\mathbf{P}$ that provides a partition of $\mathbf{P}$. One example of credibility measure $\mu$ using this is one where $\mu(B) = 1$ if B contains at least a piece of data from each of the $G_k$ and $\mu(B) = 0$ otherwise. Closely related to this is a measure in which we associate with each $G_k$ a nonnegative value $g_k$ and define $\mu(B) = \sum_{k=1}^{g} g_k \frac{|B \cap G_k|}{|B|}$. Here we also assume the $g_k$ sum to one.

Another of type credibility measure is one that contains a crucial piece of data. We say that $P_j$ is crucial if $\mu(B) = 0$ if $P_j \notin B$.

Another interesting example of credibility measure is the following. Let $B_1$ be a subset of $\mathbf{P}$. Consider a measure such that $\mu(B) = 0$ if $B_1 \cap B \neq \emptyset$ and $B_1 \subseteq B$. This measure, which we can call a balanced measure, requires that if we include any data from $B_1$ in our fusion we must use all the data in $B_1$.

Let us summarize the procedure we suggested for providing a user with quality fusion of the data in the collection $\mathbf{P}$. The first step is to calculate the subset $B^*$ of $\mathbf{P}$ with the highest quality conjunction of its component data. That is we find $B^*$ such that $\text{Qual}(B^*) = \underset{B \subseteq P}{\text{Max}} [\text{Qual}(B)]$ where $\text{Qual}(B) = \text{Sp}(D_B)\ \mu(B)$. Having found this subset $B^*$ we indicate to the client that V *is* $D_{B^*}$ is the result of our multi-source data fusion and that the credibility of this information is $\mu(B^*)$.

## 7. Multiple Fused Values from Multi-Source Data

In some situations, the presentation of a single fused value may not be sufficient or appropriate. Here we shall suggest a process that will allow us to provide multiple fused values over the data set $\mathbf{P}$.

Our point of departure is again a collection of multi-source data $\mathbf{P} = \{P_1, ..., P_q\}$. Each piece of data $P_j$ being of the form V *is* $A_j$ where $A_j$ is a fuzzy subset of the domain of V, X. In addition, we have a credibility measure $\mu: 2^{\mathbf{P}} \rightarrow [0, 1]$ where $\mu(B)$ is the degree credibility assigned to a fusion using the data in the subset B of $\mathbf{P}$.

In the preceding, we defined a process for obtaining an optimal subset $B_1$ and which provided a fused value V *is* $D_{B_1}$ with credibility $\mu(B_1)$. Here $D_{B_1} = \bigcap_{j,\ P_j \in B_1} A_j$. This approach finds the subset of data $B_1$ such that that $\text{Qual}(B_1) = \mu(B_1)\ \text{Sp}(D_{B_1}) = \underset{B_1 \subseteq P}{\text{Max}} [\mu(B)\ \text{Sp}(D_B)]$. We shall refer to this process as **Qual-Fuse**($\mathbf{P}$, $\mu$). Thus Qual-Fuse($\mathbf{P}$, $\mu$) returns $B_1$ which enables the determination of $D_{B_1}$ and $\mu(B_1)$.

In the following, we shall suggest a procedure which allows the for generation of multiple fusions from the pair ($\mathbf{P}$, $\mu$). For notational convenience in the following we shall find it convenient to denote the fuzzy subsets $A_j$ as $A_j^1$, thus our data is still $\mathbf{P} = \{P_1, ..., P_q\}$ where $P_j$ corresponds to the observation V *is* $A_j^1$. $\mu$ is still a credibility measure over $\mathbf{P}$.

The basic algorithm of our procedure is as follows.

1. initialize our system with $\mathbf{P}$, $\mu$ and set i = 1.

2. apply Qual-Fuse($\mathbf{P}$, $\mu$) this returns $B_1$ and $D_{B_1}$ and V *is* $D_{B_1}$ with credibility $\mu(B_1)$.
3. revise each of the $P_j$ to V *is* $A_j^2$ where $A_j^2 = A_j^1 - D_{B_1}$. That is we remove the subset $D_{B_1}$ from the subset $A_j^1$. We recall $A_j^1 - D_{B_1} = A_j^1 \cap \overline{D}_{B_1}$ and therefore $A_j^2(x) = \text{Min}[A_j^1(x), 1 - D_{B_1}(x)]$
4. set i = 2
5. let $\mathbf{P} = [P_1, ..., P_1]$ with $P_j$ such that $V_j$ *is* $A_j^1$
6. apply Qual-Fuse($\mathbf{P}$, $\mu$). This returns $B_i$ and the statement V *is* $D_{B_i}$ with credibility $\mu(B_i)$. Here $D_{B_i} = \bigcap_{j,\, P_j \in B_i} A_j^1$
7. additional fusion desired? No – stop, Yes – continue
8. set i = i + 1
9. calculate $A_j^1 = A_j^{i-1} - D_{B_{i-1}}$
10. go to step 5.

The final result of this process is a collection of fused values of the form

V *is* $D_{B_1}$ with credibility $\mu(B_1)$

V *is* $D_{B_2}$ with credibility $\mu(B_2)$

V *is* $D_{B_k}$ with credibility $\mu(B_k)$

The key idea we suggested here is the removal of the already presented fused value from the data remaining to be fused. This is very much in the spirit of the Mountain Clustering method [25,26]. This removal process tends to result in a situation where the $D_{B_j}$ are disjoint.

An interesting issue, one which we shall not investigate in detail here, is when to stop the process of providing additional fusions. In its simplest form, this can just be based on an input from the user, for example how fused they want values to be. A more computationally based approach could be one in which we stop when the quality of the next proposed fusion falls below some level.

## 8. Fusing Probabilistic and Possibilistic Data

An important issue in the field of data fusion concerns itself with the combination of two pieces of information, where one is expressed in terms of a fuzzy subset (possibility distribution) and the other is expressed in terms of a probability distribution [27,28]. Here we shall introduce some ideas related to this problem.

Let G be an attribute which is associated with some class of objects Z. Let X be the domain in which this attribute takes its value. Our interest here is on the determination

of the value of the attribute G for some specific entity, z*, from this class. Thus we are interested in the determination of the value of variable $G(z^*)$. We shall denote this variable as $G_*$.

Consider a piece of data about $G_*$ such a $G_*$ *is* A which is a fuzzy subset of X. Let's look at this data more carefully. First, we see it is directly about the variable of interest. That is, it is a statement about the attribute for the object of interest. Often, this information is a result of some linguistically expressed observation such as "The bomb thrower was young." As noted by Zadeh [29] this statement puts some constraint on the possible values of the variable $G_*$. It generalizes the idea of having a more crisp statement such as "The age of the bomb thrower was between 18 and 25." It of course reflects some uncertainty with respect to the source's observation. In the situation in which A is assumed normal this uncertainty can be measured by the cardinality subset A, $\sum_X A(x)$. In the case where we must deal with subnormality more sophisticated measures such as Un(A) = 1 – Sp(A) = 1 – (Max(A) – Ave(A)) should be used. We see if Max(A) = 1 then Un(A) = Ave(A) which is essentially $\sum_X A(x)$.

Let us now turn to the situation in which we have additional probabilistic information consisting of a probability distribution P over the space X where $P(x_i)$ is the probability associated with the attribute value $x_i$. In order to find a basis for fusing these two pieces of information, the possibility distribution A and the probability distribution P, we shall take advantage of a view proposed by Coletti and Scozzafava [30]. In [30] the authors suggested that an element's membership grade in a fuzzy, $A(x_i)$, can be viewed as the conditional probability of A given $x_i$, $P(A/x_i) = A(x_i)$. Having this allows us to use Bayes' rule to generate the fused information. Let P(x/A, P) indicate the probability of x given two pieces of knowledge. In particular, $P(x/A, P) = \frac{P(A / x)}{P(A)} P(x)$.

Using P(A/x) = A(x) we have $P(x/A, P) = \frac{A(x)}{P(A)} P(x)$. Furthermore, since $P(A) = \sum_{i=1}^{n} P(\frac{A}{x_i}) P(x_i)$ then P(A) can be expressed $\sum_{i=1}^{n} A(x_i) P(x_i)$. Using this, we get $P(x/A, P) = \frac{A(x)P(x)}{\Sigma_i A(x_i) \cdot P(x_i)}$. At times we shall find it convenient to express this as $P(x/A, P) = \frac{A(x)}{\Sigma_i A(x_i) \cdot \frac{P(x_i)}{P(x)}}$.

Thus the result of fusing these two pieces of data is a probability distribution with respect to the value of $G_*$. Using the notation suggested by Zadeh in [29] we can express this as $G_*$ *isp* **R** where **R** indicates a probability distribution on X such that P(x/A, P), as defined above, is the probability that $G_*$ assumes the value x. The fact

that this is the case is not surprising since the knowledge in the possibility distribution is actually saying that the value of the variable $G_*$ lies in a set, A. So we are actually finding the probability of x conditioned on the knowledge that $G_*$ lies in a set.

Let us look at this for some special cases to see if it is consistent with our intuition. First, consider the case where $P(x_i) = \frac{1}{n}$.

Here, the probability distribution is essentially providing no information. In this case, we have $\frac{P(x_i)}{P(x)} = 1$ for all $x_i$ and hence $P(x/A, P) = \frac{A(x)}{\sum_{i=1}^{n} A(x_i)}$. Thus here we obtain P(x/A, P) as simply a normalization of the possible distribution.

Consider now the case in which $A(x_i) = 1$ for all $x_i$. Here the possible distribution is providing no information. In the case $P(x/A, P) = \frac{P(x)}{\sum_i P(x_i)} = P(x)$. We get back the original probability distribution.

Consider the case where A corresponds to some crisp subset B of X. That is $A(x_i) = 1$ for $x_i \in B$. In this case $P(x/A, P) = \frac{P(x)}{\sum_{x_i \in B} P(x_i)}$. This is the classic case of conditional probability.

One issue that must be addressed is conflicting information. Consider the case where we have $A(x_1) = 1$ and $A(x_j) = 0$ for all other $x_j$ and where $P(x_1) = 0$. In this case, we see that $P(x_i)A(x_i) = 0$ for all $x_i$ and our aggregation leads to a kind of indeterminism. Here, we essentially must decide, do we believe the possibility distribution which says the answer is definitely $x_1$ or do we believe the probability distribution which says the answer is definitely not $x_i$.

Another form of conflict can be seen in the following case. Let $A = \{\frac{1}{x_1}, \frac{0.1}{x_2}, \frac{0}{x_3}\}$ and let the probabilistic information be such that $P(x_1) = 0$, $P(x_2) = 0.1$ and $P(x_3) = 0.9$. In this case we obtain $P(x_1/A, P) = 0$, $P(x_2/A, P) = 1$ and $P(x_3/A, P) = 0$. This may be somewhat disturbing. Here, while both pieces of information lend little support to $x_2$ their combination leads to its strong support.

In order to address this issue of conflict we must first consider the context in which we obtain probabilistic information. We can envision two situations when we obtain probabilistic information. One of these is in a frequent spirit and the other is of a subjective kind.

One situation where we have probabilistic information is where the probability distribution is a reflection of some observation about the attribute G over the objects in the class Z. Thus here $P(x_j)$ is the probability that an object in Z has value for attribute G equal to $x_j$. For example, if $x_j = 26$, then P(26) is the probability that "a" bomb thrower is 26. The point we want to emphasize here is that this information is not directly about

the entity of interest $z^*$. It is not information about our variable of interest $G_{x^*}$, $G(x^*)$. Although it is useful and valuable information, it is not directly about the object of interest. The important observation here is that the information contained in this type of probabilistic information is of a lower priority than the direct information contained in a statement $G_*$ *is* A. Thus, here there is a priority ordering with respect to our information and in the face of conflict we want to give preference to the direct information, $G_*$ *is* A.

The use of a probabilistic representation can also occur in the case in which the source is providing information directly about the attribute value for the object of interest. Consider the situation where the source has some uncertainty with regard to the actual value of the variable $G_*$. Here, he uses the probability framework to express his perception of the uncertainty. He is saying that my feeling about the uncertainty associated with the value of $G_*$ is similar to that of a random experiment in which $P(x_i)$ is the probability that $G_* = x_i$. Again, in this situation, the information provided by the source is also less direct that that provided by the observation that $G_*$ *is* A.

The overall point we want to make here is that often the information provided using a probabilistic representation has a lesser priority than that provided using the fuzzy representation. This is not to say that fuzzy sets are better than probability but only that the type of information represented by a probability distribution is less directly relevant.

This distinction in the priority of the two different kinds of information allows us to provide a reformulation of the aggregation of these two kinds of information to allow for an intelligent adjudication of conflicts. As a first step in this process, we shall turn to the issue of measuring the conflict or conversely the consistency between a probability distribution and a possibility distribution.

Let $\Pi : X \rightarrow [0, 1]$ be a possibility distribution over the X, thus $\Pi(x_i)$ indicates the possibility of $x_i$. Here, we shall assume this is normal, there exists some $x^*$ such that $\Pi(x^*) = 1$. Let $P: X \rightarrow [0, 1]$ be a probability distribution over X. $P(x_i)$ indicates the probability of $x_i$. The probability distribution has the added requirement that $\sum_i P(x_i) = 1$. Let $p^* = \text{Max}_i[P(x_i)]$, it is the maximal probability associated with P. We can observe that $\frac{1}{n} \leq p^* \leq 1$, where n is the cardinality of X. It is well-known that the negation of the Shannon entropy, $\sum_i P(x_i) \ln[P(x_i)]$, provides a measure of information content of a probability distribution. What is worth pointing out is that the $\text{Max}_i(P(x_i))$ provides an alternative measure of this information content [31,32]. While the Shannon measure has some properties that make it preferred, especially when we consider multiple distributions, in the case when we are focusing on one probability distribution, $\text{Max}_i(P(x_i))$ provides a simple and acceptable measure of the information content of a probability distribution.

We now introduce a measure called the consistency of $\Pi$ and P

$$\text{Consist}(\Pi, P) = \text{Max}_i[\Pi(x_i) \wedge \tilde{P}(x_i)] \text{ where } \tilde{P}(x_i) = \frac{P(x_i)}{p^*}$$

We observe that if P is such that if $P(x_i) = \frac{1}{n}$ for all $x_i$ then $p^* = \frac{1}{n}$ and $\tilde{P}(x_i) = 1$ for all $x_i$. In this case both $\Pi(x^*) = 1$ and $\tilde{P}(x^*) = 1$ and hence Consist(Π, P) = 1. Thus, the situation when P has maximal uncertainty it is consistent with any possibility distribution. On the other hand we see that if $P(x_1) = 1$ and $\Pi(x_1) = 0$ then Consist(Π, P) = 0 they are in complete conflict. In the case where $X = \{x_1, x_2, x_3\}$ and $\Pi(x_1) = 1$, $\Pi(x_2) = 0.1$ and $\Pi(x_3) = 0$ while $P(x_1) = 0$, $P(x_2) = 0.1$ and $P(x_3) = 0.9$ we get $\tilde{P}(x_1) = 0$, $\tilde{P}(x_2) = 0.11$ and $\tilde{P}(x_3) = 1$ and hence Consist(Π, P) = 0.1.

We now consider the modification of the procedure for aggregating possibility and probability distributions which uses this measure of consistency to aid in the adjudication of conflicting information.

In the preceding, we defined the aggregation of V *is* A and the probability distribution P as inducing a probability distribution where $P(x/A, P) = \frac{A(x)P(x)}{\sum_{i=1}^{n} A(x_i)P(x_i)}$.

We now provide a modification of this to account for conflicts between the input distributions. As we shall see, this is will give a priority to the information V *is* A.

Letting $\alpha = \text{Consist}(P, A)$ we define

$$P(x/A, P) = \frac{A(x)[\alpha P(x) + \bar{\alpha}\frac{1}{n}]}{\sum_{j=1}^{n} A(x_j)[\alpha P(x_j) + \bar{\alpha}\frac{1}{n}]}$$

Let us see how this works. If the two sources are consistent, $\alpha = 1$, then

$$P(x/A, P) = \frac{A(x)P(x)}{\sum_{j=1}^{n} A(x_j)P(x_j)}$$

and we get our original formulation. If the two pieces of information are completely conflicting, $\alpha = 0$ we get

$$P(x/A, P) = \frac{A(x)\frac{1}{n}}{\sum_{j=1}^{n} A(\alpha_j)\frac{1}{n}} = \frac{A(x)}{\sum_{j=1}^{n} A(\alpha_j)}.$$

Here we completely discount the information contained in the probability distribution P and simply obtain P(x/A, P) as a normalization of A.

Here, we shall refer to $F(\alpha, P_j) = \alpha P(x_j) + \bar{\alpha}\frac{1}{n}$ as the probability transform and refer to $\lambda(x_j) = F(\alpha, P(x_j))$ as the transformed probabilities. We see that in the face of conflict the transformed probabilities move toward $\frac{1}{n}$.

We further observe that if $A(x_j) = 0$, then $P(x/A, P) = 0$.

**Example:** Assume $X = \{x_1, x_2, x_3\}$, $A = \{\frac{1}{x_1}, \frac{0.1}{x_2}, \frac{0}{x_3}\}$ and $P(x_1) = 0$, $P(x_2) = 0.1$ and $P(x_3) = 0.9$. Here we get $\tilde{P}(x_1) = 0$, $\tilde{P}(x_2) = 0.11$ and $\tilde{P}(x_3) = 1$ and hence $Consist(\Pi, P) = 0.1$. In this case the transformed probabilities are:

$$\lambda(x_1) = (0.9)\frac{1}{3} = 0.3$$

$$\lambda(x_2) = (0.1)(0.1) + (0.9)\frac{1}{3} = 0.31$$

$$\lambda(x_3) = (0.1)(0.9) + (0.9)\frac{1}{3} = 0.39$$

In this case $\sum_{i=1}^{3} A(x_i)\lambda(x_i) = 0.3 + 0.031 = 0.331$ and hence

$P(x_1/A, P) = \frac{0.3}{0.331} = 0.906$, $P(x_2/A, P) = \frac{0.031}{0.331} = 0.094$ and $P(x_3/A, P) = \frac{0}{0.331} = 0$.

We must consider one other issue here. We have implicitly assumed that the possibility distribution is normal, $Max_j(x_j) = 1$. If this is not the case some problems can arise. Since

$$Consist(A, P) = Max_j[A(x_j) \wedge \frac{P(x_j)}{p^*}] \leq Max_j[A(x_j)]$$

our maximal possible consistency goes down. Here the problems of reduced consistency may be an issue related to the internal conflict of the possibility distribution rather than its incompatibility with probability distribution.

It may be interesting to consider a slight modification in the case where we have $Max_j[A(x_j)] = a^* < 1$. Here, instead of the end result being a probabilistic distribution we end up with a Dempster-Shafer belief structure m. This belief structure has n + 1 focal elements $B_j = \{x_j\}$ for j = 1 to n and $B_{n+1} = X$. Furthermore for j = 1 to n we

have $m(B_j) = a^* P(x_j/A, P)$ where the $P(x_j/A, P)$ are calculated as in the preceding. For $B_{n+1} = X$ we have $m(X) = 1 - a^*$. We shall not pursue this but leave it as a suggestion.

## 9. Alternative Measures of Certainty

Here we consider a more technical issue which may not be of interest to all readers. We want to look a little more deeply at the issue of deciding whether some subset B of X contains the value V given that we know that V *is* A which is the basis our of definition of the measure Cert[V *is* B / V *is* A]. Here, we shall, unless otherwise stated, assume A is normal.

Our definition for the measure for Cert[V *is* B / V *is* A] = $\text{Min}_x[\overline{A}(x) \vee B(x)]$ is an extremely pessimistic measure. As we see in the crisp case as long as there is one element not in B that is possible, in A, it scores a value of zero. We see that if A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} then with B = {1} or B = {1, 2, ..., 9} we get the same degree of certainty, zero. Here, there exists no consideration about the cardinalities of how many elements in A are not in B, except that there exists one.

We observe that our definition of certainty is the degree of truth of the proposition:

"**All** elements not in B are not possible given A."

which we expressed as $\text{Min}_x[\tilde{A}(x) \vee B(x)]$. In the case where B is crisp this becomes

$$\text{Cert}[V \textit{ is } B / V \textit{ is } A] = \text{Min}_{x \in \bar{B}}[\tilde{A}(x)]$$

Prade and Yager [33] suggested a softening of the measure of certainty with the concept of expectedness. In [33] they introduced the idea of **expectedness** of V *is* B given V *is* A, denoted Exp[B/A] which they defined as a degree of truth of the proposition:

"**Most** of the elements not in B are not possible given A."

In the case where B is crisp we can express this as

$$\text{Exp}[V \textit{ is } B / V \textit{ is } A] = \text{Most}_{x \in \bar{B}}[\tilde{A}(x)]$$

We note that the difference between the two concepts, certainty and expectedness, is the respective uses of the terms all and most. We observe that these two terms are examples of what Zadeh called linguistic quantifiers [34]. Here, we shall suggest a parameterized formulation which leads to a generalization of these types of measures. Let Q indicate a general member of the class of regular monotonic linguistic quantifiers [35]. Using this we introduce the idea of what we shall denote as Q–Cert[[V *is* B / V *is* A] or more succinctly Q–Cert[B/A]. Specifically we define Q–Cert[B / A] as the truth of the statement

Q of the element not in B are not possible given A.

First, we note that as suggested by Zadeh, linguistic quantifier Q can be expressed as a fuzzy subset Q [0, 1] → [0, 1] where Q(r) indicates the degree to which the proportion r satisfies the concept Q. The fact that Q is a regular monotonic linguistic quantifier requires that Q satisfy the additional three conditions: Q(0) = 0, Q(1) = 1 and Q(x) ≥ Q(y) if x ≥ y.

We note some special cases of Q. The first is Q* where Q*(1) = 1 and Q*(x) = 0 for all x ≠ 1. This corresponds to the linguistic quantifier *all*. The second special case is Q* where Q*(0) = 0 and Q*(x) = 1 for all x ≠ 0. This corresponds to the quantifier *any*. Another special case is $Q_A$ where $Q_A$ (x) = x. It is suggested that this models the linguistic quantifier *some*. Furthermore it is suggested that $Q_A$ corresponds to the quantifier that is implicit when no quantifier is explicitly expressed, it is a kind of default quantifier.

We shall formally define the truth of the proposition Q of the element not in B are not possible given A, Q–Cert[V *is* B / V *is* A], using this importance weighted OWA operator [36]. We first recall this operator.

Let $(c_j, d_j)$ be a two tuple in which $c_j$ is called the importance and $d_j$ is called the argument value. We recall that the OWA aggregation of a collection of these tuples guided by a quantifier Q. $OWA_Q[(c_1, d_1), (c_2, d_2)..., ((c_n, d_n)]$, is defined as

$$OWA_Q[(c_1, d_1), (c_2, d_2)..., ((c_n, d_n)] = \sum_{j=1}^{n} w_j \, d_{\sigma(j)}$$

where σ(j) is the index of the $j^{th}$ largest of the $d_i$ and $w_j = Q(\frac{T_j}{T}) - Q(\frac{T_{j-1}}{T})$ where $T_j = \sum_{i=1}^{j} c_{\sigma(i)}$ and $T = \sum_{i=1}^{n} c_i$, the sum of all importances.

In the following we shall express Q–Cert[V *is* B / V *is* A] using this operation. For notational convenience we assume the domain of V, $X = \{x_1,..., x_n\}$ and $B(x_i) = b_i$ and $A(x_i) = a_i$. Using this notation Q–Cert[V *is* B / V *is* A] = $OWA_Q[(\overline{b}_i, \overline{a}_i)]$ where $\overline{b}_i = 1 - b_i$ and $\overline{a}_i = 1 - a_i$ thus

$$\text{Q–Cert[V } \textit{is} \text{ B / V } \textit{is} \text{ A]} = \sum_{j=1}^{n} w_j \, \overline{a}_{\sigma(j)}$$

where σ(j) is the index of the $j^{th}$ largest of the $_i$ and $w_j = Q(\frac{T_j}{T}) - Q(\frac{T_{j-1}}{T})$ where $T_j = \sum_{i=?}^{j} \overline{b}_{\sigma(j)}$ and $T = \sum_{k=?}^{n} \overline{b}_k$.

Let us consider the environment when B is crisp. Here $b_i = 1$ if $x_i \in B$ and $b_i = 0$ if $x_i \notin B$. Thus $\overline{b}_i = 0$ if $x_i \in B$ and $\overline{b}_i = 1$ if $x_i \notin B$. In this situation, we also observe if $b_{\sigma(j)} \in B$ then $\overline{b}_{\sigma(j)} = 0$ and since $T_j = \sum_{i=1}^{j} \overline{b}_{\sigma(j)}$ then in this case, $b_{\sigma(j)} \in B$, $T_j = T_{j-1}$ and hence $w_j = 0$. Thus we see that all terms that are in B have OWA weights equal to zero. Furthermore, for those elements not in B we have $\overline{b}_{\sigma(j)} = 1$ and $T_j$ is the number of elements up to and including the $j^{th}$ largest $\overline{a}_i$ that are not in B. Effectively for those $x_{\sigma(j)} \notin B$ we have $T_j = 1 + T_{j-1}$ and for those $x_{\sigma(j)} \in B$, we have $T_j = T_{j-1}$.

Thus in this case where B is crisp situation we can just consider those elements not lying in B. We shall let $\overline{n} = |\overline{B}|$ and let $\sigma_{\overline{B}}(j)$ be the index of the element having the $j^{th}$ largest value for $\overline{a}_i$ of those lying in $\overline{B}$. Then, in this case, B is crisp, we have

$$Q\text{–Cert}[B/A] = \sum_{j=1}^{\overline{n}} w_j \, \overline{a}_{\sigma_{\overline{B}}(j)}$$

where $w_j = Q(\frac{j}{|\overline{B}|}) - Q(\frac{j-1}{|\overline{B}|})$.

Let us consider the resulting formulations for some different examples of Q. If Q is $Q_*$ then $w_{\overline{n}} = 1$ and $w_j = 0$ for all other j and $Q_*$–Cert[B / A] = $\text{Min}_{x_i \notin B}[(\overline{A}\, x_i)]$. This was our original definition of Cert[B / A]. If we select $Q = Q^*$, then we get $Q^*$Cert[B / A] = $\text{Max}_{x_i \notin B}[(\overline{A}\, x_i)]$. This is what Dubois and Prade called the unguaranteed necessity. Another special case is where Q(x) = x. In this case we have $w_j = \frac{1}{n}$ and Q–Cert[V *is* B / V *is* A] = $\sum_{j \notin B}^{\overline{n}} \overline{A}_{(x_j)}$, it is the average of $\overline{A}$ $(x_j)$ for those $x_j$ not in B.

In the following, we shall look a useful family of Q–Cert[V *is* B / V *is* A] based on a class of quantifiers parameterized by a single scalar value λ. Consider the function $Q_\lambda$ shown in figure 3.

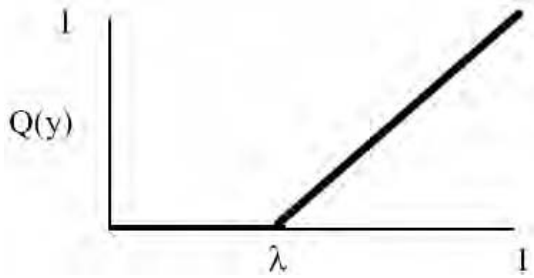



**Figure 3.** Quantifiers parameterized by β.

We shall denote λ as a strength of necessity. We easily see that when $\lambda = 1$ we get the strongest measure $Q_\lambda$–Cert[B / A] = $\text{Min}_{x_i \notin B}[(\overline{A}\, x_i)]$. When $\lambda = 0$ then we get $Q_\lambda$–Cert[B / A] = $\frac{1}{n}\sum_{j \notin B} \overline{A}\,(x_j)$. Generally we observe at as λ moves from 0 to 1, the value of $Q_\lambda$–Cert[B / A] decreases. If we impose the additional assumption that A is also crisp we add in developing a deeper intuitive understanding of the class of formulations for uncertainty we have introduced. In the case when $\lambda = 1$ to be certain of the truth of statement V *is* B we require if an outcome is possible, in A, it is also in B. In the case where $\lambda = 0$ we have

$$Q_\lambda\text{–Cert[B / A]} = \frac{1}{n}\sum_{j \notin B} \overline{A}\,(x_j) = 1 - \frac{1}{n}\sum_{j \notin B} A(x_j).$$

Here we take average possibility of the elements not in B and subtract that from one.

In this section we have described a family of definitions for the idea of the Certainty of V *is* B given V *is* A based on the parameter Q which we denoted Q–Cert[B / A]. By appropriately choosing the quantifier Q we can model the formulation we want to use for our concept of certainty. One important way in which these definitions for certainty differ is with respect to their strictness. Recalling that Q–Cert[B / A] is defined as the truth of the statement "Q of the element not in B are not possible given A" we see that the larger Q the stricter. In order to more formally capture this idea of strictness we can associate with any quantifier Q a value called its attitudinal character defined as

$$\text{A–C(Q)} = \int_0^1 Q(y)dy$$

It can be shown that A–C(Q) ∈ [0, 1]. Also we can show that for $Q = Q_*$ we get $\text{A–C}(Q_*) = 0$, for $Q = Q^*$ we get $\text{A–C}(Q^*) = 1$ and for Q(x) = x we get A–C(Q) = 0.5. Thus the smaller the A-C the stricter our concept of certainty.

## 10. Conclusion

We have discussed a number of tools to aid intelligence analysts in the representation and processing of uncertain information. The basic representation we use is that of fuzzy sets. We noted the difficulty of answering questions about the truth of some hypothesis in the face of uncertain information. We introduced the measures of possibility and certainty as a tool to enable an intelligence analyst to provide an answer in terms of an upper and lower bound on the truth of the hypothesis. We discussed methods for fusing information from multiple sources. A method for providing multiple fused values was introduced as well as a method for combining probabilistic and possibilistic information.

## Notes

1. While a number of different formal definitions have been suggested we shall find this one to be useful for our purposes as it simply captures the basic idea of the concept of specificity.
2. Here unless otherwise stared we shall assume the $A_i$ are normal, have at least one element with membership grade 1.

## References

[1] Risen, J., "Ex-inspector says C.I.A. missed disarray in Iraqi arms program," New York Times, January 27, 1, 2004.
[2] Zadeh, L.A., "Fuzzy sets," Information and Control 8, 338–353, 1965.
[3] Zadeh, L.A., "Fuzzy sets and information granularity," in Advances in Fuzzy Set Theory and Applications, edited by Gupta, M.M., Ragade, R.K. and Yager, R.R., North-Holland: Amsterdam, 3–18, 1979.
[4] Zadeh, L.A., "Fuzzy logic = computing with words," IEEE Transactions on Fuzzy Systems 4, 103–111, 1996.
[5] Zadeh, L.A. and Kacprzyk, J., Computing with Words in Information/Intelligent Systems 1, Physica-Verlag: Heidelberg, 1999.
[6] Zadeh, L.A., "A theory of approximate reasoning," in Machine Intelligence, Vol. 9, edited by Hayes, J., Michie, D. and Mikulich, L.I., Halstead Press: New York, 149–194, 1979.
[7] Dubois, D. and Prade, H., "Fuzzy sets in approximate reasoning Part I: Inference with possibility distributions," Fuzzy Sets and Systems 40, 143–202, 1991.
[8] Ruspini, E., "Approximate reasoning: past, present, future," Information Sciences 57–58, 297–317, 1991.
[9] Yager, R.R., "On the theory of approximate reasoning," Controle Automacao 4, 116–125, 1994.
[10] Yager, R.R., "Approximate reasoning as a basis for computing with words," in Computing with Words in Information/Intelligent Systems 1, edited by Zadeh, L.A. and Kacprzyk, J., Springer-Verlag: Heidelberg, 50–77, 1999.
[11] Yager, R.R., "The entailment principle for Dempster-Shafer granules," Int. J. of Intelligent Systems 1, 247–262, 1986.
[12] Yager, R.R., "Entropy and specificity in a mathematical theory of evidence," Int. J. of General Systems 9, 249–260, 1983.
[13] Yager, R.R., "Specificity measures of possibility distributions," Proceedings of the Tenth NAFIPS Meeting, U. of Missouri, Columbia, MO, 240–241, 1991.
[14] Yager, R.R., "On measures of specificity," in Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, edited by Kaynak, O., Zadeh, L.A., Turksen, B. and Rudas, I.J., Springer-Verlag: Berlin, 94–113, 1998.
[15] Dubois, D. and Prade, H., "Necessity measures and the resolution principle," IEEE Transactions on Systems, Man and Cybernetics 17, 474–478, 1987.
[16] Dubois, D. and Prade, H., Possibility Theory: An Approach to Computerized Processing of Uncertainty, Plenum Press: New York, 1988.

[17] Yager, R.R., "Decision making under various types of uncertainty," in Decision Making: Recent Developments and Worldwide Applications, edited by Zanakis, S.H., Doukidis, G. and Zopounidis, C., Kluwer Academic Publishers: Boston, 233–250, 2000.
[18] Yager, R.R., "Fuzzy modeling for intelligent decision making under uncertainty," IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics 30, 60–70, 2000.
[19] Yager, R.R., "On the valuation of alternatives for decision making under uncertainty," International Journal of Intelligent Systems 17, 687–707, 2002.
[20] Luce, R.D. and Raiffa, H., Games and Decisions: Introduction and Critical Survey, John Wiley & Sons: New York, 1967.
[21] Yager, R.R., "Approximate reasoning as a basis for rule based expert systems," IEEE Trans. on Systems, Man and Cybernetics 14, 636–643, 1984.
[22] Shafer, G., A Mathematical Theory of Evidence, Princeton University Press: Princeton, N.J., 1976.
[23] Klement, E.P., Mesiar, R. and Pap, E., Triangular Norms, Kluwer Academic Publishers: Dordrecht, 2000.
[24] Alsina, C., Trillas, E. and Valverde, L., "On some logical connectives for fuzzy set theory," J. Math Anal. & Appl. 93, 15–26, 1983.
[25] Yager, R.R. and Filev, D.P., "Approximate clustering via the mountain method," IEEE Transactions on Systems, Man and Cybernetics 24, 1279–1284, 1994.
[26] Rickard, J.T., Yager, R.R. and Miller, W., "Mountain clustering on non-uniform grids using p-trees," Technical Report# MII-2408 Machine Intelligence Institute, Iona College, New Rochelle, NY, 2004.
[27] Yager, R.R. and Petry, F., "Fusing object information and peer information," International Journal of Intelligent Systems 16, 1419–1444, 2001.
[28] Yager, R.R. and Petry, F.E., "Constructing intelligent fusion procedures using fuzzy systems modeling," Proceedings of the Fourth International Conference on Information Fusion (ISIF), Montreal, Vol. II, ThB3: 23–29, 2001.
[29] Zadeh, L.A., "Toward a perception-based theory of probabilistic reasoning with imprecise probabilities," Journal of Statistical Planning and Inference 105, 233–264, 2002.
[30] Coletti, G. and Scozzafava, R., Probabilistic Logic in a Coherent Setting, Kluwer Academic Publishers: Dordrecht, 2002.
[31] Aczel, J. and Daroczy, Z., On Measures of Information and their Characterizations, Academic: New York, 1975.
[32] Yager, R.R., "Measures of entropy and fuzziness related to aggregation operators," Information Sciences 82, 147–166, 1995.
[33] Prade, H. and Yager, R.R., "Estimations of expectedness and potential surprise in possibility theory," International Journal of Uncertainty, Fuzziness and knowledge-based systems, 417–428, 1994.
[34] Zadeh, L.A., "A computational approach to fuzzy quantifiers in natural languages," Computing and Mathematics with Applications 9, 149–184, 1983.
[35] Yager, R.R., "Quantifiers in the formulation of multiple objective decision functions," Information Sciences 31, 107–139, 1983.
[36] Yager, R.R., "On the inclusion of importances in OWA aggregations," in The Ordered Weighted Averaging Operators: Theory and Applications, edited by Yager, R.R. and Kacprzyk, J., Kluwer Academic Publishers: Norwell, MA, 41–59, 1997.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Knowledge, Uncertainty and Belief in Information Fusion and Situation Analysis

Éloi BOSSÉ, Anne-Laure JOUSSELME and Patrick MAUPIN
*Defence Research and Development Canada Valcartier*

**Abstract.** Situation Analysis, for which information fusion is a key enabler, has to deal with knowledge and uncertainty. This paper discusses the key notions of knowledge, belief and uncertainty in relation to information fusion. The aim is not to provide a theory of some sort but to help the information fusion practitioner to navigate and see the links among the numerous mathematical and logical models/tools that are available to process uncertainty-based information and knowledge.

**Keywords.** Information fusion, situation analysis, knowledge, belief, uncertainty

## 1. Introduction

Defense Research and Development Canada at Valcartier is pursuing the exploration of Situation Analysis (SA) concepts and the prototyping of computer-based situation analysis systems to support and maintain the state of situational awareness for the decision-maker. The integration of the human element at the beginning of the analysis process is an important facet of our approach. This has been discussed in [1] where the objective was to ensure a cognitive fit of the situation analysis support system to the decision-maker.

The situation analysis process, as described in [2], has to deal with both knowledge and uncertainty. To be able to deal with knowledge and uncertainty, a formalization is necessary, defining a framework in which knowledge, information and uncertainty can be represented, combined, managed, reduced, increased, and updated. Some theoretical frameworks available to model the SA process (or some parts of it) and taking account uncertainty are currently under study. These potential frameworks can be divided into categories such as qualitative, quantitative and hybrid approaches. Qualitative approaches seem to better suit reasoning on knowledge, while quantitative approaches are better candidates for uncertainty representation and management. A hybrid approach is a mix of quantified evaluations of uncertainty and high reasoning capabilities.

Characterizing and interpreting uncertainty are probably the most important and difficult tasks for the situation analysis process. From this characterization of uncertainty is derived the choice of the most adequate theory to be used for this process. The paper focuses on the data/information fusion (DF) process, a key enabler in situation analysis. Data, information, knowledge, belief and uncertainty are key notions that the data fusion designer has to play with in building a system. An enormous amount of

work has been done in mathematics and logic to deal with those notions but not necessarily in the perspective of data fusion. This paper discusses the key notions of knowledge, beliefs and uncertainty in relation to information fusion. The aim is not to provide a theory of some sort but to help the information fusion practitioner to navigate and see the links among the numerous mathematical and logical models/tools that are available to process uncertainty-based information and knowledge.

The paper is organized as follows. Section 2 describes the cognitive hierarchy. In section 3, we discuss the notions of knowledge, belief and uncertainty. Section 4 presents the mathematical and logical tools to process uncertainty-based information and knowledge. Finally, section 5 is the conclusion.

## 2. The Cognitive Hierarchy

In this section, we introduce the notion of three distinct levels of abstraction for the resource we call information: data, information and knowledge. Figure 1 illustrates the three-level cognitive hierarchy that moves from data to knowledge. In the military context, **Data** are the raw material of Command and Control and originate from feedback of actions in the situation (battlespace). They include signals from any kind of sensor, whether organic or non-organic, or communicated between any kinds of nodes in a system. **Data** are provided meaning through the act of processing. Processing involves aligning, organizing, formatting, collating, filtering, plotting and display, and any other similar conditioning function. **Information** is the name we assign to data placed in context, indexed and organized. **Knowledge** is information that has been evaluated and analyzed as per reliability, relevance and importance. Knowledge is information understood and explained and it is where we begin to develop situation awareness, by integrating together various sets of information and interpreting what they could possibly mean.

Understanding means that we have gained situational awareness, and we can apply the knowledge to effectively implement a plan or action to achieve a desired goal. Although we make clear distinctions between the classes of information, the transitions between them as we progress through the decision cycle and, concomitantly, up the cognitive hierarchy, are markedly less distinct in reality. Indeed, data gradually merges into information, information into knowledge and knowledge into understanding.

Commanders need knowledge and understanding to make effective decisions. The goal, therefore, is not to process vast amounts of data. The goal is to gain a level of understanding as close to the actual situation as is possible given the inherent uncertainty of war and presenting it to the commander in a form that approximates as closely as possible his way of understanding the situation (battlespace).

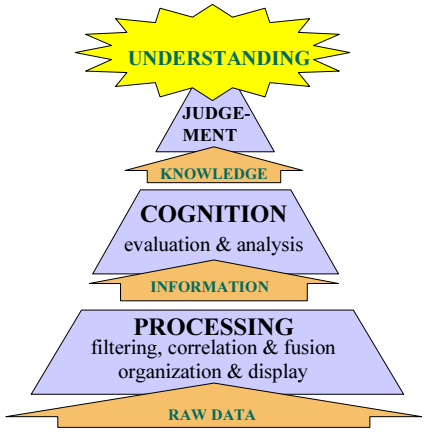



**Figure 1.** The cognitive hierarchy.

## 3. Knowledge, Belief and Uncertainty

Knowledge, belief and uncertainty are three key notions of the situation analysis process (through data/information fusion). The two basic elements involved in situation awareness are the *situation* and the *person*. In Fig. 2, the *situation* can be defined in terms of events, entities, systems, other persons, etc., and their mutual interactions. The *person* can be defined according to the cognitive processes involved in situation awareness (SAW), or simply by a mental or internal state representing the *situation*.

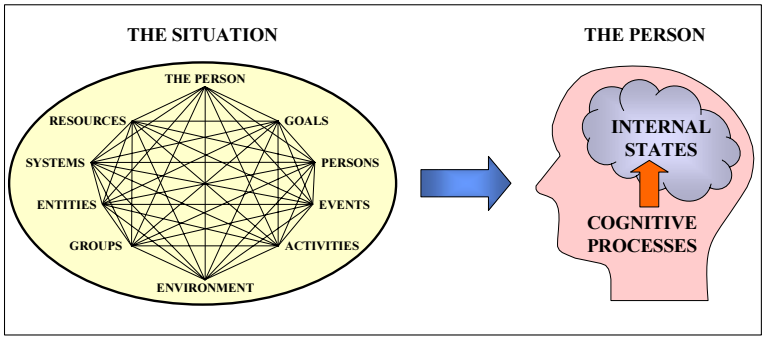



**Figure 2.** A simple illustration of the elements involved in SAW.

The simple representation illustrated in Fig. 2 is not a model in itself. It is a simple schema of the general elements, from both the perspective of the situation and the person, which should appear in a global definition of situation awareness [3].

The cognitive process of the human must be supported to help him build his mental model (understanding) of the situation. To this end, the human is using observing devices/agents (sensors or other humans) and computers (processing) to support his reasoning about the situation. Figure 3 illustrates the main challenge in situation analysis. Belief and knowledge representation is a crucial step needed to transform data into knowledge in the cognitive hierarchy. The data/information coming from the different


**Figure 3.** Measuring and reasoning about the situation.

sources must be converted into a certain language or with other means (e.g. visualization) so as they can be processed and used by the human to build his mental model in order to decide and act. One great challenge in designing a support system is to make use of the mathematical and logical tools that can allow measuring and reasoning about the situation using a common analysis framework.

### *3.1. Knowledge and Belief*

#### *3.1.1. Belief*

Following Barwise and Perry [4], we assume that there are real situations in the world. Through interaction with the world, people seek situation awareness by forming perceptions, comprehensions and projections about these situations [5]. Hereafter these perceptions, comprehensions and projections are characterized as beliefs. Whether explicitly or tacitly conceived, it is customary to describe beliefs through sentences of the form `X believes that` σ (e.g. `Fred believes that it is raining`), where σ expresses some claim about the world and `X` identifies an individual having a belief attitude toward that claim. Consequently, the remainder of this paper interprets beliefs propositionally as mental states that are ascribed to an individual (human or machine) and that make some truth functional claim about the world. Propositional expressions (e.g. `SAM#1 is targeting F18#7`) will be used to express the propositions associated with these mental states, while propositional attitude expressions (e.g. `AEW#2 believes that SAM#1 is targeting F18#7`) will be used to express the association of a propositional belief state with an individual.

### *3.1.2. Knowledge*

Situation awareness is usually associated with having perception, comprehension and projection knowledge about the world. Philosophers have struggled to produce a cogent definition of knowledge for centuries. While Plato's *justified true belief* fails to adequately define knowledge (Fig. 4), the elements of:

- truth;
- justification; and
- belief

in some way figure prominently in many accounts of knowledge [6]. For that reason, those elements serve as the cornerstones of the approach presented within this paper. We accept belief of $\sigma$ as a necessary, but not sufficient, condition for knowledge of $\sigma$. In this paper, we assume that the knowledge is propositional; the analysis of non-propositional knowledge is beyond the scope of this paper.

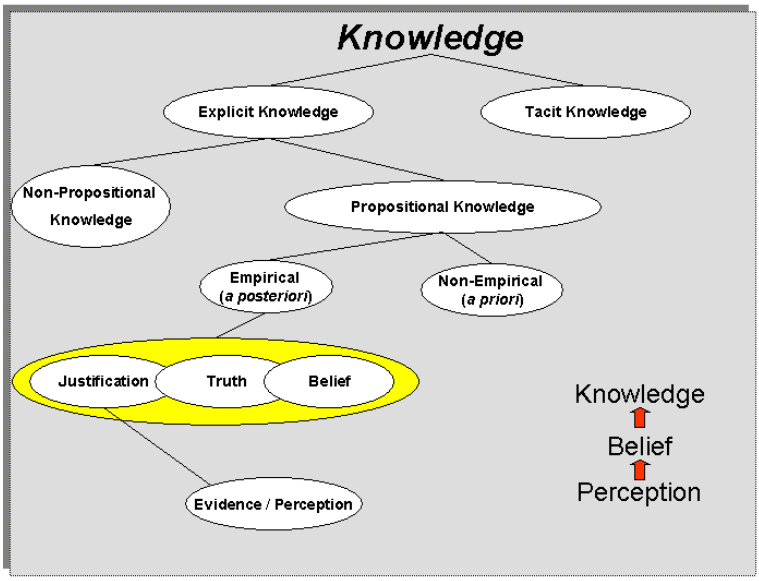



**Figure 4.** The notion of knowledge.

We are then able to determine the truth of propositional expressions, without appealing to the interpretations of their terms, by reasoning with the inference relation to determine what must be true. As equivalently noted in Lambert [7], by combining some domain theory D of beliefs about the world with a theory M expressing the meaning of the terms used in the domain theory, the consequences of D can then be deduced as $\{\tau \mid (D \cup M) \vdash \tau\}$. A consistent and complete theory M will ensure that inconsistencies and ignorance in our data fusion system derive from the beliefs D about the world. In this way the inference relation H provides the remaining element of justification. If the beliefs in D are all true, then the beliefs in $\{\tau \mid (D \cup M) \vdash \tau\}$ must also be true, and moreover, they are justified because the inference relation H can be used to explain why they must be true! The three elements of belief, truth and justification can be related in this way to provide a concept of knowledge for our automated fusion systems.

Knowledge involves belief, truth and justification. Moreover, it presents a formal framework for fusion systems which can be applied to express these elements.

- the *truth* of a proposition can be asserted by asserting a propositional expression σ expressing the proposition. The stored *belief*, by individual X, of a proposition expressed by σ, can be asserted by having $\Sigma_X$ denote the set of stored beliefs of X and asserting $\sigma \in \Sigma_X$
- the *justified* belief, by individual X of a proposition expressed by σ, can be asserted by having $\Sigma_X$ denote the set of stored beliefs of X, having $\vdash$ denote the inference relation used by X, and asserting $\Sigma_X \vdash \sigma$. So
- X believes that σ *if* $\Sigma_X \vdash \sigma$, while
- σ *if* σ is true.

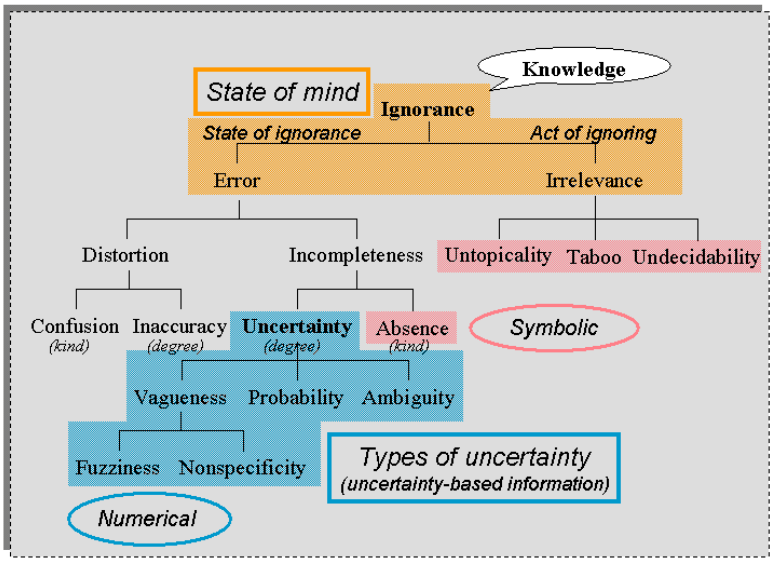



**Figure 5.** Smithson's taxonomy of ignorance (Smithson (1989)).

### *3.1.3. Smithson's Taxonomy of Ignorance*

Smithson [8] proposes a taxonomy of ignorance where uncertainty appears as a kind of ignorance "… one of the most manageable kinds of ignorance." This taxonomy is reproduced in Figure 5. Smithson interprets ignorance as nonknowledge. He initially separates ignorance into two categories: the state of ignorance (*error*) and the act of ignoring (*irrelevance*). The latter corresponds to a deliberate action to ignore something irrelevant to the problem-solving situation, whereas the former is a state (of ignorance) resulting from different causes (*distorted* or *incomplete* knowledge). For Smithson, uncertainty is incompleteness in degree (as compared to *absence* which is incompleteness in kind), and is sub-divided into three types: *probability*, *vagueness* (being either *non-specificity* or *fuzziness*) and *ambiguity*. The authors have decided to adopt Smithson's taxonomy as a starting point basis for developing fusion systems that reason with uncertainty.

### 3.2. Uncertainty

#### 3.2.1. What Is Uncertainty?

*Uncertainty* is a widely used term within the artificial intelligence and engineering communities. However, the authors in these fields of application and research do not always agree on the meaning of uncertainty, on its different types, on possible sources, synonyms, possible classifications, representations, *et cetera*. In [9], the concept of uncertainty and its related concepts such as imperfection, imprecision, vagueness, ambiguity, incompleteness, ignorance are discussed in more detail. For the sake of this paper, let us distinguish four different facets of uncertainty (Fig. 6).

#### 3.2.2. Meanings of Uncertainty

The review [9] of some authors' interpretations and visions of uncertainty, in conjunction with some general definitions of uncertainty, shows that there are two main definitions for uncertainty (Fig. 7):

I. uncertainty is a state of mind;
II. uncertainty is a property of the information.

Sense I is the feature of what cannot be predicted, what cannot be stated with exactness. However, faced with the need of formalization in order to reduce uncertainty in finality, probabilities are used and thus applied to represent a more local (physical) state of uncertainty (sense II).

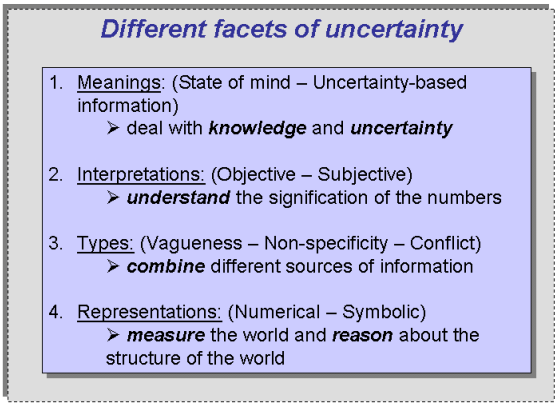



**Figure 6.** The main facets of uncertainty.

Bouchon-Meunier and Nguyen [10] also adopt this point of view when they present uncertainty as a result of imprecision, vagueness, incompleteness, etc. These causes can induce a level of uncertainty in someone's mind. In a similar manner, Smets [11] refers to sense I when mentioning subjective uncertainty. In this case, uncertainty being a mental state is not so far from ignorance.

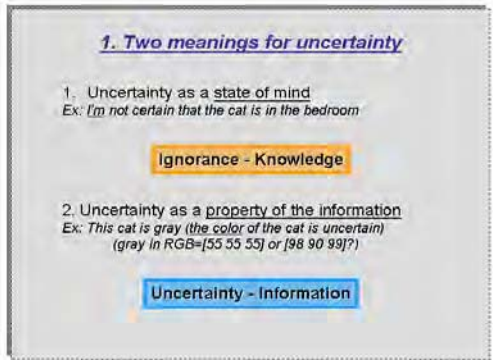



**Figure 7.** Meanings of uncertainty.

Sense II refers to a more practical aspect of uncertainty that is better captured by the term imperfection. Imperfection (of information) induces uncertainty (in our mind). Hence, when the authors discuss different kinds of uncertainty, they mean kinds of imperfection (of the data, information, or knowledge) that induce different kinds of uncertainty. When Smets distinguishes three main types of imperfection, he refers to the same things that Klir [12], Bouchon-Meunier and Nguyen or Smithson call types of uncertainty. The confusion seems to appear with subjective uncertainty (of Smets). Indeed, becoming a source of information, a person in uncertainty will provide an uncertainty information (*i.e.* inducing uncertainty in another person's mind) but this subjectivity, we think, is merely a characterization of the source of information rather than the information itself. The information can be both subjectively uncertain and vague.

### *3.2.3. Interpretations of Uncertainty*

By interpretations (or theories) of uncertainty, we mean ways of obtaining uncertainty evaluations. The two main interpretations for probabilities emerged during the last two centuries (**empirical** (objective)) and **inductive** (subjective)) can be extended to uncertainty in general. Indeed, one's uncertainty can either come from an experiment (in order to apprehend the situation), form historical records, or from one's own (or another person's) internal representation of the situation. This thus refines the distinction between *objectivity* (empirical uncertainty) and *subjectivity* (inductive uncertainty) as proposed by Smets in his thesaurus. This classification of uncertainty is detailed in figure 8.

On one hand, the *empirical* uncertainty is divided into three categories: *Classical* interpretation is based on the analysis of all possible outcomes; hence it implies a perfect description of the world. *Relative frequencist* interpretation is issued from a large amount of experiments. The evaluations are made on the "long run" and imply the use of some electronic devices. *Propensity* is a natural inclination or preference toward a particular state. This interpretation comes from physical properties and implies a physical model of the situation. Ultimately, long run frequencies may converge toward a propensity. On the other hand, the *inductive uncertainty* is separated into two categories: *Logical* interpretation (syntactic or semantic) implies the use of some reasoning, of some rules of interpretation of the world (ex.: Modus Ponens) to quantify the uncertainty. And finally, *belief-based* interpretation is relative to our *belief*, our state of mind, our intuition, our conviction, our confidence, etc. Sources able to provide this

kind of evaluation of uncertainty are essentially human sources. Every quantitative representation (formalization) of uncertainty can either be interpreted as objective or subjective.

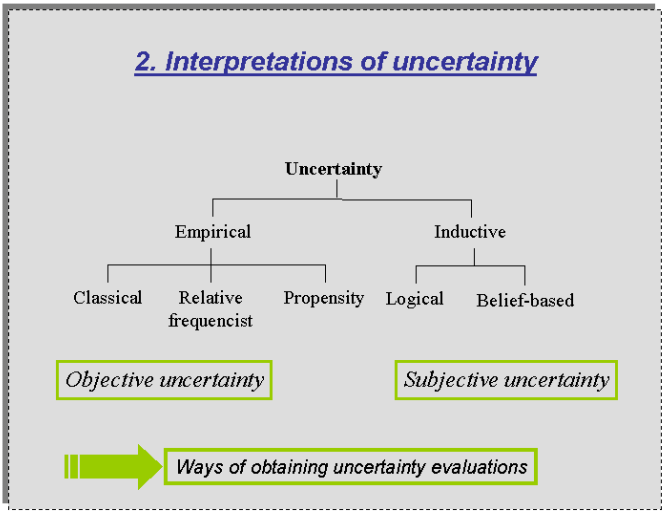



**Figure 8.** Interpretations of uncertainty.

A probability as well as a fuzzy membership can come from experiments or from purely subjective opinions (or a mix of both in the case of interviews of persons, for example when asking a large group of persons for classing human heights into three categories *tall*, *medium*, *small*. The result can be interpreted within a frequentist view, but also subjectively interpreted because each classification comes from a person. To end, subjective or objective uncertainty are not two different types of uncertainty but two different (epistemological) interpretations.

### *3.2.4. Types of Uncertainty*

Types of uncertainty are defined according to the kinds of imperfection causing them, and thus this imposes a reference to sense I of uncertainty (Fig. 6). Although the taxonomy differs from one author to the other, some key terms commonly appear: vagueness, fuzziness, non-specificity, ambiguity, conflict, probability (or randomness), incompleteness. We can then qualify uncertainty by each of these terms (uncertainty due to vagueness, uncertainty due to randomness, …) providing the different types of uncertainty.

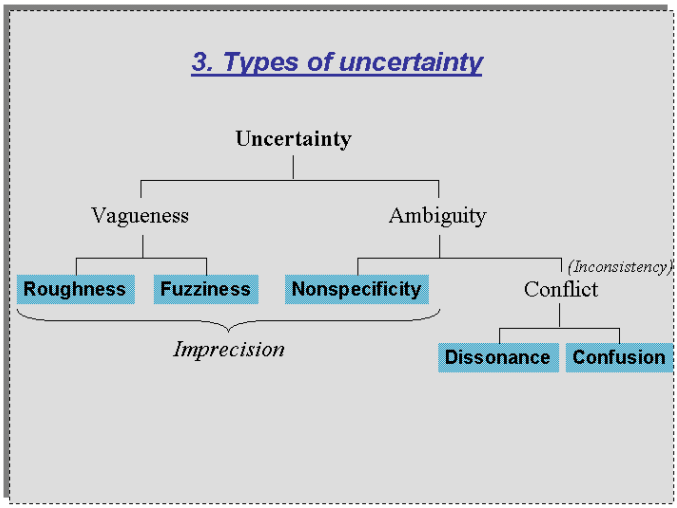



**Figure 9.** Types of uncertainty.

### *3.2.5. Mathematical Representations of Uncertainty*

To model uncertainty, many mathematical tools (see Fig. 10) have been developed, being either qualitative (modal logic, nonmonotonic reasoning, etc.) or quantitative approaches (probability theory, fuzzy sets, rough sets, random sets, belief functions, etc.). Each of these approaches is often compared based on different strengths and weaknesses, their better correspondence to a particular type of uncertainty, their requirement of prior knowledge, their computational time, or on the need of independence of data. It appears than none of the available mathematical tools is the best, and it becomes more and more obvious that depending on the kind of problem we are facing, one theory can force itself as the better choice. To support this argument, Smets in [13] makes a formal discussion on ignorance and the need for well founded theories. Another eclectic point of view is the one advanced by Klir in [14]. Besides the quantitative representation of uncertainty, uncertainty can be quantified in an equivalent manner as information can be (as proposed by Shannon in his well known theory of information). Such measures of uncertainty allow for example, “extrapolation of evidence, assessing the strength of relationships between groups of variables, assessing the influence of given input variables on given output variables, measuring the loss of information when a system is simplified and the like.” [14]

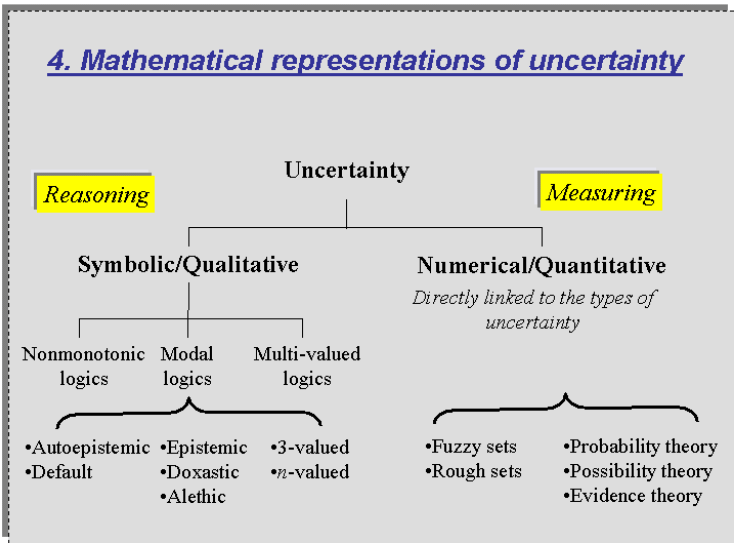



**Figure 10.** Mathematical representations of uncertainty.

## 4. Mathematical and Logical Tools

Situation analysis has to deal with both knowledge and uncertainty and thus with ignorance and information. A distinction between ignorance and uncertainty (Figures 5 & 7) appears as a crucial point in the process. Indeed, we must not confound the state of mind of the operator being uncertain when facing a decision, an act to commit (Sense I of the definition of uncertainty), and the uncertainty arising from the physical limitations of means of perception (human sensory system, sensors, measure devices), this uncertainty being of different types and easily formalizable either quantitatively or symbolically (Sense II of the definition of uncertainty).

To be able to deal with knowledge or uncertainty, a formalization is necessary for defining a framework in which knowledge, information and uncertainty can be represented, combined, managed, reduced, increased, and updated. The objective is (1) to build a model of situation directly usable by the different theories of reasoning under uncertainty, (2) to be able to deal with both knowledge and uncertainty. Some theoretical frameworks available to model the SA process (or some parts of it) and taking account uncertainty have been identified, their abilities and inabilities are currently under study.

These potential frameworks (Figure 11) can be divided into main categories: qualitative approaches (such as modal logic, non-monotonic logic, truth maintenance systems…) and quantitative approaches (such as probability theory, evidence theory, fuzzy sets, random sets, possibility theory…). Qualitative approaches seem to better suit reasoning on knowledge, while quantitative approaches are better candidates for uncertainty representation and management. Hence, a good solution for a global modelization of the situation could be a hybrid approach (quantitative logics, incidence calculus…) mixing quantified evaluations of uncertainty and high reasoning capabilities.

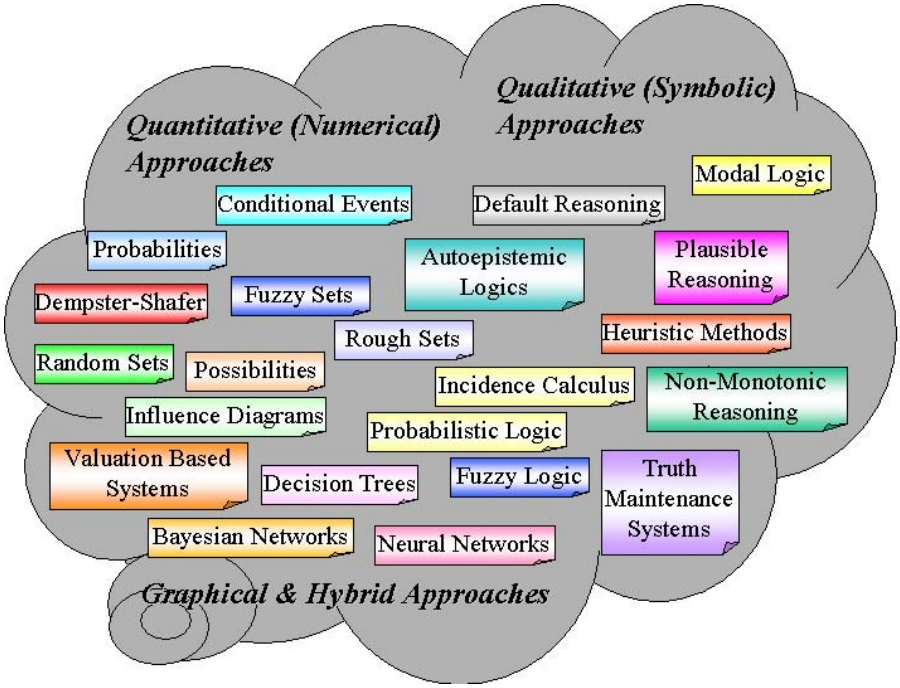



**Figure 11.** The mathematical and logical tools.

Characterizing uncertainty is probably the most important and difficult task, devolving on each subprocess of situation analysis. From this characterization of uncertainty is derived the choice of the most adequate theory to be used for this subprocess. It seems intuitively that each type of uncertainty can be present during the whole process of situation analysis. Hence, the choice of one framework for a given subprocess is premature and implies a deeper study of each subprocess.

The choice of the adequate theory can also be guided by some possibilities (or impossibilities) of information (or data) acquisition, which directly refers to what we called epistemic interpretations of uncertainty (Figure 8).

### *4.1. Qualitative/Symbolic Approaches*

In this part, some logical frameworks for uncertainty and knowledge processing are introduced. These frameworks, also called qualitative, logical, truth-functional, but also intensional [15] are all extensions of **classical logic**.

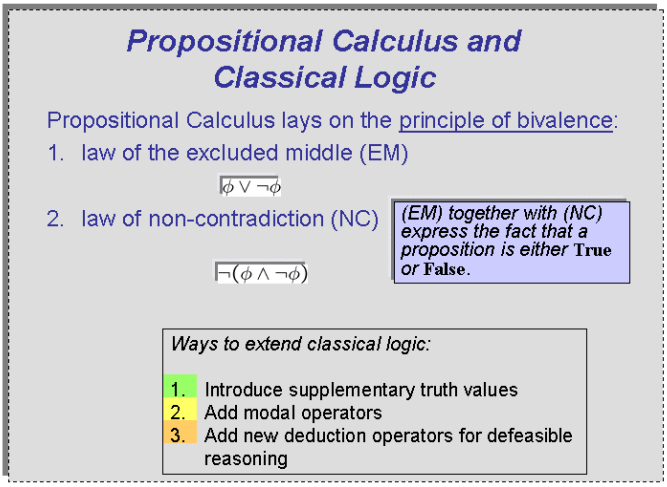



**Figure 12.** Classical logic.

As Pearl puts it, symbolic approaches are undermined by the fact that they are "semantically clear but computationally clumsy." Symbolic frameworks are needed to **reason** about the structure of the world, **understand** and explain facts about it, but when it comes to handling measures symbolic frameworks are of limited use.

Although it is generally accepted that **classical logic** (Figure 12) is not able to deal with uncertainty (even if it could be a matter of controversy), it remains the basis of all logical approaches). Initially, **logic** has been conceived to reason about certain, indubitable facts. However, facing the growing and unavoidable constraint of dealing with uncertainty, this classical framework has evolved and led to logics capable of approximate reasoning. The limitations of classical logic introduce the alternative logical approaches (Figure 13) namely *multi-valued logics*, *modal logics* and *non-monotonic logics*.

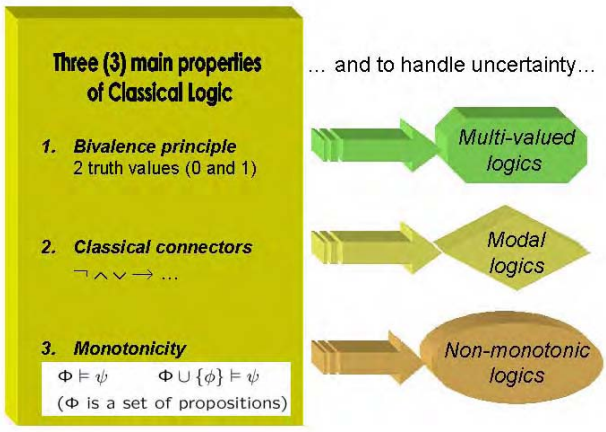



**Figure 13.** Classical logic extensions to handle uncertainty.

Classical logic (Figure 12) lies on the principle of bivalence expressing the fact that a proposition is either TRUE or FALSE. Hence, only two truth values are allowed in classical logic. Classical logic also lies on the principle of Excluded Middle (EM) expressing that a proposition is either asserted or negated making it difficult to express natural language expressions with such a bivalent logic as it fails to deal with vague terms; at which point, **multi-valued logics** are introduced.

The most common way to go beyond bivalence is to introduce supplementary truth values in the classical logic framework. With the introduction of a third truth value, a proposition can be neither TRUE nor FALSE. This is an important step forward for uncertainty processing based on the classical logic framework. Pursuing this extension to an infinity of truth values leads to fuzzy logic, an approach considered as hybrid since it combines numerical evaluations with a logical framework.

**Modal logic** is another important approach for dealing with uncertainty and knowledge that describes **in what manner** a proposition is TRUE ("possibly TRUE," "necessary TRUE," but also "knowing it is TRUE," …). In modal logic, these **modalities** are expressed by non truth-functional operators called **modal operators**. Using the appropriate combination of axioms and modalities, it is possible to build modal logic frameworks for reasoning on the important notions of **knowledge** and **belief**.

Another basic limitation of classical logic, but also of derived logics, is their inadequacy for dealing with **defeasible reasoning**, the reasoning being monotonic in these frameworks. **Monotonicity** means, practically, that once a conclusion has been accepted as TRUE it cannot be defeated when confronted with new evidence. Dropping the monotonicity property leads to **non-monotonic logics**. Although non-monotonic logics are more appropriate than classical schemes for commonsense reasoning, they are much more demanding as far as reasoning schemes are concerned. Among non-monotonic logics, *default*, *autoepistemic* and *circumscription logics* are the most popular. Note that non-monotonic reasoning is not limited to symbolic approaches. Indeed, most numerical approaches own this property by the means of some inference or combination rules such as for example Bayes' or Dempster's.

The three alternatives (Figure 13) to classical logic present different features that do not make one approach better than the other. Rather, one should select the appropriate framework for uncertain reasoning on the basis of objective criteria based on the application.

### *4.2. Quantitative/Numerical Approaches*

The aim of this part is to introduce the main mathematical frameworks for uncertainty processing. Deliberately choosing not to treat normative theories like decision and games theories, or application-related theories like pattern recognition or machine learning, although they deal with uncertainty, we focus strictly on the modeling of uncertainty and knowledge, leaving aside application considerations.

Hence, this part addresses numerical approaches for dealing with uncertainty. These methods, also called quantitative, parametric, measure-based, but also extensional (as mentioned by Pearl), are all based on **set theory**, compared to symbolic approaches that are based on classical logic. As Pearl puts it, numerical approaches are characterized by the fact that they are "computationally attractive, but semantically poor."

However, they are needed to **measure** and **perceive** objects of the world although they are less flexible and expressive when used for reasoning tasks. Nevertheless, the

actual tendency is to borrow mechanisms and representations schemes of symbolic frameworks (*i.e.* possible worlds semantics) to fill the gap, leading to hybrid approaches.

Numerical frameworks, besides allowing representations of qualitative notions such as degrees of belief, possibility, membership, etc., also allow the representation and measure of different types of uncertainty such as vagueness, ambiguity and conflict through the so-called concept of **uncertainty-based information**. Uncertainty-based information is processed following Generalized Information Theory uncertainty principles according to which different types of uncertainty-based information criteria can be maximized/minimized.

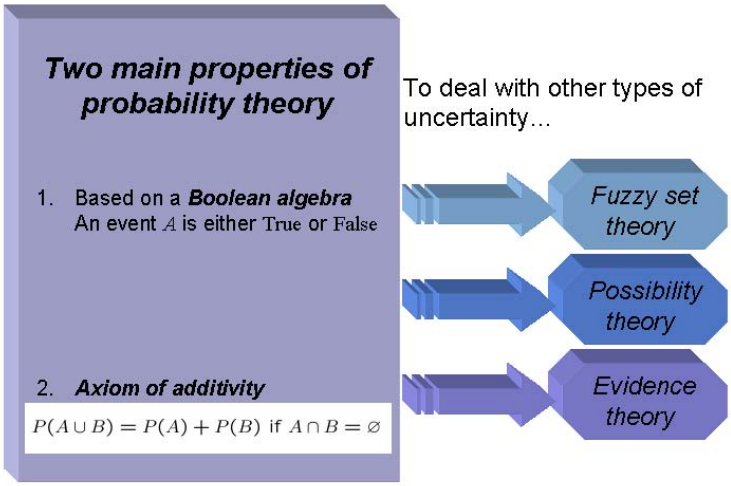



**Figure 14.** Extensions of the theory of probability.

**Probability theory** is the basis of numerical approaches, under Kolmogorov's axiomatization of the theory that lies on axioms imposing the use of (1) a σ-field algebraic structure, (2) sharp values (*i.e.* single values) for (3) numerical evaluations taken in the (4) set of real numbers $\mathbb{R}$, (5) a normalization and (6) countable and finite additivity conditions. However, we should keep in mind that non-Kolmogorovian approaches exist, based on the modification of the basic axioms. For example, one can obtain theories where negative, complex or qualitative values are allowed, but also for non-normalized probabilities. Note that dropping the additivity axiom leads to Dubois and Prade's theory of fuzzy probability, Dempster-Shafer theory or possibility theories, among others.

**Dempster-Shafer theory** is one of the most promising avenues for dealing with uncertainty. Dempster-Shafer theory, also called evidence theory or belief function theory, is a generalization of probability theory in the sense that probability functions are special cases of belief functions. Indeed, allowing belief functions to be sub-additive (contrary to probability functions that are required to be additive) expands the support of the uncertainty functions to the power set of the universe of discourse. In other words, dropping the additivity axiom of probability theory allows statements such as "I believe it will rain tomorrow with a confidence of 0.8 but I also admit that I don't

know with a weight of 0.2," to be represented and processed. This high capacity for representing ignorance is one of the most attractive features of Dempster-Shafer theory.

**Fuzzy sets theory** aims at representing another kind of uncertainty called fuzziness (or more generally vagueness), quantified using characteristic functions (or membership functions) to assign truth *degrees* to events. Thus in fuzzy sets theory the uncertainty is quantified by playing on the *arity* of the characteristic function (usually an infinity of values in the [0, 1] real interval) instead of juxtaposing a confidence function to a binary characteristic function as done in probability theory. Although probability and fuzzy sets theories are still nowadays considered as equivalent by a small community, they clearly address different kinds of uncertainty, and are rather complementary than rivals. Fuzziness is a fundamental concept of mathematics generalizing the notion of membership to a set, allowing thus the extension of other theories of uncertainty. Indeed, a degree of confidence (probability for instance) can be attributed to the realization of an event, this event being itself more or less TRUE with respect to a given reference. In other words, an event can be both fuzzy *and* random. Examples of this type are fuzzy probability, hybrid numbers and fuzzy belief functions theories.

Built on fuzzy sets theory, but giving another meaning of the characteristic function, the **theory of possibility** is a theory of partial ignorance. In fact, possibility theory, just as probability theory, can be interpreted in the more general framework of evidence theory, possibility functions being just another special case of belief functions. The theory of possibility copes with the evaluation of confidence values (*i.e.* possibilities) of *nested* elements of the power set. This restriction on the algebraic structure allows the use of fuzzy sets theory for manipulating possibility functions, without loosing the expressiveness of evidence theory when processing ignorance statements. In spite of this restriction, sets considered in this theory are crisp (a feature shared with probability theory and evidence theory), the principal concept borrowed from fuzzy sets theory being the idea of degree, here applied to the notions of possibility and necessity.

An alternative to fuzzy sets theory to represent vagueness is **rough sets theory.** This theory has been presented as a theory for dealing with vagueness and uncertainty, but more precisely it deals with *indiscernability* of objects in the universe of discourse. Our limited knowledge of this universe is represented by a partition regrouping indiscernible elements. Based on this knowledge, an information (represented by a subset of the universe) is then approximated by lower and upper bounds (*i.e.* two other subsets). Hence, our incapacity to class elements into one or another class is modeled by lower and upper bounds instead of membership degrees as done in fuzzy sets theory. Rough sets theory can be seen as a qualitative method for the processing of uncertainty, since its basic constituents are upper and lower limits, partitions and sets. The numerical evaluations are not made within the theory but constrained by these qualitative notions. Indeed, on this structure, membership functions have been defined, but also belief and plausibility functions, the latter allowing rough sets to be also interpreted in the evidence theory framework.

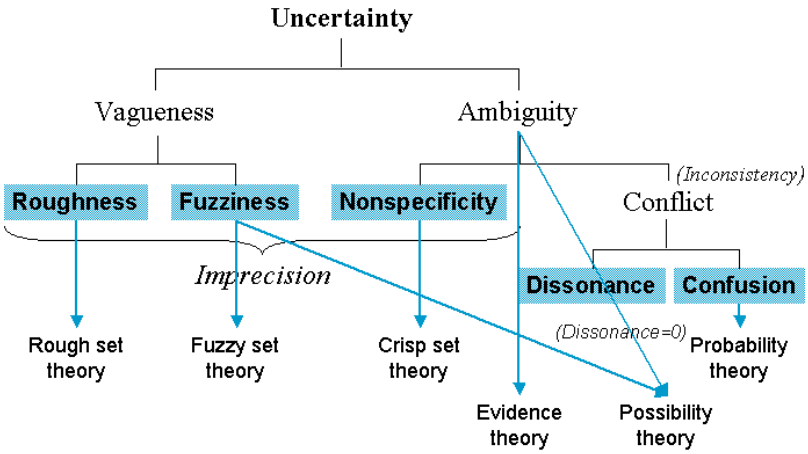



**Figure 15.** Types of uncertainty versus the numerical approaches.

Probability theory deals with randomness (confusion), and the concept of random variable is a classical means to represent events. A generalization of the random variable concept is the concept of random set that yields **random sets theory**. Whereas a random variable but also a random vector are both functions mapping elements of the universe of discourse to real values, a random set generalizes these concept mapping elements of the universe to any subset, whether or not they are a part of the real number sets. This expansion comes to consider random entities of the power set (thus assigning them "probability" evaluations). The random sets framework appears then as a sufficiently generalized one in which all six theories presented above can be expressed. Indeed, probability functions, belief functions, possibility functions, but also fuzzy sets, rough sets and finally conditional events algebra, are special cases of random sets.

Figure 15 summarizes the various numerical approaches discussed above with their associated theories. The comparison between these approaches is still a matter of discussion. None of these theories is the best; they are rather complementary, each being specially designed for different kinds of problems.

### *4.3. Graphical/Hybrid Approaches*

On the other hand, although numerical approaches are computationally efficient, they lack a clear semantics. In a tentative to resolve this conflict, the actual tendency is to **reconciliate** both approaches. Integrating mechanisms and representations schemes from symbolic and numerical frameworks leads to hybrid approaches.

Most numerical theories have their logical counterpart. The model of construction is the same for all the theories: the event (subset of $\Theta$) $A$ is seen as a logical formula $\phi$, referring to the concepts developed in the symbolic approaches. Figure 16 summarizes the various links that can be made between the numerical and symbolic approaches.

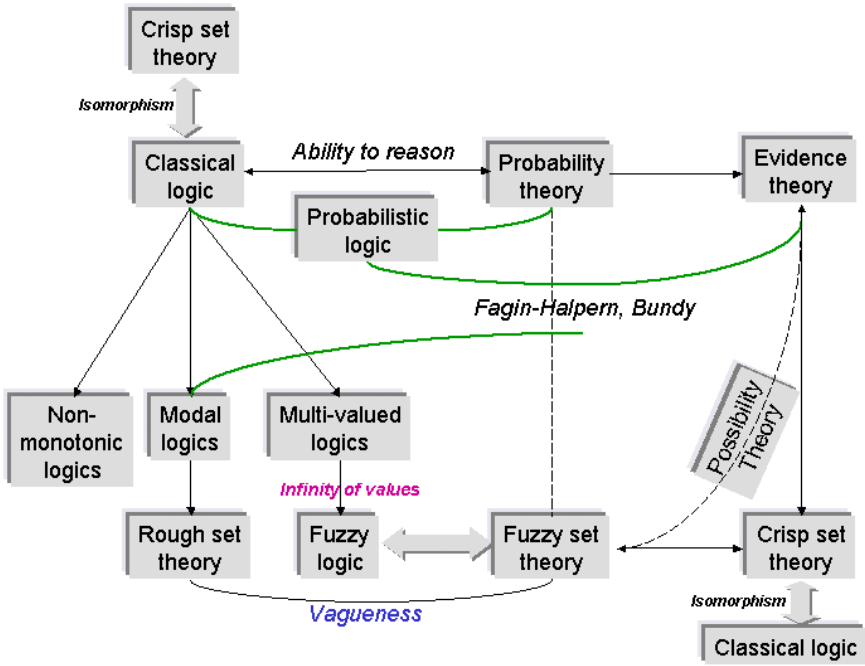



**Figure 16.** Summary of the various links between the approaches.

Probabilistic logic is a logic for reasoning about probabilities, *i.e.* the probability calculus on rules, on logical formulas. It was first introduced by Nilsson in 1986, as a "semantic generalization of logic in which the truth values of sentences are probability values" [16]. The principle of probabilistic logic lies on the fact that the truth values of propositions are their probability of occurrence. The purpose is then to deal with propositional probabilities *i.e.* probabilities assigned to particular propositions or assertions. Probabilistic logic then combines logic with probability theory, and reduces to ordinary logic when the probabilities of all sentences are either 0 or 1. This approach is based in the possible worlds semantic. Other approaches were later developed by Halpern and Fagin [17,18]. The theoretical basics of probabilistic logic are those of probability theory on the one hand, and of classical logic on the other.

One of the first papers proposing fuzzy logic for natural language modeling is due to Lakoff [19]. For a point of view arguing that fuzzy logic is nothing else than classical logic in disguise see Elkan's paper [20] and the corresponding answer of Klir in [21]. Degrees of memberships of fuzzy sets theory are equaled to truth values in fuzzy logic.

The incidence calculus was proposed in 1985 by Alan Bundy [22,23] as a probabilistic logic for reasoning under uncertainty. This is a method for managing uncertainty in a numerical way. Different from other numerical approaches, in incidence calculus probabilities are associated with a set of possible worlds rather with formulas directly. The probability of a formula is then calculated through the incidence set assigned to the formula. Incidence calculus itself appears to be a unification of both symbolic and numerical approaches. It can therefore be regarded as a bridge between the two reasoning patterns [24].

The graphical approaches will not be addressed in this paper due to paper space constraints.

## 5. Conclusions

Knowledge and belief analysis as well as uncertainty management are central to Situation Analysis. The methods for modeling and processing uncertainty naturally differ from one scientific community to another. For example, concerning perception, probability theory is adopted by most electrical engineers. In the case of reasoning, logical approaches are rather used by the artificial intelligence community (logical AI), philosophers and logicians.

In a situation analysis context, the need for manipulation of numerous theoretical frameworks of different natures rapidly becomes a problem. Traditionally, this problem is tackled using two means:

1. a first approach consists in making the different theoretical frameworks "communicate" with each other. For example, within the same system of SA, one module can process information using probability theory whereas another parallel (or serial) module can use fuzzy set theory. The difficulty becomes then to define algebraic transformations such that the modules can coherently communicate;
2. a second approach aims at using "general" frameworks, and two tendencies appear: a) Use a single formalism, such as random sets theory for quantitative information or autoepistemic logic for logical information. b) Directly process two conceptually distinct pieces of information through a single formalism. This latter approach uses modal logics and possible worlds semantics to reason on both knowledge and uncertainty. A related approach defines a semantics based on the random worlds in order to integrate statistical data processing and reasoning on knowledge.

Therefore, as outlined by several authors including Pearl [15] "Extensional (syntactic) systems are computationally convenient but semantically sloppy, while intensional (semantic) systems are semantically clear but computationally clumsy." It seems, that much of the difficulty encountered in the formalization of approaches dealing with computation and reasoning under uncertainty lie in the reconciliation of the syntactic and semantic aspects of the problem.

## References

[1] Bossé, É., Paradis, S., Breton, R., '*Decision Support in Command and Control: A Balanced Human-Technological Perspective*', Chapter of this NATO ASI Book, 2004.
[2] Roy, J., *From Data Fusion to Situation Analysis*, Proc. of the ISIF Fusion2001 Conference, Montréal, Canada, July 2001.
[3] Breton, R. & Rousseau, R., "*Situation Awareness: A Review of the Concept and its Measurement*," CRDV-TR-2001-220, 2002.
[4] Barwise, J. and J. Perry. *Situations and Attitudes*. The MIT Press – A Bradford Book, Cambridge, Massachusetts, 1983.
[5] Lambert, D.A., *Assessing Situations*, Proceedings of IEEE Information, Decision and Control, pp. 503–508. 1999.
[6] Lehrer, K. Knowledge. Clarendon Press, Oxford, 1974.

[7] Lambert, D.A. "*Grand Challenges of Information Fusion*," Proc. of the ISIF Fusion2003 Conference, Cairns, Australia, July 2003.
[8] Smithson. M. *Ignorance and Uncertainty*. Springer-Verlag, Berlin, 1989.
[9] Jousselme, A.-L., Maupin, P., Bossé, É., '*Uncertainty in a Situation Analysis Perspective*', Proc. of the ISIF Fusion2003 Conference, Cairns, Australia, July 2003.
[10] Bouchon-Meunier, B. and Nguyen, H.T. *Les incertitudes dans les systèmes intelligents*, volume 3110 of *Que sais-je*? Presse Universitaires de France, Paris, 1996.
[11] Smets, P. *Imperfect Information: Imprecision – Uncertainty*. In A. Motro and P. Smets, editors, Uncertainty Management in Information Systems. From Needs to Solutions, pages 225–254. Kluwer Academic Publishers, 1997.
[12] Klir, G.J. On the alleged superiority of probabilistic representation of uncertainty. IEEE Transactions on Fuzzy Systems, 2(1):27–33, February 1994.
[13] Smets, P. Varieties of Ignorance and the Need of Well founded Theories. Information Sciences, 57–58:135–144, 1991.
[14] Klir, G.J. and Yuan, B. Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall International, Upper Saddle River, NJ, 1995.
[15] Pearl J., Probabilistic reasoning in intelligent systems: Networks of plausible inference. San Mateo, CA: Morgan Kaufmann Publishers, 1988.
[16] Nilsson., *Nils Probabilistic logic*. Artificial Intelligence, 28(1):71–87, 1986.
[17] Halpern J.Y. *An analysis of first-order logics of probability*. Artificial Intelligence, 46:311–350, 1990.
[18] Fagin, R., Halpern, J.Y., and Megiddo, N. *Logic for reasoning about probability*. Information and Computation, 87(1,2):78–128, 1990.
[19] Lakoff, G. *Hedges: A study in meaning criteria and the logic of fuzzy concepts*. Journal of Philosophical Logic, 2:458–508, 1973.
[20] Elkan, C. *The paradoxical success of fuzzy logic*. IEEE Expert – Intelligent Systems and Their Applications, 9(4):3–8, 1994.
[21] Klir, G. and Yuan, B. *Elkan goes wrong – again*! IEEE Expert – Intelligent Systems and Their Applications, 9(4):25–26, 1994.
[22] Bundy, A. *Incidence calculus: A mechanism for probabilistic reasoning*. Journal of Automated Reasoning, 1:263–283, 1985.
[23] Bundy, A. *Incidence calculus*. The Encyclopedia of AI, pages 663–668, 1992.
[24] Liu, W. and Bundy, A. *On the relations between incidence calculus and ATMS*. In Proc. of the second European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU93), Lecture Notes in Computer Sciences 747, pages 249–256, Granada, Spain, 1993. Springer.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Information Evaluation: A Formalisation of Operational Recommendations

Vincent NIMIER
*ONERA Centre de Châtillon BP 72, 29 avenue de la Division Leclerc,*
*92322 Châtillon, France, nimier@onera.fr*

**Abstract.** An important step in the intelligence gathering process is the fusion of information provided by several sources. The objective of this process is to build an up-to-date and correct picture of the current situation with the overall available information in order to make adequate decisions. In the framework of intelligence, as opposed to the air defense domain for example, there is no real automatic process and the fusion of information is performed by human operators. The STANAG 2022 proposes a methodology for the evaluation of information in this framework of intelligence and for human processing. However, given the increase in the amount of information that the human operator must process, some automatic processing is being considered. Therefore there is a need for a formal methodology for the evaluation of information. In this paper, we present some considerations about the correspondence between the STANAG 2022 information evaluation methodology and a formal and mathematical evaluation approach given by an automatic process.

**Keywords.** STANAG 2022, information evaluation, reliability, credibility, conflict

## 1. Introduction

An important step in the intelligence gathering process is the fusion of information provided by several sources. The objective of this process is to build an up-to-date and correct view of the current situation using the overall available information in order to make adequate decisions. To succeed in this process, it is important to associate, with each available information, some attributes like the number of the sources that support it, their reliability, the degree of truth of the information etc. The Standardization Agreements (STANAG) 2022 of the North Atlantic Treaty Organization (NATO) proposes some elements of a framework for such common definitions. However, actually, in the operational intelligence process, these attributes are managed by the human operator when fusing information provided by the different sources. Indeed, there is no real methodology to do this in a formal manner. When relaying this fusion process to a machine, we need to develop formal definitions and algorithms to manage these attributes in addition to fusing information. Furthermore, in a context of interoperability where different systems exchange information, common definitions of these attributes shall be shared.

The purpose of this paper is first to analyze the STANAG 2022 recommendations about information evaluation and then to set a first step in the definition of a formal and

non-ambiguous system for evaluating information. Indeed, as it will be shown, the present recommendations, written in natural language, are rather ambiguous, imprecise and open to discussion. Secondly, knowing that in general an automatic fusion process is based on some mathematical theories such as probability, possibility, and Dempster-Shafer, we propose some correspondence between the result that this automatic process may provide and an operational evaluation in accordance with the evaluation process described in the STANAG 2022.

This paper is organized as follows. Section 2 presents a review of STANAG 2022 recommendations and points to the main notions that underline these recommendations. Section 3 analyzes the different assumptions in the evaluation proposed in the STANAG. Section 4 presents the link between the concepts described in the STANAG and the results of automatic processing based on the use of mathematical theory. Section 5 presents a model for a data fusion process which takes into account the notion of reliability. Section 6 presents the conclusions of this work.

## 2. Review of STANAG 2022 Recommendations

The Annex to STANAG 2022, Edition 8 ([1]) explicitly mentions that the aim of information evaluation is to indicate the degree of confidence that may be placed in any item of information which has been obtained for intelligence. This is achieved by adopting an alphanumeric rating system. This system combines a measurement of the reliability of the source of information with a measurement of the credibility of that information when examined in the light of existing knowledge.

Upon examining the whole text we should point out that the two main concepts in this evaluation system are the reliability of the sources and the credibility of the information. These concepts are defined in the STANAG 2022 recommendations, as follows:

The reliability of a source is designated by a letter between A and F signifying various degrees of confidence as indicated below.

- a source is evaluated A if it is completely reliable. It refers to a tried and trusted source which can be depended upon with confidence;
- a source is evaluated B if it is usually reliable. It refers to a source which has been successfully used in the past but for which there is still some element of doubt in particular cases;
- a source is evaluated C if it is fairly reliable. It refers to a source which has occasionally been used in the past and upon which some degree of confidence can be based;
- a source is evaluated D if it is not usually reliable. It refers to a source which has been used in the past but has proved more often than not unreliable;
- a source is evaluated E if it is unreliable. It refers to a source which has been used in the past and has proved unworthy of any confidence;
- a source is evaluated F if its reliability cannot be judged. It refers to a source which has not been used in the past.

Credibility of information is designated by a number between 1 and 6 signifying varying degrees of confidence as indicated below.

- if it can be stated with certainty that the reported information originates from another source than the already existing information on the same subject, then it is classified as "confirmed by other sources" and rated 1;
- if the independence of the source of any item of information cannot be guaranteed, but if, from the quantity and quality of previous reports, its likelihood is nevertheless regarded as sufficiently established, then the information should be classified as "probably true" and given a rating of 2;
- if, despite there being insufficient confirmation to establish any higher degree of likelihood, a freshly reported item of information does not conflict with the previously reported behavior pattern of the target, the item may be classified as "possibly true" and given a rating of 3;
- an item of information which tends to conflict with the previously reported or established behavior pattern of an intelligence target should be classified as "doubtful" and given a rating of 4;
- an item of information which positively contradicts previously reported information or conflicts with the established behavior pattern of an intelligence target in a marked degree should be classified as "improbable" and given a rating of 5;
- an item of information is given a rating of 6 if its truth cannot be judged.

The previous definitions of information evaluation can be questioned. Indeed, since they are given in natural language, they are quite imprecise and ambiguous, and many points are open to discussion. Furthermore, this evaluation is given in a discrete scale while generally automatic processing gives evaluation results in a continuous scale between 0 and 1.

However, even if the STANAG recommendations are a bit vague, they still present three basic concepts that form a cornerstone in the evaluation system. These concepts are the following:

- the reliability of a source;
- the number of independent sources that support an information;
- the fact that the information tends to conflict with some available information.

It is clear that these three concepts are independent and that they have to be included in all quotation systems, whether these systems are automatic or not.

## 3. Analysis of the Different Evaluation Concepts

### *3.1. Basic Concepts*

The concepts used in the STANAG were introduced in an operational framework with a human operator working on the fusion process. In order to automate some tasks of the fusion process, we need to find some correspondence between operator reasoning and the result of automatic processes using mathematical theory. For this we attempt in this section to give a more comprehensive and formal definition of reliability and credibility based on some mathematical theory.

### *3.2. Reliability*

In mathematical logic, the reliability of a source is defined in a binary way as follows: *an information source is (totally) reliable if and only if the information it delivers is true in the real world.* For instance, a sensor which measures the temperature is totally reliable if and only if the temperature it indicates is the correct one; a human expert is totally reliable if and only if any information (opinion, conjecture etc) he gives is true.

According to the recommendations of STANAG 2022, the reliability of a source is not a binary notion but a notion of scale and is defined in reference to its use in the past. It can be measured, for example, as the ratio of the number of times the source gave true information to the number of times it gave information. However, this definition does not take into account the actual environment of use of the information source. For instance, even if it is known to be reliable, an infra-red sensor loses reliability when it rains. So the conditions of use of the source must be taken into account.

It must also be noticed that a consensus has not yet been reached as to a formal definition of reliability. For instance, we can read in the APJ 2.1 "*every piece of information produced by an impeccable source is not necessarily correct.*" If "impeccable" intends to mean "reliable," this sentence is contradictory to the definition given previously. Here, it implies that the reliability of a source is not defined by its ability to deliver truth and that even a reliable source can be wrong. A source can be wrong not because it is not genuine but because its model may be miss-adapted to the situation.

To give some mathematical models to the above considerations, we note **A** the event "the source is used in favorable condition." That means that it is used such that its performances are well known and are the best possible. In other words, it gives true information when possible. So the reliability can be modeled by the probability P(**A**) which is equal to 1 if it is totally reliable and equal to 0 if it is unreliable. Furthermore, we will see how, thanks to this model, the sentence: "*every piece of information produced by a* ***totally reliable*** *source is not necessarily* ***true***" can be modeled.

### *3.3. Credibility*

The credibility of an information is a more natural concept for a mathematical information fuser. Indeed it can be considered as the likelihood function of the information. This credibility is evaluated according to the following two basic concepts.

#### *3.3.1. Conflict*

This paragraph addresses the concept of conflict with a focus on its use in probability. First we define the concordance function using a kronecker product. The conflict function is a straightforward derivation of this last function. Secondly, we expand this function to take into account the notion of uncertainty in the decision process.

Let D define a set of hypotheses $D=\{H_1, \ldots, H_n\}$. After observation, a declaration $d=i$, $i \in \{1, \ldots, n\}$ is given by a source. Then the conflict between each hypothesis and the declaration can be simply defined by the relation:

*The declaration $d=i$ is in conflict with an hypothesis $H_j$ if $i \neq j$. If $i=j$, we say the declaration concords with the hypothesis.*

From a strict mathematical point of view, we can define concordance by the mathematical function $\delta_{ij}$ which is equal to 1 if i=j and equal to 0 otherwise. With this definition, the conflict can be described by the function $1 - \delta_{ij}$.

These defined functions are binary. Moreover, we can give some degrees to the concordance or the conflict between the declaration and an hypothesis by defining the probability $P(d=i/H_j)$. In numerous cases the concordance function is maximum when the declarations concord (i=j) and at the limit $P(d=i/H_i)=1$, this implies that $P(d=i/H_j)=0$ if $i \neq j$.

*3.3.2. Independence of Sources*

In this paper we consider that two sources are independent if they are two physical entities. Two informations are not independent if they come from the same source at different time periods.

## 4. Definition of a Mathematical Function for Evaluation

As we have seen in section 3.2, the concept of reliability can be directly modeled by a probability. Credibility is more difficult to define because it encompasses the notion of conflict in addition to the number of sources.

Let us now consider a simple model for a "closed world" assumption and an exclusive decision process. The "closed world" means it is possible to establish the set of object classes and this set is exhaustive. We denote $D= \{H_1, \ldots, H_n\}$ this set of classes. An exclusive decision process means that all informations given by a source correspond to a decision about one and only one elementary hypothesis.

In the following we consider the general idea:

Under the closed world assumption, we will not consider the rating of 6 for the credibility. The reason is straightforward: if an hypothesis belongs to the frame of discernment that means this hypothesis is possible, if not, we do not need to put it in this frame. Thus in the following, we will consider that if no information is available on one hypothesis, the credibility of this hypothesis is equal to 3. In the same spirit, if one information confirms one hypothesis then the credibility of this hypothesis equals 2 and if two informations confirm one hypothesis the credibility of this hypothesis will be equal to 1. Then the scale given to the credibility is between 1 and 5.

*4.1. Mono Hypothesis Approach*

In this section we suppose that an automatic process fuses information based on one of the following approaches based on mathematical theory. The objective is that this process provides results according to the recommendations such that an operator can manage the results with other information given by human intelligence.

*4.1.1. Heuristic Approach*

Based only on a priori knowledge we suppose that all hypotheses will have a credibility of 3. Because no sources except for a priori knowledge are available, the reliability of the source can be evaluated by F. In the following, under the assumption of the closed

world and with only a priori information, we assume that the credibility of an hypothesis, relative to the identification of a fact, is equal to 3, and the reliability of the source is equal to F.

If an information $d=H_i$ given by a source, arrives at the fusion process, the hypothesis $H_i$ will have a credibility of 2 "probably true" and the other hypotheses $H_j$, for $j \in \{1,\ldots n\}$, with $j \neq i$ will have the credibility of 4 "doubtful." If a second information $d=H_j$ arrives at the fusion process, two cases are then possible. The second information confirms the first one and then $H_i$ will have the credibility of 1 "confirmed." In this case, the other hypothesis will have the credibility of 5 "improbable." In the case where the two informations are different, both informations conserve the credibility of 2 and the others are left with a credibility of 4. In the case of a third or more informations arriving at the fusion process, we can derive a general procedure.

Let $Q(H_i)$ be the number of informations $d = H_i$, provided by a set of sources that confirm the hypothesis $H_i$. Let us suppose that the p sources give one information. Then:

If $Q(H_i) = p$ the credibility of this hypothesis is 1 and the credibility of the others is 5.

If $Q(H_i) \in \{1, \ldots, p-1\}$ the credibility of this hypothesis is 2 and the credibility of the hypothesis with $Q(H_i)=0$ is equal to 4.

If no information is available about any of the hypotheses, $Q(H_i)=0$ for all of the hypotheses and their credibility is equal to 3.

### *4.1.2. Probabilistic Approach*

The procedure in the previous section is not totally satisfactory and becomes ill suited when the number n of delivered informations is large. The main reason is that the evaluation of the credibility of an information cannot be restrained to a scale of five. In this section we propose the use of the probability theory which allows a continuous scale from 0 to 1. Let $P(H_i) = q/p$ the number of informations that confirm a hypothesis over the number of sources that give informations. The previous procedure can be rewritten as follows:

If $P(H_i) = 1$ the credibility of this hypothesis is 1 and the credibility of the others is 5.

If $P(H_i) \neq \{0,1\}$ the credibility of this hypothesis is 2 and the credibility of the other hypotheses with $P(H_i)=0$ is equal to 4.

If no information is available about any of the hypotheses the credibility is equal to 3.

In this framework of probability theory, an additional extension of the first procedure can be made by considering that the source not only provides an information of the form $d=H_i$ but it also provides an evaluation of the credibility based on its own sources of information and knowledge. In this case, a source k provides information of the form $P(d_k= H_i/H_j)$ with $j \in \{1, \ldots,n\}$. This information is not a unique decision but a vector of decisions with their probabilities. If many sources are available a fusion center can fuse the information given by the sources and calculate a probability of the form: $P(H_i/ d_1, \ldots,d_n)$. Then the credibility can be given by:

If $P(H_i/ d_1, \ldots,d_n) = 1$ the credibility of this hypothesis is 1 and the credibility of the others is 5.

If $P(H_i/ d_1, \ldots, d_n) \neq \{0,1\}$ the credibility of this hypothesis is 2 and the credibility of the hypotheses with $P(H_i/ d_1, \ldots, d_n) = 0$ is equal to 4.

If no information is available about any of the hypotheses the credibility is therefore equal to 3.

In a practical situation the first and the third cases nearly never occur. We can adopt some heuristic approach to determine a threshold s so that the procedure becomes:

If $P(H_i/ d_1, \ldots, d_n) > s$ the credibility of this hypothesis is 1 and the credibility of the other hypotheses is 5.

If $P(H_i) \neq \{1-s, s\}$ the credibility of this hypothesis is 2 and the credibility of the hypotheses with $P(H_i) < 1-s$ is equal to 4.

If no information is available about any of the hypotheses the credibility is equal to 3.

### *4.1.3. Possibility Approach*

Thanks to the possibility theory framework, we can note that the definition of the credibility of one hypothesis is closely related to the credibility given to the other hypotheses in the same frame of discernment. For example in the first procedure, if a hypothesis has a credibility of 1 the others have the credibility of 5. This consideration allows the procedure to be rewritten in terms of possibility and necessity as follows:

| | | |
|---|---|---|
| If $\Pi(H_i) = 1$ $N(H_i) = 1$ | then the credibility is equal to 1 | "confirmed" |
| $\Pi(H_i) = 1$ $0 < N(H_i) < 1$ | then the credibility is equal to 2 | "probable" |
| $\Pi(H_i) = 1$ $N(H_i) = 0$ | then the credibility is equal to 3 | "possible" |
| $0 < \Pi(H_i) < 1$ $N(H_i) = 0$ | then the credibility is equal to 4 | "doubtful" |
| $\Pi(H_i) = 0$ $N(H_i) = 0$ | then the credibility is equal to 5 | "improbable" |

where $\Pi(H_i)$ and $N(H_i)$ are the possibility and the necessity of the hypothesis $H_i$. The evaluation of these two terms is application dependent. A possible way to gain coherence with the examples given above is the following:

If no information is available then $\Pi(H_i)=1$ and $N(H_i)=0$.

If p is the number of independent sources and n the number of decisions for the hypothesis $H_i$ then $\Pi(H_i)=1$ and $N(H_i)=n/p$.

### *4.2. Multi Hypothesis Approach Dempster-Shafer Theory*

In the previous section, the considered adopted model is very simple: the frame of discernment is closed and the decisions given by the sources have a one to one correspondence with each hypothesis of this frame. In the following, we adopt a more realistic model. We suppose the decision can take its value in a subset of hypotheses. For example the frame of discernment D represents all terrestrial vehicles. A source may give an information of the form "the object is an armored vehicle" representing a subset of D composed by the elementary hypotheses {AMX10, Char Leclerc, ….}. For this application framework we will use the Dempster-Shafer theory. Let D be the frame of discernment and $d_i$ the decision given by a source which takes its value in $2^D$. If Pl and Cr represent the plausibility and the credibility respectively, then the operational evaluation is given by the following relations:

| | | |
|---|---|---|
| If $Pl(d_i)=1$ $Cr(d_i)=1$ | then the operational credibility is equal to 1 | "confirmed" |
| $Pl(d_i)=1$ $0<Cr(d_i)<1$ | then the operational credibility is equal to 2 | "probable" |
| $Pl(d_i)=1$ $Cr(d_i)=0$ | then the operational credibility is equal to 3 | "possible" |
| $0<Pl(d_i)<1$ $Cr(d_i)=0$ | then the operational credibility is equal to 4 | "doubtful" |
| $Pl(d_i)=0$ $Cr(d_i)=0$ | then the operational credibility is equal to 5 | "improbable" |

In the relation above we make the distinction between the credibility given by the Dempster-Shafer theory and noted Cr and the operational credibility given by STANAG 2022.

This approach based on the Dempster-Shafer theory makes it possible to consider the case of two sources working with different granularity of classification. For example, one source may send an information of the form "this object is an AMX 10" and an other source may give an information of the form "this object is an armored vehicle." The sources refer to the same object. In this case the question is: what information must be considered and what operational credibility should be given to this information? If we know that an AMX 10 is an armored vehicle then we can consider that the information "armored vehicle" is confirmed by the sources and has a rating of 1 and the information "AMX 10" has only a rating of 2. This phenomena is naturally taken into account by the theory of Dempster-Shafer.

## 5. Model for Data Fusion

### *5.1. Introduction*

In the previous section, we defined in a concrete manner the concept of evaluation in the frameworks of probability theory, possibility theory and Dempster- Shafer theory. We will now study the use of this evaluation rating in a fusion process. We limit ourselves to the framework of probability theory.

### *5.2. Probability*

We consider the architecture figure 1, where two sensors $C_o$ and $C_r$. are linked to a fusion center FC.

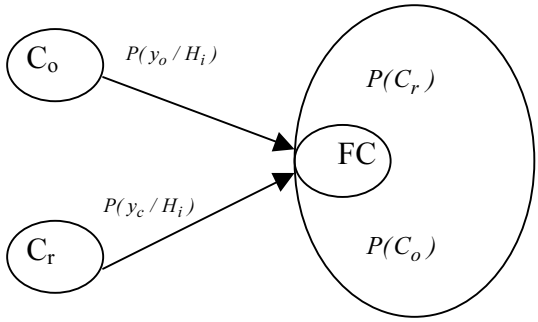



**Figure 1.** STANAG 2022.

Each sensor has its own model. Then it is possible for each sensor to send its credibility under the form of probability. If the fusion center is automated (figure 1) or under the form described by the STANAG 2022 (thanks to the conversion given section 4.1 if the fusion center is an operator). We suppose that the fusion center knows the models of each of the sensors.

Therefore it knows each of their limits and can deliver a reliability in the form of a probability $P(C_o)$ or $P(C_c)$. It follows that for an automatic process the fusion including the use of reliability is given by the formula:

$$P(H_i / y_o, y_r) = P(H_i / y_o, y_r, C_o, C_r) P(C_o \cap C_r) + P(H_i / y_o, y_r, \overline{C_o}, C_r) P(\overline{C_o} \cap C_r) + P(H_i / y_o, y_r, C_o, \overline{C_r}) P(C_o \cap \overline{C_r}) + P(H_i / y_o, y_r, \overline{C_o}, \overline{C_r}) P(\overline{C_o} \cap \overline{C_r})$$

where the probability of each hypothesis is given by a formula of the form

$$P(H_i / y_o, y_r, \overline{C_o}, C_r) = \frac{P(y_o / H_i, \overline{C_o}) P(y_r / H_i, C_r) P(H_i)}{P(y_o, y_r / \overline{C_o}, C_r)}$$

for example in the second term, the reliability coefficient of the sensors $P(\overline{C}_o \cap C_r)$ can be decomposed into individual reliabilities based on:

$$P(\overline{C}_o \cap C_r) = (1 - P(C_o)) P(C_r)$$

For the likelihood of each hypothesis there are two terms:

The first one $P(y_o / H_i, C_o)$ is derived based on the internal sensor model. The second one $P(y_o / H_i, \overline{C_o})$ is the likelihood of the hypothesis $H_i$ knowing that the sensor is out of the range of its validity domain. This function is defined by the fusion center and depends on the strategy adopted.

After the probability of each hypothesis is calculated the evaluation process given in section 4.1 may be applied to be in accordance with the STANAG and then manageable by a human operator.

## 6. Conclusions

This work intends to formalize some STANAG recommendations about information evaluation in the fusion process. Some of the informal concepts underlying the recommendations have been given a formal interpretation. However, as it has been shown, no consensus exists yet on these definitions and more work is needed. We presented a simple version of a fusion process taking into account the two concepts of credibility and reliability as recommended in the STANAG 2022 but based on a classical mathematical tool such as probability theory. The correspondence between mathematical results and the operational recommendations was given such that the fusion process is coherent in a chain composed of both the automatic fusion process and the human fusion process.

## References

[1] Annex to STANAG 2022 (Edition 8). North Atlantic Treaty Organization (NATO) Information Handling Services, DODSID, Issue DW9705.
[2] S. Konieczny and R. Pino-Pérez. On the logic of merging. In Principles of Knowledge Representation and Reasoning (KR'98), pages 488–498, 1998.
[3] S. Konieczny and R. Pino-Pérez. Merging with Integrity Constraints In Proceedings of ESCQARU'99.
[4] C. Baral and S. Kraus and J. Minker and V.S. Subrahmanian. Combining multiple knowledge bases. IEEE Trans. on Knowledge and Data Engineering, 3(2), 1991.
[5] L. Cholvy. "Proving theorems in a multi-sources environment." Proceedings of IJCAI, 1993.
[6] C. Liau. "A conservative Approach to Distributed Belief Fusion." Proceedings of FUSION, 2000.
[7] V. Nimier. "Supervised Multisensor tracking algorithm," FUSION-98, International Conference on Multisource-Multisensor Information Fusion, Las Vegas, USA, July 6–9, 1998, pp. 149–156.
[8] V. Nimier. "Introducing Contectual Information in Multisensor tracking Algorithms" Advances in intelligent Computing, Springer, 1994.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Fusion Techniques for Airborne and Spaceborne Identification of Objects

## *Imaging and Non-Imaging Sensor Fusion*

Dr. Pierre VALIN
*Senior Member of Engineering, R&D Department, Lockheed Martin Canada,*
*6111 Royalmount ave., Montréal, QC, H4P 1K6, Canada*

**Abstract.** Information fusion from dissimilar sensors is best performed through extraction of attributes that can be measured by each of these sensors. In this way both imaging (Synthetic Aperture Radar (SAR), and Forward Looking Infra-Red (FLIR)) and non-imaging sensors (Identification Friend or Foe (IFF), Electronic Support Measures (ESM), radar) can be treated on an equal footing. In order to properly identify the target platform through repeated fusion of identity declarations, the attributes measured must be correlated with known platforms through a comprehensive *a priori* platform database. This comprehensive database is carefully analyzed for attributes that can be provided by sensors and additional knowledge that can be interpreted at all levels of fusion. The identity (ID) of the target platform can be provided in a hierarchical "tree" form where leaves are unique IDs but branch nodes correspond to a taxonomy obeying certain standards. In some cases, precise attribute measurement is either impractical or of moot value, so fuzzification is performed through appropriate membership functions. The actual identity of tracked ships is performed by an algorithm utilizing the Dempster-Shafer theory of evidence. The algorithm can mathematically handle conflict, which possibly appears as the result of countermeasures and/or poor associations, and ignorance, which may be present in the cases when sensors provide ambiguous or hard to interpret results. Since each imaging sensor has its own measurement potential, customized classifier solutions must be designed for optimal performance. A series of FLIR classifiers is presented and fused through a neural net fuser, while a hierarchical SAR classifier is shown to perform well for combatant ships, which are most likely to be imaged by the SAR. The complete fusion solution is demonstrated in a series of realistic scenarios involving both friendly and enemy ships.

**Keywords.** Attributes, platform database, fuzzification, Dempster-Shafer evidential reasoning, SAR, FLIR, imagery classifiers, fusion

### 1. Introduction

Reasoning over attributes (or situations) plays a big role in military domains, where complementarity of sensor information can lead to quicker and more stable identification (ID) through information fusion. The focus of this lecture is an application of reasoning over attributes to airborne surveillance of ships by both passive (or covert) and active sensors to properly identify targets in a hostile environment.

The attributes, over which one has to reason during the single Object Refinement phase of fusion (level 1), can originate from imaging or non-imaging sensors, and be

kinematical, geometrical or relate more directly to the ultimate finding of a unique ID (if those come from intelligent sensors such as Identification Friend or Foe (IFF), Electronic Support Measures (ESM), or imagery classifiers). The concept of a unique ID itself contains an ambiguity related to the desired taxonomy, over which reasoning occurs, and this choice depends on the operator's need for a decision aid tool. This taxonomy can span all levels of fusion, and appropriate attributes must be found also for fusion Levels 2 and 3 (Situation Refinement, and Implication Refinement respectively). These 2 levels can also be referred to as Situation and Threat Assessment (STA) and Resource management (RM). An example *a priori* platform database (PDB) is discussed that contains platform characteristics, with supporting databases covering emitter-to-platform mapping, geopolitical allegiance information (which can be mission-specific), etc. In addition, physical attributes often require fuzzification for proper treatment by evidential reasoning, and fuzzy logic is just one of many reasoning frameworks that can be applied to the task.

When sensors provide imagery information, a certain amount of pre-processing is required at the sub-object refinement stage (Level 0 of fusion), whose main task is to do feature extraction for attribute determination. These imagery classifiers digest this very specific attribute information separately from the information contained in the PDB, mainly because classifier design and implementation has to remain flexible for continuous improvement, while the a priori PDB should remain fairly static.

## 2. Attribute Measurements

For ID estimation to be properly achieved in Level 1 DF, all possible attributes that can be measured by all of the sensors must be listed in the PDB. These can be split into three groups: kinematical, geometrical, and identification attributes, as explained further below.

Kinematical attributes can be estimated through tracking in the *positional estimation* function of DF, and through reports from IFF and datalink. Since the tracker can provide speed, acceleration and sometimes altitude, attributes such as maximum (max) acceleration, max platform speed, minimum (min) platform speed, cruising speed, and max altitude either serve as bounds to discriminate between possible air target IDs or suggest the plausible IDs. However, speed reports should be fused only if they involve a significant change from past historical behavior in that track. The reason is two-fold:

1. first, no single sensor must attempt to repeatedly fuse identical ID declarations; otherwise the hypothesis that sensor reports are statistically independent is violated;
2. second, the benefits of the fusion of multiple sensors are lost when one sensor dominates the reports.

Geometrical attributes can be estimated by algorithms which post-process imaging information from sensors such as Forward Looking Infra-Red (FLIR), or Electro-Optics (EO) and Synthetic Aperture Radar (SAR). Classifiers that perform such post-processing can be thought of as Image Support Modules (ISM) performing much the same functionality as the ESM does for the analysis of electromagnetic signals. These ISMs can provide the three geometrical dimensions of height, width and length (for FLIR and EO), and also Radar Cross Section (RCS) of the platform as seen from the front, side or top (for the SAR, and radiometric radar). In addition, the distribution of

relevant features may be needed for classifiers, but may be considered part of the algorithms that generate plausible IDs.

Identification attributes can be directly given by the ESM, as outputs of the FLIR and SAR ISM, from acoustic signal interpretation (for surface and sub-surface targets), and from Doppler radar (for airborne targets). The ESM requires an exhaustive list of all the emitters that are carried by the platform, since it will provide an emitter list with some confidence level about the accuracy of the list that reflects the confidence in its electromagnetic spectral fit. However an IFF response can lead to an identification of a friendly or commercial target but the lack of a response does not necessarily imply that the interrogated platform is hostile. One has to distribute the lack of a response between at least two declarations: the most probable foe declaration and a less probable friendly or neutral declaration that allows for an IFF equipment that is not working or absent. On the other hand, the ISMs are usually designed not only to provide the best single ID possible but also to estimate confidence in higher levels of an appropriate taxonomy tree.

Note that each physical quantity has a different dimension (speed, acceleration, length, RCS, etc.) and that an accurate determination is not necessarily needed for fusion. Indeed it is convenient, for example to bin the attribute "speed" into fuzzy classes like "Very Fast," "Fast," "Medium," "Slow" and "Very Slow" (separately for air and surface targets). Indeed, spreading the mass corresponding to a given measurement of a physical quantity over several such bins, is a special case of a more complex fuzzy membership function, as illustrated in Figure 1 below.

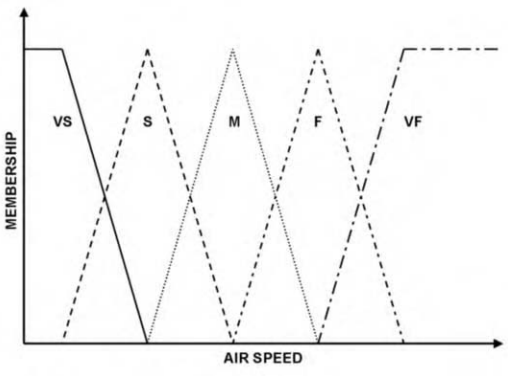



**Figure 1.** Example of fuzzification of speed.

In addition, a measured value of speed only indicates that the target is capable of that speed, not that it corresponds to either the maximum or minimum speeds listed in the PDB, but more likely to the cruising speed. Finally, fuzzification can be justified by acknowledging that a given value of a physically measured quantity (say, speed) can have different interpretations according to various military experts.

## 3. The Set of a Priori Databases

To summarize, from the above considerations of attributes, the PDB must contain at least the following level 1 attributes:

1. *Kinematical*: maximum velocity > cruise velocity > minimum velocity (only for air targets), maximum altitude (for air targets), maximum depth (for sub-surface targets), maximum acceleration (for air targets).
2. *Geometrical*: length, height, width, RCS (top), RCS (side), RCS (front).
3. *Identification*: propulsion type, number of propellers, number of blades, number of cylinders, and an exhaustive emitter list.

These attributes are listed for each individual platform ID, of which there can be many produced entities, undecipherable except by visual inspection at close ranges (Halifax class frigate, CP-140, Tribal class destroyer, EH-101, etc.). These individual IDs must be further organized in a taxonomy tree, usually derived from some standard, either STANAG 4420 or MIL-STD 2525 (A or B), and which contains at the very least the category (air, surface, subsurface, land), type (air fixed wing, ship combatant, helicopter, submarine, etc.), subtype (transport, attack, support, training, etc.), and owner (country).

The Levels 2 and 3 databases, i.e. the STA/RM DB should contain all the platform parameters relevant for STA as well as RM, i.e., missiles (number and detailed properties) on enemy ships are relevant for STA, while the same information on possible own-platforms is relevant for RM. In a Network Centric Warfare (NCW) context, the lethality of enemy platforms in the red force is important for STA, and the lethality of cooperating Participating Units (PUs) is relevant for RM within the blue force. A non-exhaustive list contains elements pertaining to:

1. *Platform and Mission*: displacement, number of operational copies of the platform, list of hull numbers & names (if it can be provided for ID in harbors, air fields, etc.), range of deployment, platform type (with amplification), and role, crew (for full operation).
2. *Armament* (type and number of examples present on platform, both HW as well as humans for mission deployment): Surface-to-Surface Missiles (SSMs), Surface-to-Air Missiles (SAMs), Close-In Weapon Systems (CIWSs), Air-to-Surface Missiles (ASMs), Air-to-Air Missiles (AAMs), CIWSs, conventional bombs, troop complement (number of special force for assault, landing or parachuting), lethality, guns, torpedo tubes.
3. *Sensors* (mostly passive, in order to estimate probability of own-platform detection, excluding the radar already in PDB): Infra Red Search and Track (IRST), sonars (e.g., Hull-Mounted Sonar, towed-array, sonobuoy, tethered sonar), and imaging sensors (e.g., EO, FLIR, SAR).
4. *Air Platforms on Deck* (for surface ships): Number of helicopters, aircraft.

Although the PDB is the most important of the a priori databases and is rarely modified, there are supporting databases which are necessary and flexible, such as the Geo-Political Namelist (GPL), which contains acronyms of the countries that own platforms in the PDB, and which provides the country's current allegiance in the conflict (depending on the mission: friend, foe or neutral). If language communications can be intercepted, the languages spoken in different countries are also listed. The Emitter Namelist (ENL) offers a correspondence between the index number of each emitter and its name (actual, or NATO), various characteristics (Pulse Repetition Frequency (PRF), frequency, detection range, etc.) and can offer (optionally) a cross-listing of platforms on which each emitter can be found.

## 4. Identification Fusion

It has been shown that sensors provide attributes, which can be cross-correlated through the PDB with a set of platforms consistent with that attribute. Let us call the set of such IDs a "proposition" $I=\{P_i, P_j, \ldots P_n\}$ with an appropriate associated probability. If the attributes are fuzzified through membership functions, then a declaration can be made consisting of a set of propositions, each with its own probability of being correct. This can also happen with IFF reports, and with results from ISMs. The choice of a method for combining such propositions depends on such factors as:

a) must process incomplete information ==> Notion of *ignorance*
b) sensor performance is not always monitored ==> Notion of *uncertainty*
c) must handle conflicts between contact/track ==> Notion of *conflict*
d) must not require a priori information ==> No Bayesian reasoning
e) real-time method ==> Possibility of truncation is required
f) operator wants best ID ==> Give preference to single ID (singleton)
g) operator wants next best thing ==> Doublet (2 best IDs), triplet, etc…
h) must resist Counter Measures ==> Conflict again (emitter is not in PDB)
i) must resist false associations ==> ESM report associated to wrong track
j) must be tested operationally ==> Complex scenarios needed
k) ordinary method must explode ==> Large complex PDB needed

Thus, one requires a reasoning method where *ignorance, uncertainty* and *conflict* have mathematical meaning, which is robust, and which can be simplified to reduce computational complexity. It is well known that incomplete, uncertain, and sometimes conflicting information, is ideally suited to Dempster-Shafer (DS) evidential reasoning, where "mass" or Basic Probability Assignment (BPA) plays the role of the probability. Indeed, when set intersection is null for certain combinations between the new contact and the existing track, conflict exists. Furthermore, when one is uncertain of the correctness of the declared proposition, and its associated probability, it is wise to assign a small mass to the ignorance, as well as a best estimate for the larger mass of the declared proposition. Finally, there exists a well-documented and tested truncation method to keep the method real-time [2,4–14]. The resulting combined propositions can then be ranked according to their small size (singletons first) and their mass value, as one wishes.

The important DS combination rule is called the Orthogonal Sum (OS) and involves renormalization due to the conflict $K$

$$m(C_k) = \sum_{i,j\ni A_i \cap B_j \equiv C_k} \frac{m(A_i)m(B_j)}{1-K} \qquad K = \sum_{A_i \cap B_j \equiv \Phi} m(A_i)m(B_j) \tag{1}$$

In the case of the PDB having a hierarchical taxonomy tree, the OS can be replaced by a Hierarchical OS (HOS), which redirects the combined mass to the first non-conflicting level of the tree [1], rather than renormalizing.

DS also provides the possibility of calculating other quantities from the masses of the set of combined propositions, such as the belief *Bel(A),* which gives the lower probability of a set A (composed of several subsets), and the plausibility *Pls(A),* which gives its upper probability. Decisions can then be made either on the probability di-

rected to the set A itself (its mass), the belief in A, which is the sum of all evidence supporting A, or the plausibility of A, which is unity minus the evidence against A:

$$Bel(A) = \sum_{A_i \in A} m(A_i) \qquad Pls(A) = 1 - Bel(\neg A) \tag{2}$$

## 5. Imagery Classifiers

Because of the very different physical characteristics of the imaging sensors usually present on airborne surveillance aircraft (such as the Canadian CP-140 Aurora version of the USA's P3-C), one should expect imagery classifiers to be very customized ISM, and indeed they are.

The FLIR is mainly a short-range passive sensor, providing low contrast, or low Signal-to-Noise Ratio (SNR) images with sometimes significant artifacts, used when covert operations are required. It is often used on merchant ships at short distances. On the other hand, the SAR is a long-range active sensor with very high SNR, whose main purpose is to detect enemy ships at standoff distances. It has many modes, 2 for land use (StripMap and Spotlight Non-Adaptive), and 2 for sea (Range Doppler Profiling and Spotlight Adaptive). For the purposes of this lecture the Sea Spotlight Adaptive mode is used on ships, while in other lectures of this series, the StripMap mode is usually of interest.

### *5.1. FLIR ISM Classifiers and Their Fusion*

The presently available data set is composed of 2545 FLIR ship images [15,16], each of which belongs to one of the following 8 types (originally classes):

1. Destroyer
2. Container
3. Civilian Freighter
4. Auxiliary Oil Replenishment
5. Landing Assault Tanker
6. Frigate
7. Cruiser
8. Destroyer with Guided Missile

However, these types are not uniformly represented (type 3 is the least at 7%, and type 4 the most at 17%). The images are randomly divided into a training set and a test set, and the sizes of those are varied between 1,000 and 1,500 images to test the sensitivity (which is shown to be small a posteriori). For every image, the ship silhouette was thresholded manually. Figure 2 shows typical silhouettes for the 8 types.

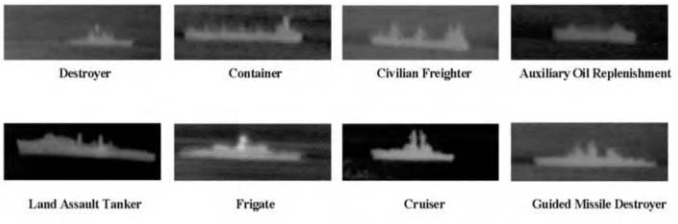



**Figure 2.** Ship silhouettes for the 8 classes of ships.

The features (or attributes) used are 7 moment functions about the center of mass which are invariant under translation, rotation and scale, because the images are respectively not necessarily centered, nor always in the same aspect angle, and the zoom factor is unrecorded. Since these moments deliver information primarily of the global shape of the object and represent poorly the details of the object, 4 more are extracted by fitting an auto regressive model to a one-dimensional sequence of the projected image along the horizontal axis. Figure 3 below gives the frequency distribution of attribute 1, which is related to the mean two-dimensional relative size.

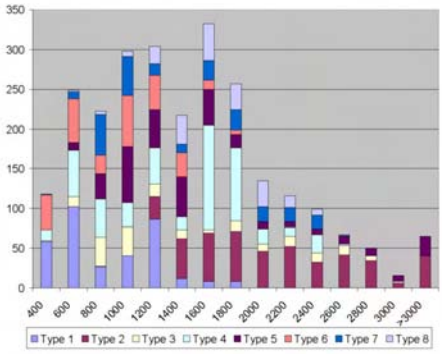



**Figure 3.** Frequency graph for attribute 1.

Such frequency graphs for each of the 11 attributes contribute independently their interpretation of which type is represented by the image, with the final classification being obtained by the class with the highest score. Four classifiers have been designed, which are in order of increasing performance:

1. DS, with an average performance of 75%
2. Bayes, with an average performance of 77%
3. K-nearest neighbors, with an average performance of 93–95% for different K values and different distance measures
4. Neural Net (NN), with an average performance of 93%

In the case of DS, the frequency graphs yield the masses used in 10 consecutive fusion cycles, each reporting quite a lot of conflict, and a priori distributions are not used. Significant improvements can be made if individual DS classifiers are obtained by selecting different sets of features giving better results for all the types than the generic DS classifier, as shown below in Table 1.

**Table 1.** Generic DS performance versus individual DS classifier performance

| | Generic | Individual |
|---|---|---|
| Class 1 | 0.871 | 0.932 |
| Class 2 | 0.864 | 0.958 |
| Class 3 | 0.070 | 0.242 |
| Class 4 | 0.957 | 0.984 |
| Class 5 | 0.629 | 0.747 |
| Class 6 | 0.735 | 0.803 |
| Class 7 | 0.490 | 0.686 |
| Class 8 | 0.885 | 0.928 |
| Total | 0.745 | — |

In the case of the Bayesian classifier, each attribute independently votes for the type, by using a sum, rather than product of conditional probabilities, since the data set is quite limited. In addition the a priori information is used, resulting, for example, in the least frequent type 3, having a better recognition rate of 52%.

In addition 2 fusers were designed and implemented:

a) a Neural Net classifier using any combination of classifiers, except the Neural Net one;
b) a DS classifier using all but the DS classifier,

The NN classifier and the DS classifier were excluded from the NN and the DS fusers respectively in order to avoid dominance of any given form of reasoning. In all cases improvements are noted, particularly when fusing classifiers of poor performances, since the complementarity of the information is then optimally utilized.

a) to 96% for the NN fuser of 3 classifiers; and
b) to 98% for the DS classifier, with the masses being related to the performance measures of the various classifiers, according to complex formulae.

### *5.2. SAR ISM Hierarchical Classifier*

The data set of SAR imagery (particularly unclassified imagery of combatant ships) is scarce compared to FLIR imagery, hence the approach for a SAR classifier has to be different. One first identifies one feature that can distinguish between the 2 major categories, namely line combatant ships and merchant ships: the presence of major superstructures is predominantly in the mid-portion of combatant ships, while those superstructures are at the bow or stern for merchant ships, with the exact knowledge based rules being classified. To compensate for the absence of real imagery, one generates a random range profile of ships and bins them in 9 bins. Then one selects 32,211 profile vectors, which according to classified based rules, correspond to the following ship distribution:

– 16,259 line combatants;
– 5,922 merchant ships;
– 10,030 others.

The 9-bin profile vectors were normalized such that the sum be one with no given cell exceeding 0.4. Since it is claimed that a NN with one hidden layer can classify any problem as long as enough neurons are taken, one chose to optimize a single layer NN through varying the number of hidden neurons, and found out that the best category

NN had 29 hidden neurons on a single hidden layer, in the sense that it offered the best compromise between approximating the knowledge based rules, and good overall performance.

Having achieved category discrimination, if the confidence level on a "line ship" declaration is high enough (lets say > 50%) from the NN, then an estimate of the line ship type should be initiated. This is performed using a Bayes classifier based on the length distributions for Frigates, Destroyers, Cruisers, and Aircraft Carriers as obtained by browsing Jane's Fighting Ships and their probability density distributions approximated by the curves shown in Figure 4 below (a similar attempt form merchant ships fails because length is not a good discriminator in this case).

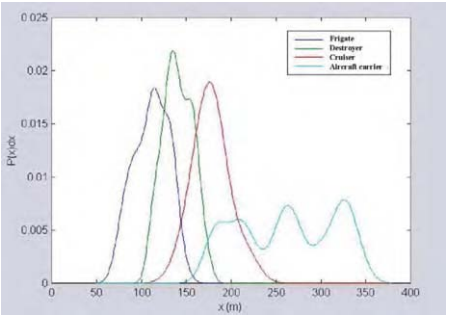



**Figure 4.** Probability density distribution for length of line combatant ships.

Given that a ship length range is evaluated from the ship end points in the imagery, one calculates the mean *a posteriori* probability $P_{avg}(t|s)$ that a ship image of extracted length interval *s,* belongs to type *t,* by averaging the standard Bayes rule over that interval. Naturally the *a priori* probability $P(t)$ is context-dependent and could be set by the radar operator prior the mission.

$$P_{avg}(t \mid s) = \underset{length\ range}{Avg}\left(\frac{p(s \mid t)P(t)}{p(s)}\right) \qquad p(s) = \sum_i p(s \mid t_i)P(t_i) \tag{3}$$

The resultant SAR ISM classifier is thus a hybrid that first performs image segmentation and ship length determination from acquisition parameters (step 1), then identifies category (step 2), then line type (step 3), and if enough imagery were to be provided, could refine the ID even further in the yet unimplemented step 4. Its architecture is displayed in Figure 5 below, and fusion through DS is performed after each step [3]. Imagery for the scenarios is provided by a simulator from DRDC-Ottawa.

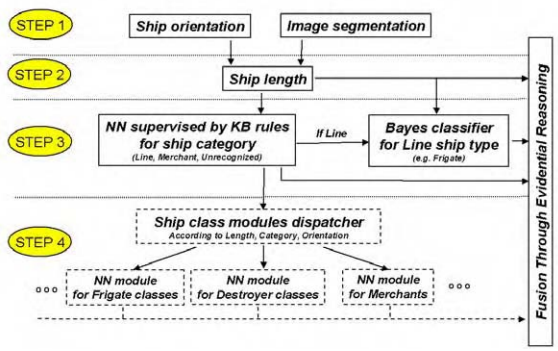



**Figure 5.** Hierarchical design of the SAR ISM classifier.

## 6. Test Scenarios

The complete set of fusion algorithms that can lead to timely ID were mainly tested for 2 scenarios, in which radar and ESM contacts were provided by the CASE-ATTI sensor module provided by DRDC-V:

1. Maritime Air Area Operations (MAAO) which involves the ID of 3 enemy Russian ships (Udaloy destroyer, Kara cruiser, and Mirka frigate) in the presence of ESM countermeasures, and which fuses the SAR ISM results since the enemy line ships are of different types;
2. Direct Fleet Support (DFM) involving American and Canadian convoys, which are also imaged by the SAR, but for which miss-association can occur, due to the geometry of the scenario.

At appropriate times in the MAAO scenario, several ESM contacts are received for each hostile vessel, one such contact being incorrect for the platform (chosen arbitrarily to be the Udaloy destroyer), in order to test the robustness of the DS evidential reasoning algorithm under countermeasures. This discrepancy will prompt the use of SSAR imaging of the Udaloy and members of its convoy. As soon as the operator has imaged the Udaloy, he/she will then in short order image the Kara cruiser and the Mirka frigate with roughly the same acquisition parameters, since the surveillance aircraft's motion over such a short period of time is not very significant. To create the imagery of enemy ships, a simulator from DRDC-Ottawa was used with permission, and in fact improved. The imagery thus generated is unclassified, and its interpretation by the SAR ISM is reproduced below in Figure 6 (removal of artifacts by thresholding, and centerline detection by a Hough transform are clearly visible). The category is always properly achieved by the NN and the type ID by the Bayes classifier in red is correct in all cases.

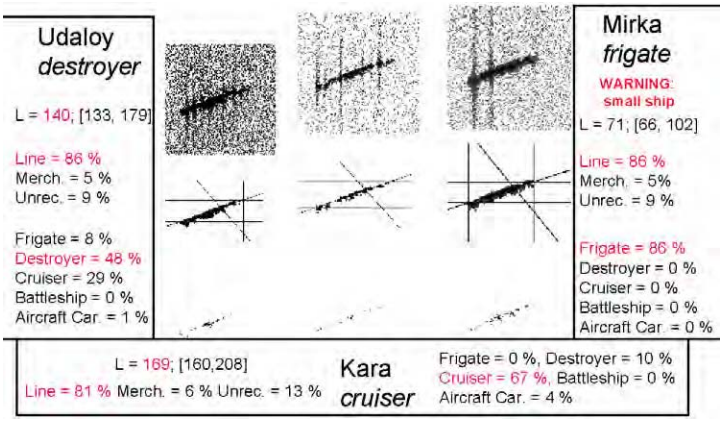



**Figure 6.** SAR imagery and ISM performance for Russian ships in MAAO scenario.

The PDB contains many Russian ships with emitters common to the 3 ships in the MAAO scenario, hence many ESM reports are necessary to achieve correct ID (these are chosen at random). The most complicated DS reasoning occurs for the Udaloy II as mentioned previously and is shown in Figure 7 below. In this Monte-Carlo case, no emitter report was detected during this time by the ESM, that could distinguish the 2 versions of the Udaloy in the PDB (triangles in blue are discriminating emitter reports and a SAR ISM fusion confirming an Udaloy triplet of same RCS signature).

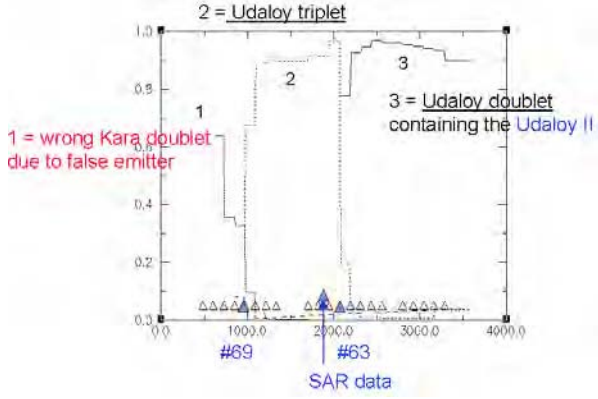



**Figure 7.** ID for the Udaloy II after countermeasures and with SAR ISM fusion.

If one concentrates on American ships in the DFS scenario, the resulting SAR imagery and ISM interpretation achieved is depicted in Figure 8 below.

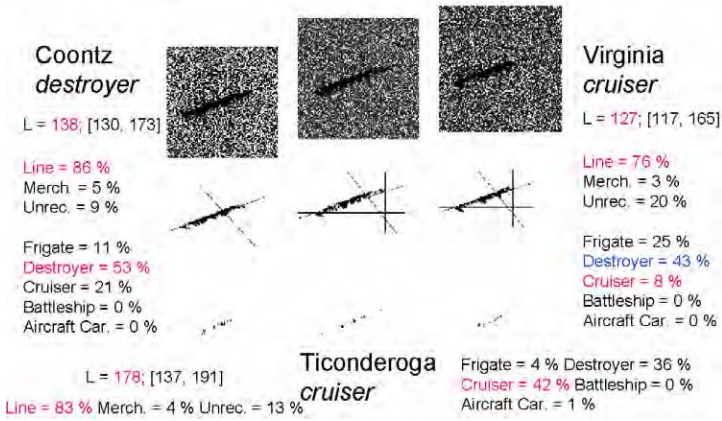



**Figure 8.** SAR imagery and ISM performance for American ships in DFS scenario.

In this case, it should be noted that the SAR ISM incorrectly identifies the Virginia cruiser as a destroyer because its length is small for a cruiser, according to Figure 4 (blue indicates a mistake by the SAR ISM, red a successful ID). If the imagery is done sufficiently early in the scenario, the ESM reports will eventually confirm the Virginia (rather than a destroyer such as the Spruance), otherwise it may not (depending on the Monte-Carlo run).

## 7. Conclusions

Through the use of a priori databases and the DS reasoning framework, all sensors and ISMs contribute declarations of (possible multiple) propositions, which can be fused to achieve a correct platform ID. The scheme is robust in the sense that it can handle conflicts, ignorance, and ambiguities, which can result from inadequate performances from sensors or ISMs, or from miss-associations in extreme tracking conditions. It can be truncated to achieve real-time performance. Its implementation has been done for various customers in Object-Oriented code, and it is presently used in the Knowledge-Based System (KBS) CORTEX at Lockheed Martin (LM) Canada.

## Acknowledgements

The author would like to thank all the LM Canada co-authors in the references below, especially Dr. Eloi Bossé of Defense Research and Development Canada (DRDC) at Valcartier for contractual work leading to much of this research, and to Dr. Alexandre Jouan, formerly of LM Canada, now at DRDC-Valcartier, who performed the scenario simulations.

## References

[1] Simard, M.-A., Valin, P., and Lesage, F. (2000) The ID box: a Multisource Attribute Data Fusion Function for Target Identification, Proc. SPIE Aerosense Conf. 4051 on Sensor Fusion: Architectures, Algorithms and Applications IV, Orlando, April 24–28, pp. 180–190.

[2] Valin, P., and Boily, D. (2000) Truncated Dempster-Shafer Optimization and Benchmarking, Proc. SPIE Aerosense Conf. 4051 on Sensor Fusion: Architectures, Algorithms and Applications IV, Orlando, April 24–28, pp. 237–246.
[3] Valin, P., Tessier, Y., and Jouan, A. (1999) Hierarchical Ship Classifier for Airborne Synthetic Aperture Radar (SAR) Images, Proc. 33rd Asilomar Conference on Signal, Systems and Computer, 24–27 October, Pacific Grove, pp. 1230–1234.
[4] Valin, P., Jouan, A., and Bossé, E. (1999) Fusion of imaging and non-imaging sensor information for airborne surveillance, Proc. SPIE Aerosense Conf. 3719 on Sensor Fusion: Architectures, Algorithms, and Application III, Orlando, 7–9 April, pp. 126–138.
[5] Jouan, A., Gagnon, L., Shahbazian, E., and Valin P. (1998) Fusion of Imagery Attributes with Non-Imaging Sensor Reports by Truncated Dempster-Shafer Evidential Reasoning, in FUSION 98, Las Vegas, 6–9 July, Vol. II, pp. 549–556.
[6] Simard, M.-A., and Jouan, A. (1997) Platform Database and Evidential Combination Method to Fuse ESM Detections with the Attribute Data From Other Sensors, NATO AGARD Symposium on EW Integration for Ships, Aircraft, and Land Vehicles, May, Ottawa, pp. 2–1 to 2–16.
[7] Bossé, E., and Simard, M.A. (1997) Identity and Attribute Information Fusion Using Evidential Reasoning, Proc. SPIE Aerosense Conf. 3067 on Sensor Fusion: Architecture, Algorithms, and Applications, Orlando, 20–25 April, pp. 38–49.
[8] Shahbazian, E., Gagnon, L., Duquet, J.-R., Macieszczak, M., and Valin, P. (1997) Fusion of Imaging and Non-Imaging Data for Surveillance Aircraft, Proc. SPIE Aerosense Conf. 3067 on Sensor Fusion: Architecture, Algorithms, and Applications, Orlando, 20–25 April, pp. 179–189.
[9] Valin, P., Couture, J., and Simard, M.A. (1996) Position and Attribute Fusion of Radar, ESM, IFF and Data Link for AAW missions of the Canadian Patrol Frigate, Proc. of Multisensor Fusion and Integration for Intelligent Systems (MFI 96), Washington, Dec. 8–11, pp. 63–71.
[10] Couture, J., Boily, E. and Simard, M.A. (1996) Sensor Data Fusion of Radar, ESM, IFF and Data Link of the Canadian Patrol Frigate and the Data Alignment Issues, in Proc. SPIE' Aerosense Conf 2759 on Signal and Data Processing of Small Targets 1996, Oliver E. Drummond, ed., 8–12 April, Orlando, pp. 361–372.
[11] Simard, M.A., Bossé, E., and Couture, J., (1996) Data Fusion of Multiple Sensors Attribute Information for Target Identity Estimation Using a Dempster-Shafer Evidential Combination Algorithm, Proc. SPIE Aerosense Conf. 2759, on Signal and Data Processing of Small Targets 1996, Oliver E. Drummond, ed., 8–12 April, Orlando, pp. 577–588.
[12] Bégin, F., Kamoun, S., and Valin, P. (1994) On the Implementation of AAW Sensors Fusion on the Canadian Patrol Frigate, Proc. SPIE Aerosense Conf. 2235, on Signal and Data Processing of Small Targets 1996, Oliver E. Drummond, ed., 4–8 April, Orlando, pp. 519–528.
[13] Simard, M.A., Valin, P., and Shahbazian E., (1993) Fusion of ESM, Radar, IFF and other Attribute Information for Target Identity Estimation and a Potential Application to the Canadian Patrol Frigate, AGARD 66th Symposium on Challenge of Future EW System Design, 18–21 October Ankara AGARD-CP-546, pp. 14.1–14.18.
[14] Bossé, E., and Simard, M.A. (1998) Managing Evidential Reasoning for Identity Information Fusion Optical Engineering, Vol. 37, no. 2, pp. 391–400.
[15] Tremblay, C. and Valin, P., (2002) Experiments on Individual Classifiers and on Fusion of a Set of Classifiers, Proceedings of the 5th International Conference on Information Fusion (FUSION 2002), Annapolis, MD, 8–11 July 2002, pp. 272–277.
[16] Valin, P. (2002) Methods for the Fusion of Multiple FLIR Classifiers, Proceedings of the Workshop on Signal Processing, Communication, Chaos and Systems: a Tribute to Rabinder N. Madan, Newpoert, RI, June 20 2002, pp. 117–122.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Distributed Fusion: Learning in Multi-Agent Systems for Time Critical Decision Making

Galina L. ROGOVA[a], Peter SCOTT[b] and Carlos LOLLETT[b]
[a] *Encompass Consulting*
[b] *State University of New York at Buffalo*

**Abstract.** A discussion of a fusion problem in multi-agent systems for time critical decision making is presented. The focus is on the problem of distributed learning for classification into several hypotheses of observations representing states of an uncertain environment. Special attention is devoted to reinforcement learning in a homogeneous non-communicating multi-agent system for time critical decision making. A system in which an agent network processes observational data and outputs beliefs to a fusion center module is considered. Belief theory serves as the analytic framework for computing these beliefs and composing them over time and over the set of agents. The agents are modeled using evidential neural networks, whose weights reflect the state of learning of the agents. Training of the network is guided by reinforcements received from the environment as decisions are made. Two different sequential decision making mechanisms are discussed: the first one is based on a "pignistic ratio test" and the second one is based on "the value of information criterion," providing for learning utilities. Results are shown for the test case of recognition of naval vessels from FLIR image data.

**Keywords.** Multi-agent systems, data fusion, time critical decision making, evidence theory, neural network, reinforcement learning

## 1. Introduction

The problem of time critical decision making arises in many applications in which it is necessary to monitor a dynamic environment to produce probable explanations of the current situation based on prior knowledge and incoming transient information, and to identify the threat. This situation, for example, can appear in target recognition problems when additional observations can improve the quality of recognition and help avoid errors, but at the same time the cost of delay may be high and there are competing demands on the sensors required for additional observations. The same considerations appear in the detection and recognition of a chemical or biological terrorist attack, which requires timely decision making and swift response but at the same time the cost of a false alarm can be very high. The information used for decision making is usually generated by multiple, often geographically distributed sources, and sequential fusion of these sources is required to produce a reliable and timely assessment of the present state of environment. Distributed multi-agent situation understanding systems are to replace the traditional centralized decision making approaches for real-world systems to achieve efficiency, robustness and scalability [1].

They can achieve faster response, higher reliability, and greater accuracy than traditional centralized decision making systems. Multi-agent systems are able to simultaneously model, explain and simulate natural phenomena, providing a practical method for designing complex distributed systems based on the concepts of agent, communication, cooperation, and coordination of actions [2].

There is no generally accepted definition of "agent" as well as definition of "multi-agent system" in the literature [3] and in most cases definitions given are tailored to the application for which the multi-agent system is designed. In general, agents are defined as "entities with goals, actions, and domain knowledge situated in an environment" [4]. Agents in multi-agent systems for decision making usually have a common goal and have to cooperate to achieve it.

A distributed multi-agent system for decision making can then be defined as a set of intelligent agents, which can observe different parts or particular characteristics of an evolving situation and then contribute to an ongoing decision making process. Due to intrinsic or extrinsic agent differences, each agent's beliefs, based on its observational database, are in general different and must be intelligently fused to achieve the desired goal. Since such systems operate in a dynamic environment they should be designed with the ability to learn in order to adapt to changes in the environment.

This paper addresses the problem of learning in multi-agent systems for time-constrained classification of observations representing states of an uncertain environment, into several hypotheses. Special attention is devoted to reinforcement learning in a homogeneous non-communicating multi-agent system for decision making. We consider a system in which a network of agents processes observational data and outputs beliefs to a fusion center module. Belief theory serves as the analytic framework for computing these beliefs and composing them over time and over the set of agents. The agents are modeled using evidential neural networks, whose weights reflect the state of learning of the agents. Training of the network is guided by reinforcements received from the environment as decisions are made. Two different sequential decision making mechanisms will be discussed and compared. The first one is based on a "pignistic ratio test" and the second one is based on "the value of information criterion." The focus is on the latter decision making process, which weighs the expected benefits of acquiring additional information vs. cost.

Section 2 presents a discussion on designing multi-agent systems for decision making and a taxonomy of learning in such systems. Specific requirements for designing a system for situation monitoring are described in section 2. The architecture and elements of a cooperative non-communicating multi-agent system for multi-hypothesis classification are presented in section 3. Section 3.4 shows simulation results for a test case involving recognition of naval vessels from FLIR data. Section 4 contains a final discussion.

## 2. Multi-Agent Systems for Decision Making

The design of a multi-agent system for decision making requires the identification of domain characteristics, the choice of agent capabilities and organization, the type of agent interaction and cooperation, the learning method, the framework for uncertainty representation, and the fusion method.

The general architecture of a multi-agent system for decision making is presented in Figure 1.

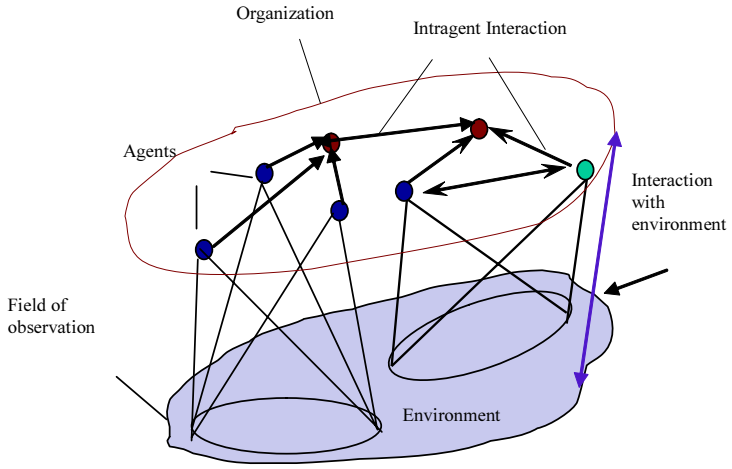



**Figure 1.** Multi-agent system.

The domain characteristics are very important since they define most of the system features. Relevant domain characteristics may include [4]: the amount of time pressure, the cost of communication and failure, the cost of processing of observations, the type of interaction with the environment, and the domain uncertainty. According to [5] there are 3 types of domain uncertainty to be considered: the existence and amount of *a priori* information about the domain, agent knowledge about the outcome of their own decisions, and agent knowledge about the outcome of other agents' decisions.

Agents in a multi-agent system can be homogeneous or heterogeneous. Homogeneous agents have the same internal structure, hypotheses, goals, and reasoning model although they might observe different characteristics of the environment and/or different parts of the environment. Agents may be heterogeneous in many ways and differ in any of the characteristics mentioned above, e.g. they might have different hypotheses or different reasoning models. Adding the possibility of heterogeneous agents in a multi-agent domain adds potential power at the price of higher complexity [4]. Although in a general multi-agent system agents can either cooperate to achieve common goals (benevolent or cooperative agents) or compete with each others (competitive agents), decision making systems comprise cooperative agents only.

Another important characteristic of multi-agent systems is the type of interaction between agents: the geometry of interaction (which agent speaks to which), what kind of information they share (observations, decisions, beliefs, mixtures), and whether, how, and which agents interact with the environment. The fusion method to be selected depends on the system characteristics, the type of information observed by the agents (numeric, symbolic, mixed) and is designed within the uncertainty framework chosen (belief theory, probability, possibility) for domain knowledge representation.

The system makes decisions about its dynamic environment and has to learn to improve its decision making ability by making its behavior "more appropriate to the environment in which the system is embedded" [6]. Multi-agent learning is "learning that is done by several agents and that becomes possible only because several agents are present" [7]. There are two possible ways to divide types of multi-agent learning. It is possible to consider a taxonomy of learning based on the type of cooperation during learning [8]:

- distributed data gathering with individual learning (all agents are involved in collecting data, which is transmitted to one particular agent which learns;
- individual learning and knowledge integration, in which agents learn independently and do not communicate during the learning process;
- distributed learning, in which the agents interact and exchange knowledge during the learning process.

While individual learning and individual learning with knowledge integration have been intensively studied in the literature, distributed learning received less attention until recently (see e.g. [9,10]). Distributed learning has definite advantages since, unlike individual learning with knowledge integration that does not change the agents' decision making ability, the developed adaptive learning process improves not only the performance of the whole system but the performance of individual agents.

It is also possible to consider a taxonomy of learning based on the type of information obtained by the agents from the environment (see, e.g. [11,12])

- unsupervised learning (clustering) which requires no interaction with the environment. Learning is based only on the internal structure of observations;
- supervised learning, in which the environment serves as a teacher that explicitly provides the desired decision (reference vector) as a feedback used to modify the decision making function of the agents;
- reinforcement learning, the strategy by which agents learn behavior through recalling the reinforcements received during trial-and-error interactions with a dynamic environment. In contrast with supervised learning, reinforcement learning does not rely on "exemplary supervision" or complete models of the environment. In learning through trial-and-error, a decision is made and the environment returns a reinforcement signal reflecting the degree of correspondence of the selected hypothesis to the long term goals.

All system design choices should be considered within the framework of the particular domain for which the system is designed. The problem of learning for decision making for situation monitoring has the following characteristics, which influence particular choices of the system architecture and agent characteristics:

- noisy and uncertain dynamic environment with insufficient *a priori* statistical information;
- geographical distribution;
- multiple Sensors with different expertise;
- a large amount of often heterogeneous information, which creates scalability problems;
- potential for the existence of malevolent agents;
- time constraints;
- high cost of error.

These characteristics call for certain choices to be made while designing a system in this domain. Having them in mind we propose to consider a homogeneous evidential cooperative non-communicating multi-agent system guided by reinforcement learning [13–16]. The proposed cooperative non-communicative architecture helps to reduce communication cost and communication delay as well as sensitivity to the failure of a key agent or to spoofing and sabotage by malevolent agents [17]. Utilization of distrib-

uted reinforcement learning allows dealing with the problem of learning in a dynamic uncertain environment and simplifies the domain knowledge elicitation problem. Neural networks used for building this system aid in dealing with the complexity of learning [18] as well as with coping with inexact, incomplete, or inconsistent knowledge and expert knowledge acquisition. Evidence theory (see, e.g. [19,20]) is proposed as a framework for modeling agents and an information fusion process to cope with the noisy and unreliable observations and the lack of *a priori* statistical information and training data. The architecture of this system and a detailed description of its elements are presented in the following section.

## 3. Evidential Cooperative Non-Communicating Multi-Agent System Guided by Reinforcement Learning

The evidential hierarchical homogeneous multi-agent system described in this paper comprises agents with a common internal structure, including domain knowledge, a common set of hypotheses, and a common procedure for assigning a level of belief to each hypothesis. They are able to extract different features from the environment but are unable to communicate directly with one another or obtain feedback from the environment. They passively acquire information from observations at discrete times $t = 1,2,...t^*$, $t^* \le T$, where $t^*$ is the time of selection of any particular hypothesis and $T$ is the deadline by which a classification decision is required. At each time $t$ each agent produces beliefs in each hypothesis under consideration and transmits these beliefs to the Fusion Center. The Fusion Center combines all the beliefs obtained from the agents up to and including time $t$ within the framework of evidence theory and produces cumulative pignistic probabilities for each hypothesis. The decision maker then maps the current set of cumulative pignistic probabilities into one of K+1 actions: "defer decision" or "decide hypothesis $k$," $k$=1,…,K.

The process of system adaptation to the environment is guided by reinforcement learning. Our goal is to teach the system to minimize the average loss associated with a incorrect decision by taking advantage of the agents' collective knowledge and the feedback from the environment. During the learning process the decision maker, upon deciding hypothesis $i$ at time $t$*, presents the decision to the environment, which returns a real reinforcement signal reflecting the utility of the decision. This signal is then fed back and distributed among the lower level agents modeled as reinforcement learning neural networks, which utilize it to incrementally improve their policy of mapping observed features into beliefs.

The proposed system architecture is presented in Figure 2.

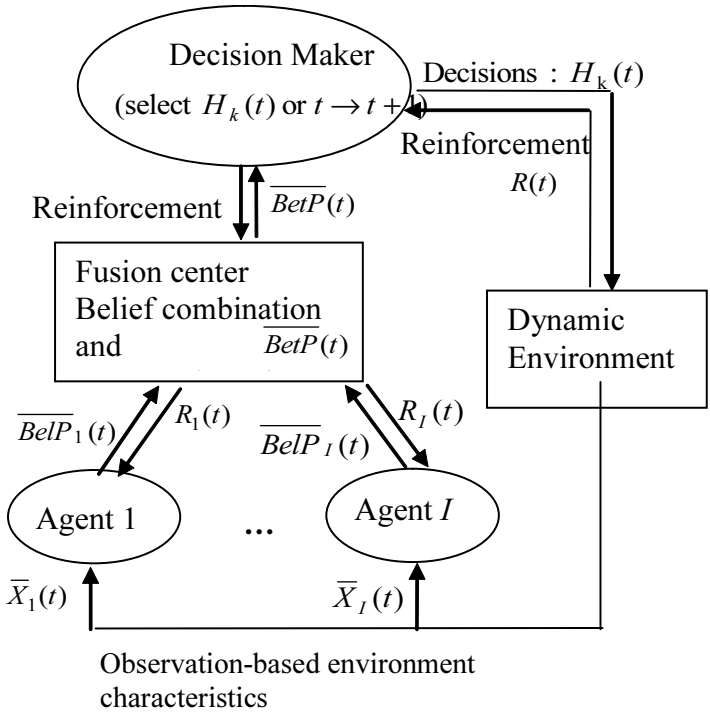



**Figure 2.** Proposed architecture.

In order to implement such an information fusion-based coordination and learning scheme, it is necessary to define a fusion model, a function relating each agent's belief functions to the reinforcement signal obtained by the fusion center, an agents' learning model and a decision making model. The description of the system elements will be presented in the following sections.

### *3.1. Agent Model*

#### *3.1.1. Agents' Architecture*

Each individual agent $i\,(1 \le i \le I)$ observes states of the environment and extracts a particular type of information represented by a feature vector $\overline{X}^i = (x_1^i, x_2^i, ..., x_{M_i}^i)$, where $M_i$ is a feature vector dimension. Let $\Theta = \{\theta_1, ..., \theta_K\}$ be a frame of discernment, where $\theta_k$ is the hypothesis that a pattern under consideration belongs to class $k$. For each agent $i$ and each class $k$, we can define a proximity measure between a feature vector $\overline{X}^i$ and a class representative vector $\overline{W}_k^i$ characterizing a strength of support for hypotheses $\theta_k$:

$$\Phi(\overline{X}^i, \overline{W}_k^i) = \Phi(d(\overline{X}^i, \overline{W}_k^i)), \tag{1}$$

where $\Phi$ is a decreasing function of distance $d(\overline{X}^i, \overline{W}_k^i)$ between $\overline{X}^i$ and $\overline{W}_k^i$ such as

$$0 \le \Phi(d(\overline{X}^i, \overline{W}_k^i)) \le 1 \text{ and } \Phi(d(\overline{X}^i, \overline{W}_k^i)) = 1 \text{ if } d(\overline{X}^i, \overline{W}_k^i) = 0. \tag{2}$$

In this paper, we consider

$$\Phi(\overline{X}^i, \overline{W}_k^i) = \alpha \exp(-\gamma \| \overline{X}^i - \overline{W}_k^i \|^2), \tag{3}$$

where $\alpha$ and $\gamma$ are constants defined experimentally and the $L_2$ norm is employed.

$\Phi(\overline{X}^i, \overline{W}_k^i)$ yields a simple support function $m_k^i$:

$$\begin{aligned} &m_k^i(\theta_k) = \Phi\left(\overline{X}^i, \overline{W}_k^i\right), \quad m_k^i(\Theta) = 1 - \Phi\left(\overline{X}^i, \overline{W}_k^i\right), \\ &m_k^i(A) = 0 \ \forall A \neq \theta_k, A \subset 2^\Theta \end{aligned} \tag{4}$$

A combination of all K simple support functions (4) with the Dempster rule [19] leads to the basic probability assignment for hypotheses $A \subset 2^\Theta$:

$$\begin{aligned} &m^i(\theta_k) = m_k^i(\theta_k) \prod_{l \neq k} m_k^i(\Theta), \quad m^i(\Theta) = 1 - \sum_k m^i(\theta_k), \\ &m^i(A) = 0 \ otherwise \end{aligned} \tag{5}$$

The process of computing basic probability assignments is modeled by an evidential reinforcement neural network trained to optimize beliefs in each hypothesis (Figure 3). The architecture of the neural network is similar to the architecture of the neural network described in [14]; however, the learning algorithm is different from the one in [14] since we consider delayed real reinforcement instead of associative binary reinforcement.

The neural network shown in Figure 2 is a feed-forward network with $M^i$ input nodes and $K$ output nodes, where $M^i$ is the dimension of observations of agent *i* and $K$ is the number of hypotheses in $\Theta$. The activation function of the first hidden layer ($L_1$) is represented by $\Phi(\overline{X}^i, \overline{W}_k^i)$. The activations of the hidden layer $L_2$ are simple support functions obtained as in (2) and (3), the output activations of the layer are the combinations of simple support functions (4) and represent basic probability assignments for each hypothesis. The connections between $L_1$ and $L_2$ and between $L_2$ and the output layer are fixed. The weight vectors $\overline{W}_k^i$ between input nodes and $L_1$ can be viewed as the center of a cluster corresponding to a certain class of input observations.

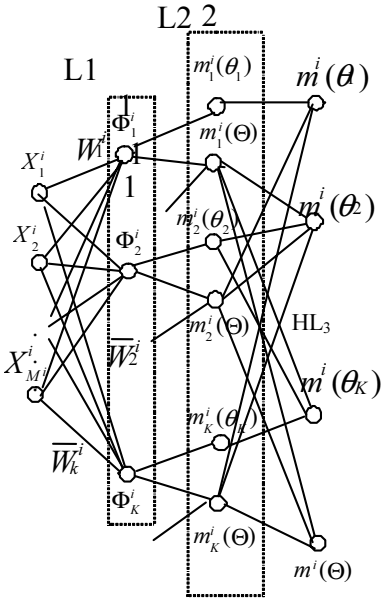



**Figure 3.** Agent architecture.

*3.1.2. Learning Algorithm*

The agents learn to optimize average reinforcement obtained from the fusion center. They are trained by a competitive reinforcement learning method similar to the methods used in [13,15]. Weight training is done in two stages, initialization and refinement. First, weights are computed independently for each agent by the unsupervised learning of K clusters followed by that one-to-one cluster-to-class mapping which maximizes that agent's expected cumulative reinforcement over the training data set. The centroids of these clusters are assigned as initial weight vectors for the corresponding classes.

In the refinement stage, initial weights are further trained in a sequential decision making, multi-agent regime in which agent beliefs are combined, a decision is made or deferred, and at the decision time, reinforcement is received from the environment and is distributed among the agents by the fusion center. Each agent then uses reinforcement obtained from the fusion center to change weights according to the rule described below.

We next give explicit form to this procedure. Let episode *j* be subsequences of observations $X^i_{T_{j-1}+\Delta t},...,X^i_{T_j}$ made by agents $i=1...I$ between two decision maker-environment interactions at time $T_{j-1}$ and $T_j$. Let $N_j$ be the number of observation times in episode $j$, i.e. $N_j=(T_j-T_{j-1})/\Delta t$ where $\Delta t$ is the time interval between two subsequent observations. Let also $t_{n_j}=T_{j-1}+n_j\Delta t$, $t_{n_j}\in[T_{j-1}+\Delta t,T_j]$, where $1\le n_j\le N_j$ ($n_j=1$ corresponds to an observation made at $T_{j-1}+\Delta t$ and $n_j=N_j$ to an observation made at time $T_j$). Let $\overline{R}^i(T_j)$ be a vector of reinforce-

ment obtained by agent *i* at the end of episode *j* (time $T_j$). Each coordinate $R_k^i(T_j)$ of vector $\overline{R}^i(T_j)$ is reinforcement for each pair "observation-hypothesis $\theta_k$" computed by the fusion center according to the rule described in the next subsection. During episode *j* the weights are not changed, while after reinforcement at time $T_j$ the weights are changed according the following rule:

$$\overline{W}_k^i(T_j) = \overline{W}_k^i(T_{j-1}) + \Delta\overline{W}_k^i(T_j), \tag{6}$$

where

$\overline{W}_k^i(T_j)$ : weight vector for agent *i*, hypothesis *k* after episode *j*

$$\Delta\overline{W}_k^i(T_j) = \rho(\overline{X}^i - \overline{W}_k^i(T_{j-1}))R_k^i(T_j), \tag{7}$$

$\rho$ : learning rate;

$\overline{X}^i$ : the average observation vector for agent i during episode j.

*3.2. Fusion Center*

The role of the fusion center is twofold. First it combines basic probability assignments of all the agents in all the hypotheses up to and including time *t* and computes pignistic probabilities of each hypothesis based on these cumulative basic probability assignments. Then it passes them upward to the decision-maker (see Fig. 2). These probabilities represent the decision-oriented "betting" (pignistic) values associated with each hypothesis based on all the observations up to and including time *t*. Second, the fusion center receives and distributes downward to the agents a real reinforcement signal obtained from the environment via the decision-maker at the end of each episode.

Combination of basic probability assignments is carried out with the Dempster rule:

$$m_c^{t_{n_j}}(A) = m^{t_{n_j}}(A) \oplus m_c^{t_{n_j-1}}(A), \tag{8}$$

where $m_c^{t_n}(A)$ is a combination of basic probability assignments produced by the fusion center in episode *j* up to and including time $t_{n_j}$, $m^{i,t_{n_j}}(A)$ and $m^{t_{n_j}}(A)$ are basic probability assignments produced by agent *i* and the fusion center at time $t_{n_j}$, respectively.

$$m^{t_{n_j}}(A) = \bigoplus_i m^{i,t_{n_j}}(A) \tag{9}$$

The pignistic probabilities computed for the basic probability assignment under the closed world assumption [20] representing a combination of all basic probability assignments produced by the fusion center in episode *j* up to and including time $t_{n_j}$ are:

$$BetP^{t_{n_j}}(\theta_k) = \sum_{\substack{\theta_k \in A \\ A \subseteq 2^\Theta}} \frac{m_c^{t_{n_j}}(A)}{|A|}. \tag{10}$$

The second task of the fusion center, which distributes reinforcement downward to individual agents, is a credit assignment problem, which is one of the major problems in reinforcement learning. Reinforcement obtained from the environment is based on the temporal combination of beliefs produced by several agents. The fusion center is faced with the problem of deciding to which degree each agent deserves credit or blame for the final decision.

In the general multi-agent sequential decision setting, the credit assignment problem is considered in the framework of a classifier system, in which each agent has a set of rules (state-action pairs) to be used for selecting a certain action. Each rule has a strength representing the importance of this rule.

Let us consider one episode, consisting of multiple observations made by multiple agents during which some set of rules fires. This episode ends when the first reinforcement signal *R* is received from the environment. There are numerous ways to allocate credit to rules [13,15–18,21–23]. The basic profit-sharing method introduced in [21] is the assignment of credit back to the agents so as to uniformly reinforce all rules which fired. The underlying notion is that if we are ignorant of the relative significance of the various rules to the level *R* of reinforcement actually received, all rules that fired should be reinforced uniformly. But more generally, it is logical that if we could identify them, only those rules that fired which contributed to the decision resulting in *R* should share the profit or blame (R itself) and their strength should be adjusted accordingly. Similarly, in basic profit-sharing it is assumed that rules that do not fire should not share in the profit, since we do no know what their effect might have been. But once again, it is logical that if we could identify the unfired rules which would have supported the decision that led to reinforcement R, these rules should be reinforced as well. For instance, if we decide class k at time t and get a positive reinforcement R, then those rules which fired favoring k should be positively reinforced, and those which did not fire but would have favored k had they fired likewise should be positively reinforced. Unlike the general profit-sharing scenario, the determination of what rules supported the decision which was made can unambiguously be made in the present case. In the other schemes [15,23] profit sharing is restricted to equal reinforcement of all rules that fired and none to those which did not fire.

In our system, a rule is an observation-hypothesis pair and a strength of this rule is a function of a corresponding weight vector of the neural network. The rule that fires is an observation-hypothesis pair consisting of an observation made by one agent and the hypothesis that agent selects as most likely based on that observation. However, in the case of delayed reinforcement, there is no information about how the decision likely made by each agent individually within an episode contributes to the decision of the fusion center at the end of the episode unless this decision is the same as the decision of the fusion center at the end of the episode.

In general, let the hypothesis selected by the system at the end of the episode be $\theta_k$. If the rule "observation-hypothesis $\theta_k$" fires it can share the same reinforcement as the system and its strength should be changed accordingly. At the same time, we do not have any specific information on how any rule "observation-hypothesis $\theta_j, \forall j \neq k$" affected the system decision and it is not very beneficial to change their strength. At the same time, any rule "observation-hypothesis $\theta_k$" can share the system reinforcement since these are the rules that, had they fired, would have strengthened the decision maker's choice.

Given this state of affaires we introduce the following profit sharing rule:

$$R_k^i(T_j) = R(T_j)e_k^i(T_j) \tag{11}$$

where

$R_k^i(T_j)$ is reinforcement sent by the fusion center to agent *i* to update the strength of the rule "observation-hypothesis $\theta_k$,"

$e_k^i(T_j)$: agent *i*'s eligibility factor for hypothesis $\theta_k$ at the end of episode *j*.

$e_k^i$ =1 if $\theta_k$ is the hypothesis selected by the system at the end of episode *j*, $e_k^i$ =0, otherwise.

### *3.3. The Decision-Maker*

#### *3.3.1. Sequential Decision Making*

The task of the sequential decision maker is to employ information obtained from the fusion center to choose whether to decide now (and if so which hypothesis), or defer the decision and request another observation. If the action is *decide now* and a certain hypothesis is selected, the decision maker presents the decision to the environment, which evaluates it, notes the decision time *t,* and returns a reinforcement signal used for adjusting agent weights and the decision function of the system. Delayed reinforcement learning is used, since in most cases a decision will be made, and reinforcement received, only after several observations. If the action *defer decision* is chosen, agents will provide additional information based on new observations to the fusion center, which will combine this information with information obtained at the previous step and transmit this updated information to the decision-maker.

The decision is deferred if we expect to improve the outcome of the decision making process by acquiring one or more additional observations while suffering from increased decision latency. There may be several criteria to consider when we decide whether we can improve our decision by deferring it. For example, we will tend to wait if we expect that the probability of selecting the right decision or the confidence of our decision will increase sufficiently [13,24,25]. This leads to the "pignistic probability ratio test." However, utilization of this test does not allow us to explicitly include cost of time and resources into our decision making and this approach can be used if we just need to guarantee that a hypothesis will be selected before a certain deadline.

In order to explicitly include benefits of deferring a decision, a different policy based on the Maximum Expected Utility Principle [23] has been designed [15]. The decision "wait for a new observation" or "decide on hypothesis $\theta_k$" is based on the value of information criterion [9]. According to this criterion, a new observation is needed if the difference between the maximum expected utility (maximum taken over all possible decisions) with the new observation and without the new observation is greater than the cost of obtaining this new observation. The expected utility is assumed to be a decreasing function of time since it includes the opportunity costs and increased decision latency. A detailed description of both policies is presented below.

*3.3.2. Pignistic Probability Ratio Test*

Fu's Modified Sequential Probability Ratio Test [22] employs time-varying decision thresholds, which can be set to achieve quasi-optimal performance while guaranteeing that the decision will be made by a fixed deadline. The model introduced in [17] and described below uses this approach with two modifications. First, multiple hypotheses rather than binary hypotheses are considered. Second, instead of the likelihood ratio test, the pignistic probabilities over the frame of discernment, which are available to the decision maker after a subsequence of observations, are utilized. Let $T_d$ be the deadline by which a decision must be made, $n_d$ be the maximum number of observations permitted before the decision maker reaches the deadline.

We will decide hypothesis $\theta_k$ for episode *j* at time $T_{j-1} < t_{n_j} \le T_d$ ($n_j \le n_d$) if there is a hypothesis $\theta_k$, which satisfies the inequality:

$$BetP^{t_{n_j}}(\theta_k) \ge \exp(\alpha_{n_j} t_{n_j}) BetP^{t_{nj}}(\neg\theta_k) \tag{12}$$

and if this inequality is not satisfied for any $k$ at any earlier time in the jth episode. In the case that decision rule (12) is satisfied by two or more hypotheses at the same earliest time $t_{n_j}$, the hypothesis with the largest pignistic probability $BetP^{t_{n_j}}(\theta_k)$ is selected. $\alpha_l$, $l = 1 \ldots n_d$, is a set of time-varying scalars, called the threshold sequence, varying between initial value $\alpha_1$ and terminal value $\alpha_{n_d}$. The threshold sequence sets the trade-off between error rate and decision latency. $\alpha_1$ specifies the level of confidence needed to make an immediate decision after a single observation. Anchoring the threshold sequence with the value $\alpha_{n_d} = -\ln(K)/t_{n_d}$ (*K* hypotheses in the frame of discernment) guarantees that a decision is made by the deadline $T_d$. The threshold sequence is often set to vary linearly, but if early decisions are to be encouraged the sequence can be made convex, and if ultimate accuracy is more important than latency, concave.

### *3.3.3. Utility-Based Model of the Sequential Decision Maker*

This subsection is focused on a model of the decision-maker which, rather than comparing the current best hypothesis probability to a threshold, compares the expected utility of additional observations with the disutility of waiting. The disutility of waiting includes cost represented by the time required for additional testing and processing of the test results, the cost of observational, computational and communications resources used to acquire additional observations, and the opportunity cost associated with delayed decision and action. The system to be described here maximizes the expected utility of delayed decisions while minimizing decision latency (time to decision).

One of the major problems with using expected utility is that obtaining utility values from domain experts can be slow and expensive, and produce subjective, possibly inconsistent, utility values. Here we simplify the utilities-from-experts problem by employing reinforcement learning to learn utilities directly from environmental feedback signals.

Reinforcement learning has been successfully used for learning utilities in medical diagnosis systems [27] and we employ this approach here, with several notable modifications. First, we consider a decision deadline rather than assuming the process can continue until all the tests have been exhausted. Second, we deal with multiple hypotheses rather than with binary decisions ("does a patient have this particular disease?"). Third, we assume that utilities decrease with each additional observation (increased opportunity cost), and, fourth, the decisions take into account not only a decision maker's belief based on current observations but the decision maker's cumulative-over-time belief up to and including the current observation. Fifth, the process of learning utilities by the fusion center is coupled with the process of learning the beliefs of the agents employed by the system.

As it was mentioned earlier, the core difficulty of utilizing the Maximum Expected Utility Principle is a problem of finding the utility for each decision. There are several reinforcement methods developed for learning by delayed reinforcement [12]. One of them, the Temporal Difference method [12,28], has been proven to be successful for learning utilities [27]. Here we employ learning by Temporal Difference to converge towards an optimal policy to be used by the decision maker. The process of learning utilities is thereby coupled with the process of learning the beliefs of the agents employed by the system. In essence we learn how valuable each hypothesis is to us as we learn the correct beliefs that each hypothesis is true given the observations the agents have made.

Temporal difference methods (TD($\lambda$) [28]) learn by employing the difference between temporally successive predictions. They have two major advantages over other prediction methods, they are more incremental and easier to compute, and they learn faster. TD($\lambda$) uses a sequence of input data $X_n$, $n = 1,...,N$ to produce a sequence of estimates $P_n$, which are functions of corresponding inputs $X_n$ and weights $W_n$. These estimates are predictions of the final reinforcement $Z$, which is defined as a data point $N$+1.

TD($\lambda$) updates weights according to the following equation:

$$\Delta W_n = \alpha(P_{n+1} - P_n)\sum_{m=1}^{n} \lambda^{n-m} \nabla_w P_m . \tag{13}$$

Here $\alpha$ is a parameter in the range [0, 1] defining the learning rate and $\nabla_W P_m$ is a vector of partial derivatives of $P_m$ with respect to weights. In the online version of this algorithm:

$$\begin{aligned} W_{n+1} &= W_n + \alpha(P_{n+1} - P_n)e'_n, \\ e_{n+1} &= \nabla_W P_{n+1} + \lambda e_n \end{aligned} \tag{14}$$

where $e_n$ is the eligibility factor [12,28]. Let $U(t) = (u_{kl}(t))$ be a time-dependent utility matrix where $u_{kl}(t)$ is the utility of selecting, at time t, hypothesis $\theta_k$ when the true state of nature is $\theta_l$.

Given observations at time $t_{n_j} \in [T_{j-1} + \Delta t \quad T_j]$ of episode *j*, the agents determine their beliefs and the fusion center combines them and computes $\overline{BetP}^{t_{n_j}}$. We define the vector of expected utilities at time $t_{n_j}$ as $\overline{E}(t_{n_j}) = U(t_{n_j}) \cdot \overline{BetP}^{t_{n_j}}$. With each deferred decision we also consider a cost $C \in [0\ 1]$, which does not depend on time and represents the disutility of increased decision latency together with the cost associated with resource utilization. If we consider the elements of the utility matrix as weights to be learned and $\overline{BetP}^{t_{n_j}}$ as inputs at time $t_{n_j}$, we can learn utility by changing the weights in a way similar to that presented in equation (13).

At the end of each episode *j* (at time $T_j < T_d$) when the decision maker decides to stop observations and select a hypothesis $\theta_k$, reinforcement $R$ becomes known and the *k*th row of the utility matrix, corresponding to the selection of hypothesis $\theta_k$, is updated as follows:

$$U^k_{T_j} = U^k_{T_{j-1}} + \sum_{j=1}^{J} \Delta U^k_{t_{n_j}}, \tag{15}$$

where,

$$\begin{aligned} \Delta U_{t_{n_j}} &= \alpha\ U(T_{j-1})(\overline{BetP}^{t_{n_j+1}} - \overline{BetP}^{t_{n_j}})\overline{e}'_{t_{n_j}}, \\ \overline{e}_{t\,n_j} &= \overline{BetP^{t_{n_j}}} + \lambda \overline{e}_{t\,n_j-1} \end{aligned} \tag{16}$$

$$\Delta U^k_{T_j} = \alpha\ (R - U^k_{T_{j-1}} \overline{BetP}^{T_j}) e_{T_j} \tag{17}$$

and $J = (T_j - T_{j-1}) / \Delta t$

According to the value of information criterion, decision time $T_j$ is computed as the first time such that the difference between the expected value of the maximum expected utility (MEU) at the next observation time $T_j + \Delta t$ and value of the MEU at the current time $T_j$ is not larger than the cost associated with the additional observations, that is, decision time $T_j$ corresponds to the first time satisfying

$$Exp\left\|\{\overline{E}(T_j + \Delta t)\}\right\|_\infty - \left\|\overline{E}(T_j)\right\|_\infty \le C \tag{18}$$

where $\overline{E}(T_j + \Delta t)$ is the expected utility vector at the next time, $\overline{E}(T_j)$ is the current expected utility vector, *Exp* is the expectation operator and $\|\cdot\|_\infty$ is the max norm.

To apply (18) we need to compute the expected value of the MEU after one additional observation. To determine this without approximation we need to know all possible next observations and their relative likelihoods so that the pignistic probabilities can be updated. Since this information is not available in our case, we approximate the expected MEU by utilizing a training set and agents' weights in the episode under consideration.

Let $\overline{X}_k$ be an average of the training vectors belonging to a set containing vectors that get positive reinforcement for hypothesis $\theta_k$ before reinforcement learning is implemented. Let $\overline{BetP}_k^{t_{n_j}}$ be the pignistic probability produced by the system processing observation $\overline{X}_k$ at time $t_{n_j}$. Then the predicted maximum expected utility could be obtained as:

$$Exp\left\|\{\overline{E}(T_j + \Delta t)\}\right\|_\infty = \{\max_k (U(T_j)\overline{BetP}_k^{T_j})\} * \overline{BetP}^{T_j} \tag{19}$$

There are two ways to define the learning process. In the first case (corresponding to parametric modeling of the time-dependent utility matrix), we assume that while utilities decrease with time, the relative utilities of distinct matrix elements remains constant. We thus represent time-dependent utility $u_{kl}(t)$ as $u_{kl} \cdot f(t)$ and $U(t) = (u_{kl} \cdot f(t)) = U \cdot f(t)$, where $f(t)$ is a decreasing function of time with $f(1) = 1$. The derivative of this function defines the temporal pattern of utility change. The function is selected such that it guarantees that the observations will be stopped and a decision made before the deadline *t=T*. We refer to this as the Parametric Time-Dependent (PTD) case. The second case is where $U(t)$ is time dependent without parametric constraints, the General Time Dependent (GTD) case.

The PTD case is equivalent to eliciting expert utilities when experts are required to provide "absolute" time-independent utilities. In this case reinforcement is represented

by a "time independent" matrix and computed as $R/f(t) = (r_{kl})/f(t)$. In this case equations (14)–(16) can be rewritten as follows:

$$\begin{aligned} \Delta U_{t_{n_j}} &= \alpha U_{T_{j-1}} \cdot (\overline{BetP}^{t_{n_j+1}} - \overline{BetP}^{t_{n_j}})\bar{e}'_{t_{n_j}}, \\ \bar{e}_t &= \overline{BetP}^{t_{n_j}} + \lambda \bar{e}_{t_{n_j-1}} \end{aligned} \tag{20}$$

$$\Delta U^k_{T_j} = \alpha(R/f(t) - E_k(T_j))e_{T_j}, \tag{21}$$

and $T_j$ is computed as the first time such that:

$$Exp\left\|\{\bar{E}(T_j + \Delta t)\}\right\|_\infty f(T_j + \Delta t) - \left\|\bar{E}(T_j)\right\|_\infty \cdot f(T_j) \le C \tag{22}$$

and predicted expected utility $Exp\left\|\{\bar{E}(T_j + \Delta t)\}\right\|_\infty$ is computed as in (19).

The second GTD case (repetition begins) is equivalent to an expert knowledge elicitation process when it is assumed that expert knows and can explicitly determine how the utilities change with time as the deadline approaches. In this case equations (17)–(19) are used without changes (repetition ends).

In this case there are no constraints linking the values of $U(t)$ over time. For each t the utility matrix U(t) must be learned independently. This case (repetition begins) is equivalent to an expert knowledge elicitation process when it is assumed that expert knows and can explicitly determine how the utilities change with time as the deadline approaches. In this case equations (17)–(19) are used without changes (repetition ends).

### *3.4. Experiments and Results*

The multi-agent system for decision making described in the paper is problem independent and does not impose any restrictions on the kind of features or information used by each agent. In order to discuss the implementation issues and evaluate the performance of this system we conducted a series of experiments with both decision making methods discussed in the previous sections: the pignistic probability ratio test and the method based on the value of information criterion. The experiments were conducted with 2545 Forward Looking Infrared (FLIR) ship images from the US Naval Air Warfare Center, China Lake, California, originated by Dr. Jack Sklansky of the UC at Irvine and obtained from Dr. Pierre Valin of Lockheed Martin Canada.

Each image belongs to one of the eight classes listed in Table 2. These classes were further aggregated into two groups: *friend* and *foe*. All container ships were in group friend, all fighting ships in group *foe*, and so on. Figures 4 and 5 present typical silhouettes for the 8 classes listed in Table 1.

**Table 1.** Ship classes

| Class Name | Class Number | Group Name | Number of Images |
|---|---|---|---|
| Destroyer | 1 | Foe | 340 |
| Container | 2 | Friend | 455 |
| Civilian Freighter | 3 | Friend | 186 |
| Auxiliary Oil Replenishment | 4 | Friend | 490 |
| Landing Assault Tanker | 5 | Foe | 348 |
| Frigate | 6 | Foe | 279 |
| Cruiser | 7 | Foe | 239 |
| Destroyer with Guided Missile | 8 | Foe | 208 |

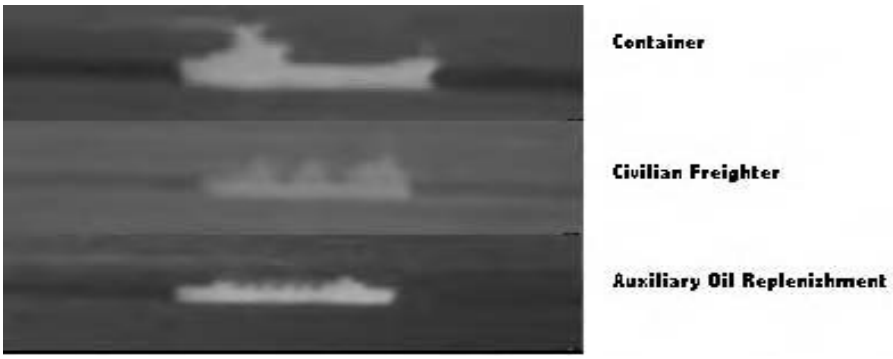



**Figure 4.** Images of the 3 classes of ships (Friends).

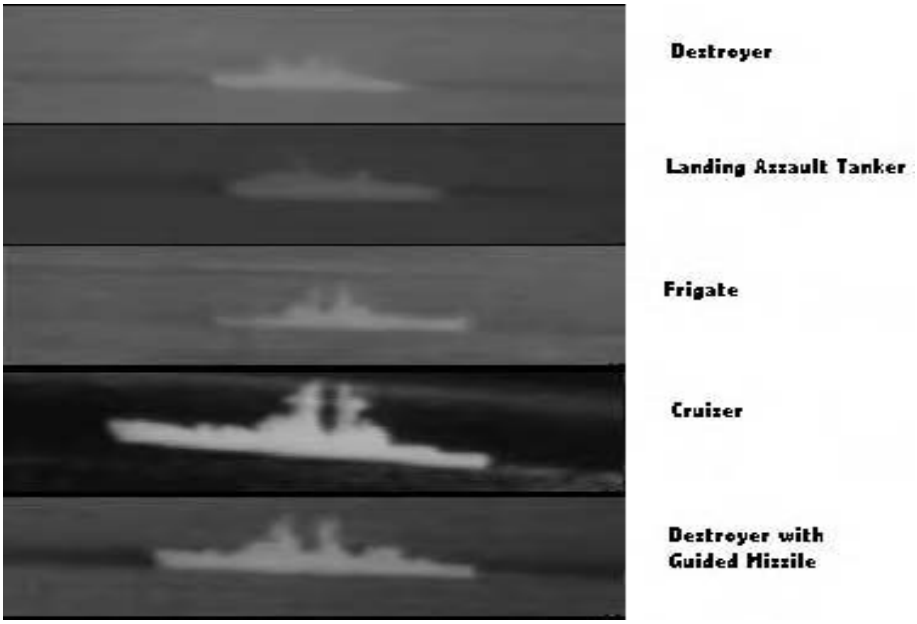



**Figure 5.** Images of 5 classes of group *foe* ships.

The features used include seven moments given in [29]. These moments are invariant under translation, rotation and scale. But these moments deliver information primarily on the global shape of the object and represent poorly the details of the object. In [30] four features were added by fitting an auto regressive model to a one-dimensional sequence of the projected image along the horizontal axis. The quality of each image varies considerably depending on the distance of the ship to the camera and the noise in the image.

In our experiments we divided all the features into two groups. The first group included the invariant moments, the second group contained the remaining features. We conducted experiments with two agents, each of them using the specific group of features described above as input. Since the features of these two groups represent different properties of the image and have low class conditional correlation, we can expect that agents based on these features are "error-independent" (see e.g. [31,32]) and can be successfully combined.

Testing was performed using cross-validation in which the group reserved for testing was fixed at 20% of the total available dataset. The training and test data were grouped into episodes of ten patterns drawn from the same object class.

In experiments with the utility based decision making, the object class was first selected with uniform probabilities over the eight classes, and samples were then randomly chosen from the portion of the test image set associated with that object class. All the results shown below were obtained after 60 epochs and were averaged over 3 runs. The initial set of weights for the agents was obtained after one epoch of unsupervised learning of K clusters of each agent separately followed by that one-to-one cluster-to-class mapping which maximizes that agent's expected cumulative reinforcement over the training data set.

The reinforcement matrix considered in the experiments represents a specific attitude of the decision maker towards false alarm, correct binary recognition (*friend/foe*), and correct recognition of each ship class. In our experiments, the environment gives the highest positive reinforcement (reward) in the case of correctly recognized classes, a smaller positive reward is given to a correctly recognized group, while punishment for group error *foe/friend* is more severe than punishment for error *friend/foe*. In particular, the following reinforcement matrix $R = (r_{lk}) \cdot f(t)$ was used:

$r_{lk} = 1$, if selected class $i$ is the same as observed class $k$ ($i$=$k$),

$r_{lk} = 0.7$, if both selected class i and observed class $k$ belong to the same group (friend or foe),

$r_{lk} = -0.5$, if selected class belongs to the group *friend* and observed classes belong to the group *foe*.

$r_{lk} = -1$, if selected class belongs to the group *foe* and observed classes belong to the group *friend*.

The following initial values for utility were considered:

$u_{lk} = 1$, if $i$=$k$ and $u_{lk} = -1$, otherwise, i.e., the highest positive utility corresponds to correct class recognition and highest negative utility corresponds to wrong class recognition.

Experiments were conducted for 4 different cost values: c=0, 0.05, 0.1, 0.2 and the parametric time dependent (PTD) case: $U(t) = f(t)U$, where *f(t)* was represented as a function of the number of steps before and up to the time of decision making and is computed as follows. Let for each episode $n_t = [t/\Delta t]$, then

$$f(n_t) = \frac{\beta}{1+\beta}$$
$$\beta = (\delta + \ln(K))\left(1 - \frac{n_t}{N}\right) - \ln(K), \tag{23}$$

where $K$ is the number of classes, $N$ is the maximum number of observations in an episode, $\delta$ is a positive parameter, $\Delta t = T/N$, and $T$ is the length of an episode. In our experiments we used $\delta = 9$ and $N$=10.

In each experiment we evaluated the performance of the system. We emphasized the ability of the system to learn to maximize utility while preserving interest in the correct group designation, the important binary distinction *friend-foe*. We were also interested in the more refined identification of ship class and the ability of the system to decrease the average number of observations in an episode (the average number of observations used by the system to arrive at a decision) as the result of learning.

The performance of the (repetition follows) system, in which reinforcement is used for learning agents' decision function as well as utility matrix with one, in which reinforcement is used for modifying agents decision function only (repetition ends). The results of experiments for PTD utility learning are presented in Figures 6–11. The system training process is presented in Figures 6–9, in which utility, the class and group recognition rates, and the average number of steps are shown as functions of the number of epochs in the training process. The figures show that the learning process stabilizes after only 20 iterations.

Graphs in Figures 10, 11 compare the performance of the (repetition follows) system in which reinforcement is used for learning agents' decision function as well as utility matrix with one, in which reinforcement is used only for modifying agents' decision function only (repetition ends). Figure 10 presents utility and the number of steps as functions of cost. Figure 11 shows class and group recognition rate as functions of cost. The graphs in the figures demonstrate the superior performance of the system, which includes utility learning. It is shown that the average utility of each decision increases as the result of learning while the latency decreases. Although the accuracy of group recognition rate increases significantly as the result of utility training, the class recognition accuracy does not improve as much. This fact can be explained by the nature of the reinforcement function, which emphasized group recognition and did not provide significant punishment for incorrect class recognition, while the initial utility matrix was specifically designed to emphasize correct class recognition only.

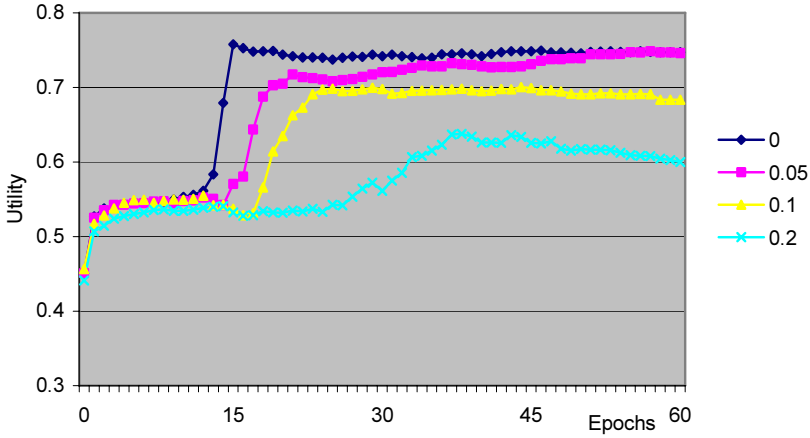



**Figure 6.** Utility as a function of the number of epochs for various costs.

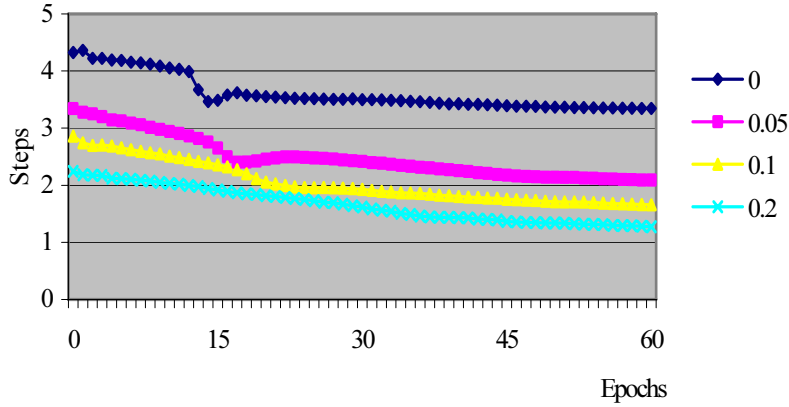



**Figure 7.** The number of the steps as a function of the number of epochs.

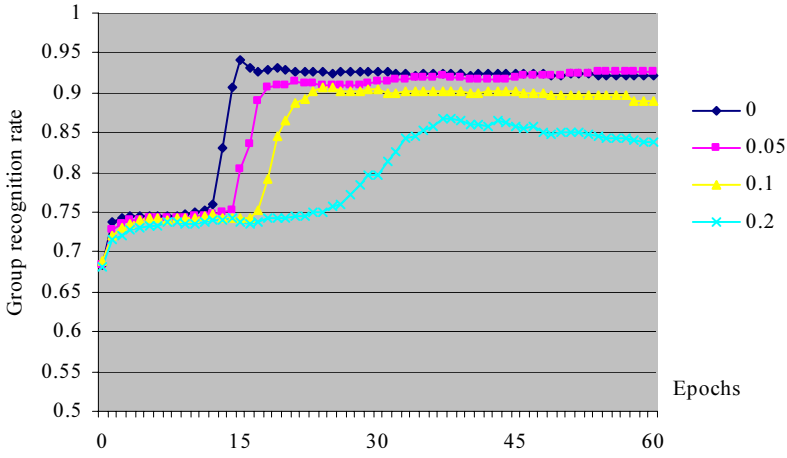



**Figure 8.** Group recognition rate as a function of the number of epochs for various costs.

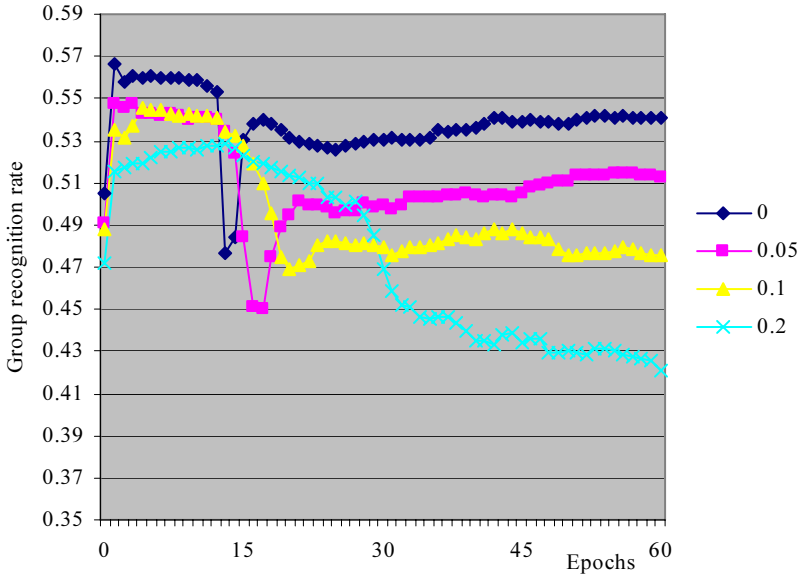



**Figure 9.** Class recognition rate as a function of the number of epochs.

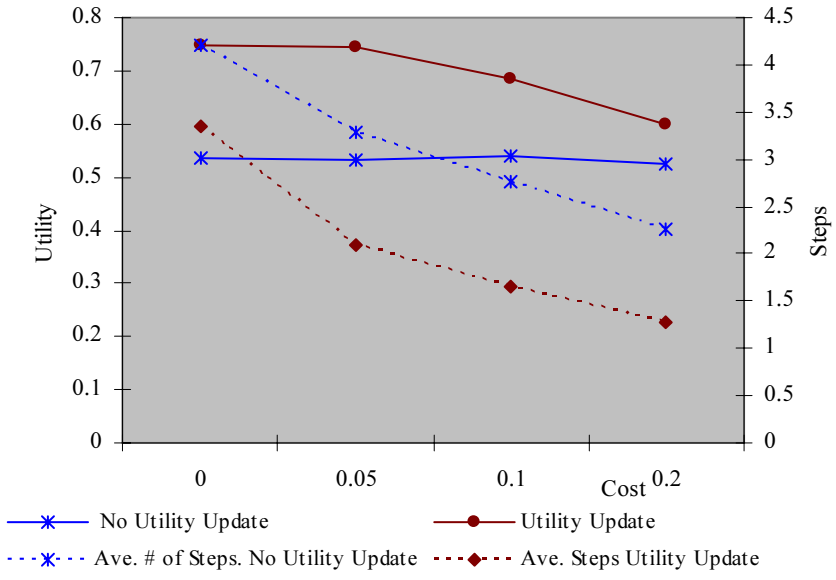



**Figure 10.** Utility and the average number of steps as functions of cost.

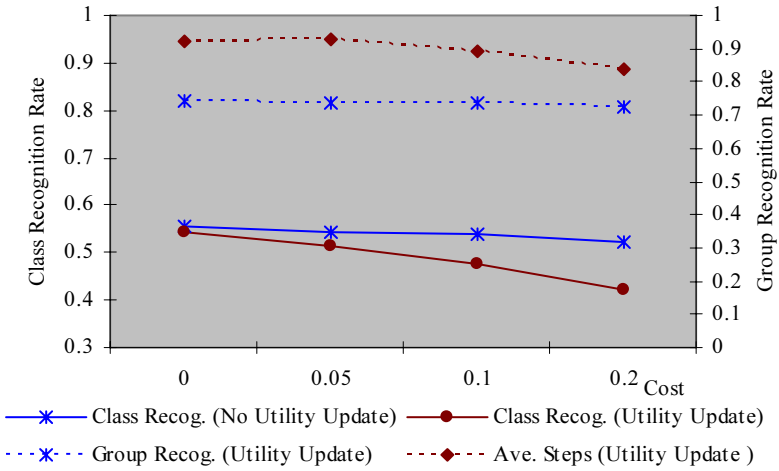



**Figure 11.** Group and class recognition rate as functions of cost.

Another set of experiments with the ship data was conducted with the pignistic probability ratio test (the confidence based decision making). The threshold function was computed as in equation (23). The reinforcement matrix used was the same as the reinforcement matrix used in the experiments conducted with the system, in which decision making was guided by the value of utility. The pignistic probability ratio test is based on the relative confidence of the decisions only and does not explicitly refer to the cost of observations. For this reason we compare the results of the pignistic ratio test with the results obtained by the utility based decision making, when C=0. The comparison results are shown in Table 2.

**Table 2.** Systems performance (C=0)

| | | utility | group recognition rate | class recognition rate | avg. # of steps |
|---|---|---|---|---|---|
| pignistic probability ratio test | | 0.537 | 0.743 | 0.587 | 4.87 |
| utility based | no utility update | 0.534 | 0.742 | 0.557 | 4.21 |
| decision making | utility update (ptd) | 0.747 | 0.921 | 0.541 | 3.35 |

It can be seen from Table 2 that the results achieved by the pignistic ratio test are similar to those achieved by the utility base decision making without utility update, while the number of steps required to obtain these results are slightly higher for the pignistic ratio test. It can be also seen that the incorporation of the utility learning process can significantly improve the performance of the system.

## 4. Conclusions

This paper presents a discussion of issues of distributed fusion for situation monitoring in dynamic environments. The focus is on the problem of distributed learning for the classification of sequential observations representing states of an uncertain environment into several hypotheses. Special attention is devoted to reinforcement learning in

a homogeneous non-communicating multi-agent system for time critical decision making and issues of designing such a system are discussed. In particular, a system in which an agent network processes observational data and outputs beliefs to a fusion center module is explored. Belief theory serves as the analytic framework for computing these beliefs and composing them over time and over the set of agents. The agents are modeled using evidential neural networks, whose weights reflect the state of learning of the agents. Training of the network is guided by reinforcements received from the environment as decisions are made. Two different sequential decision making mechanisms are discussed and compared. The first one is based on a "pignistic ratio test" and uses a time varying threshold to compare the pignistic probability of each hypothesis with its complement. The time varying threshold is used to assure that a decision is made before a deadline. The second one is based on "the value of information criterion," which weighs the benefits of acquiring additional information vs. cost. An approach to building a utility-based model of cost-sensitive decision making in time-constrained situations has also been introduced. Reinforcement learning of a time varying utility matrix coupled with a second distinct reinforcement learning process for training the beliefs of the agents is presented.

The reported case study shows the feasibility and benefits of employing a temporal difference model in the context of evidence theory for sequential decision making. The case study demonstrates that decision utilities, as well as the system classification ability, can be improved through reinforcement learning of the utility matrix. It shows that both parametrically constrained time-dependent utility matrices can be learned from feedback on the quality of decisions rather than directly elicited from an expert, which simplifies the slow, expensive and potentially error-prone expert knowledge elicitation process.

The presented methods are not problem specific and can successfully be used for both military and non-military applications. They can be used, for example, in multi-sensor moving target recognition, in single-sensor classifiers in which the feature set is partitioned across several distinct computing agents and parallel-processed to reduce decision latency, in the situation assessment problem requiring selection of a certain hypothesis about the state of the environment, for building reinforcement learning-based sensor management algorithms, or to improve medical decision making in life-threatening situations.

Additional experiments are necessary to determine the relative performance of these two homogeneous non-communicating multi-agent systems with statistical reliability, and to investigate the problem of incorporating agents' reliability into the sequential decision making process. More research is also needed in order to address fundamental issues of the problem of distributed learning in problem solving systems such as utilization of *a priori* knowledge, incorporation of symbolic and numeric information, convergence of the process and related issues.

## Acknowledgements

The research described here was supported by the U.S. Air Force Research Laboratory (AFOSR and Sensor Directorate) under contracts No. F49620-01-1-0371 and F30602-96-C-0336 and by the National Imagery and Mapping Agency under contract No. NMA 202-97-D-1033 DO#33.

## References


[1] T. Lee, S. Ghosh, A. Nerode, Asynchronous, Distributed Decision making Systems With Semi-Autonomous Entities: A Mathematical Framework, IEEE Trans. On SMC-B, V. 30, No. 1, 2000, 229–239.

[2] J. Ferber, Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence Addison-Wesley Pub, 1999.

[3] S. Russel, P. Norvig, Artificial Intelligence: A Modern Approach. Prentice Hall, 1995.

[4] P. Stone, M. Veloso, Multiagent Systems: A Survey from a Machine Learning Perspective, Autonomous Robotics, V. 8, 3, 2000.

[5] K. Deker, Environment centered Analysis and Design of Coordination Mechanisms, PhD Thesis, University of Massachusetts, 1995.

[6] L.P. Kaelbling, *Learning in Embedded Systems*, MIT Press; 1993.

[7] G. Wei, Distributed reinforcement learning, Robotics and Autonomous Systems, 15, 135–143, 1995.

[8] P. Brazdil, et al. Learning in Distributed Systems and Multi-Agent Environments, in Proc. of the European Working Session on Learning, Springer-Verlag, 412–423.

[9] M. Tan, Multi-agent Reinforcement Learning: Independent versus Cooperative Agents, In Proc. of the 10th International Conference on Machine Learning, pp. 330–337, 1993.

[10] Towell, G., *et al.* Refinement of Approximate Domain Theories by Knowledge-based neural networks, In Proc. of the 8th National Conference on Artifitial Intelligence, pp. 723–728, 1990.

[11] D. Ballard, An Introduction to Natural Computation, MIT Press, 1997.

[12] R.S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, MIT Press 1998.

[13] G. Rogova, P. Scott, C. Lollett, Distributed Reinforcement Learning For Sequential Decision Making, in: Proc. of the FUSION'2002-Fifth Conference on Multisource-Multisensor Information Fusion, 2002.

[14] G. Rogova, J. Kasturi, Reinforcement Learning Neural Network For Distributed Decision Making, in: Proc. of the FUSION'2001-Fourth Conference on Multisource-Multisensor Information Fusion, 2001, Montreal, Canada.

[15] G. Rogova, C. Lollett, P. Scott, Utility-Based Sequential Decision making In Evidential Cooperative Multi-Agent Systems, In: Proc. of the FUSION'2003-Sixth Conference on Multisource-Multisensor Information Fusion, Cairns, Australia, 2003.

[16] G. Rogova, R. Menon, Decision Fusion For Learning In Pattern Recognition, In: Proc. of the FUSION'98, First Conference on Multisource-Multisensor Information Fusion, 1998, 191–198.

[17] Sian, S.S. Adaptation based on cooperative learning in multi-agent systems, Decentralized AI-2, Y. Demazeu & J.-P. Muller (Eds.), Elsevier Science, 1991, 257–271.

[18] G. Wei and S. Sen, editors, Adaptation and learning in Multiagent Systems.

[19] Shafer, G., A Mathematical Theory of Evidence, Princeton, MIT Press, 1976.

[20] P. Smets and R. Kennes, The transferable belief model, Artificial Intelligence, vol. 66, 1994, 191–243.

[21] J.J. Grefenstette, Credit assignment in rule discovery systems based on genetic algorithms, Machine Language, v. 3 n. 2–3, p. 225–245, Oct. 1988.

[22] S. Aria, K. Sykara, Multi-agent reinforcement learning for planning and conflict resolution in a dynamic domain, Proceedings of the Fourth International Conference on Autonomous Agents, 2000.

[23] Sen, S., Sekaran, M., Multi-agent coordination with learning classifier systems, in the Proc. of IJCAI'95 Workshop, 1995, 219–233.

[24] Baum, C.W., Veeravilli V.V., A sequential procedure for multihypothesis testing, IEEE Transactions Information Theory Vol. IT-40, 1994–2007 November 1994.

[25] Fu, K.S., Sequential methods in pattern recognition and machine learning, Academic Press, New York, 1968.

[26] J. von Neuman, O. Morgenstern, Theory of Games and Economic Behavior. Princeton University Press, Princeton, NJ, 1947.

[27] M. Stesmo, T. Sejnowski, Using Temporal-Difference Reinforcement Learning to Improve Decision-Theoretic Utilities for Diagnosis, In: Proc. 2nd Joint Symposium on Neural Computation, University of California, San Diego and California Institute of Technology, June 1995.

[28] Sutton, R.S. (1988). Learning to predict by the method of temporal differences. Machine Learning, 3, 9–44.

[29] Park, Youngtae and Jack Sklansky, Automated Design of Linear Tree Classifiers. Pattern Recognition, Vol. 23, No 12, pp. 1393–1412, 199.

[30] M.K. Hu, Visual pattern recognition by moment invariants, IRE Transactions on Information Theory, No. 8, 179–187 (1962).

[31] G. Rogova, Combining the results of several neural network classifiers. *Neural Networks* 7, 5, pp. 777–781, 1994.
[32] P. Langley, "Induction of Recursive Bayesian Classifiers," Proc Eighth European Conf. on Machine Learning, Vienna, Austria, 1993.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Active Robotic Sensing as Decision Making with Statistical Methods

Lyudmila MIHAYLOVA[a], Tine LEFEBVRE[b],
Herman BRUYNINCKX[b] and Joris De SCHUTTER[b]
[a] *Universiteit Gent, SYSTeMS group, B-9052 Zwijnaarde, Gent, Belgium*
*E-mail: mila.mihaylova@ieee.org*
[b] *Katholieke Universiteit Leuven, Dept. of Mechanical Engineering, Celestijnenlaan 300B, Belgium*

**Abstract.** Active robotic sensing is a large field aimed at providing robotic systems with tools and methods for decision making under uncertainty, e.g. in a changing environment and with a lack of sufficient information. Active sensing (AS) incorporates the following aspects: (*i*) where to position sensors, (*ii*) how to make decisions for subsequent actions in order to extract maximum information from the sensor data and minimize costs such as travel time and energy. We concentrate on the second aspect: "where should the robot move at the next time step?" and present AS in a probabilistic decision theoretic framework. The AS problem is formulated as a constrained optimization with a multi-objective criterion combining an information gain and a cost term with respect to generated actions. Solutions for AS of autonomous mobile robots are given, illustrating the framework.

**Keywords.** Active sensing, autonomous robots, data fusion, estimation, decision making, information criterion, Bayesian method

## 1. Introduction

For a long time the goal of building autonomous robotic navigation systems has been central to the robotics community. In order to perform different tasks, autonomous robots need to move safely from one location to another. This is only possible when the robot is equipped with sensors, e.g. cameras, encoders, gyroscopes, contact or force sensors. To perform a task, the robot first needs to know: "Where am I now?" After that, the robot needs to decide "What to do next?" and to perform a sequence of actions. The latter decision making process is called *active sensing* (AS). The *active sensing*, *active perception* paradigm is introduced by Bajcsy [1,2], Aloimonous et al. [3], and Ballard [4] in a context of a task-directed choice of the controllable parameters to a sensing system. The developed methods aim at adaptively changing camera parameters (e.g. positions, focus or aperture), and at efficient data processing in order to improve perception. Either the sensor parameters or the processing resources allocated to the system are controlled [5].

An *action* is a particular kind of event leading to a change in the robot state or in the state of the world. The states capture all the information relevant to the robot decision-making process. Previewing both immediate and long-term effects is a requisite

for choosing actions: the robot should take both actions to bring itself closer to its *task completion*: e.g. reaching a goal position within a certain tolerance, and actions for the purpose of *information gathering*, such as searching for a landmark, surrounding obstacles, reading signs in a room, in order to keep its uncertainty small enough at each time instant. The robot should then be able to deal with static as well as with unknown and dynamic obstacles (e.g. moving people) and in this way perform *quickly changing tasks* in a *quickly changing environment*. Examples are: *mobile robot* navigation where the robot has to move safely and quickly under uncertainty; *industrial robot* tasks in which the robot is uncertain about the positions and orientations of its tools and work pieces, e.g. drilling, welding, polishing [6]; *vision applications*: active selection of camera parameters such as focal length and viewing angle to improve object recognition procedures [7,8].

*Active vision* (or *active sensing* in general) refers to the control of sensing parameters to improve the robustness of the feature extraction process. Active vision applications can be divided into four major classes [5]: *active vision*, *active perception*, *animate vision*, *purposive vision*. *Active vision* introduced by Aloimonous et al. [3] is a mathematical analysis of complex problems such as stability, linearity and uniqueness of solutions. A large group of active vision methods focus on the search for a solution in the *robot configuration space* (the space describing all possible positions of the robot). Other methods [9] center their work on the *image feature space*. According to the position of the camera, methods may employ a stationary camera head or a moving camera around the object so as to look at it from different viewpoints, the so-called *viewpoint planning* [10]. The goal of *active perception* as defined by Bajcsy [2] is to elaborate strategies for setting sensor parameters in order to improve the knowledge of the environment. *Animate vision* [4] is based on the analysis of human perception. The aim of *purposive vision* is to extract from the environment the information needed to ensure the realization of the task at hand. However, visual information includes uncertainty caused by quantization or calibration errors. In addition, visual processing is costly because the amount of image data is large, and because of the relatively complicated reasoning involved. When navigating using visual information, the robot has to reach a reasonable compromise between safety and efficiency [11]. If the robot reduces the number of observations in order to move fast, this can lead to cumulative motion uncertainty. On the other hand, the augmentation of the number of observations leads to safer motion, but arrival at the goal configuration will be delayed.

### *1.1. Estimation, Control and Active Sensing*

The inherent modules of an intelligent sensing system are estimation, control and active sensing. The *estimation* part is presented by stochastic estimators, which are based on the sensor and robot models, and after fusing the sensor data, generate estimates of the robot states and parameters. Knowing the desired task, the *controller* is charged with the task completion as accurately as possible. Motion execution can be achieved either by feedforward commands, feedback control or a combination of both [12]. Next, *active sensing* (AS) is the process of determining the inputs by optimizing an optimality criterion. These inputs are then sent to the controller [13,6,14]. AS is a *decision making process*, made at each time instant, or delayed to some time period [15], or performed upon request, after processing data from one or more sensors [16,17] through *multi-sensor data fusion*. AS is challenging for many reasons: *nonlinearity* of the robot and sensor models; need of an *optimality criterion* able to account for information gain and

other costs (such as travel distance, distance to obstacles); the high *computational load* (time, number of operations), especially important for *on-line* tasks; *uncertainties* in the robot model, the environment model and the sensor data; often measurements do not supply information about all variables, i.e. the system is *partially observable*; *geometric sensing* problems [17] such as those requiring a description of the shape, size and position of objects.

The rest of the paper is organized as follows. Section 2 formulates the AS problem within a statistical framework and considers the most often used optimality criteria for information extraction. Section 3 presents the main groups of optimization algorithms for AS. Section 4 gives examples and finally, Section 5 terminates with conclusions.

## 2. Problem Formulation

*Active robotic sensing* can be considered as a trajectory generation for a *stochastic dynamic* system described by the model

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \eta_k), \tag{1}$$

$$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}, \mathbf{s}_{k+1}, \xi_{k+1}), \tag{2}$$

where $\mathbf{x}$ is the system state vector, $\mathbf{f}$ and $\mathbf{h}$ are in general nonlinear system and measurement functions, $\mathbf{z}$ is the measurement vector, $\eta$ and $\xi$ are, respectively, system and measurement noise (*additive* or *multiplicative*), with covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$. $\mathbf{u}$ denotes the input vector of the state function (e.g. the robot speed), $\mathbf{s}$ stands for a sensor parameter vector as input of the measurement function (an example is the focal length of a camera). Subscript $k$ denotes the time step. Further, we denote both $\mathbf{u}$ and $\mathbf{s}$ inputs to the system with $\mathbf{a}$ (actions). The sensors introduce uncertainties due to *statistical errors* (usually well modeled by probability measures) and *quantization errors* (friction or other uncertainties that are more difficult to model via statistical methods) [17].

A *multi-objective performance criterion* (often called *value function*) is needed to quantify, for each sequence of actions $\mathbf{a}_1, \ldots, \mathbf{a}_N$ (also called *policy*), both the information gain and some costs in task execution:

$$J(\mathbf{x}, \mathbf{z}) = \min_{\mathbf{a}_1, \ldots, \mathbf{a}_N} \{\sum_j \alpha_j U_j + \sum_l \beta_l C_l\}. \tag{3}$$

This measure is general and appropriate for almost all sensing tasks, without assumptions as to particular sensing modality or the task at hand. It is composed of a weighted sum of *rewards*: (*i*) terms $U_j$ characterizing the minimization of *expected uncertainties* (maximization of *expected information extraction*) and (*ii*) terms $C_l$ denoting other *expected costs*, such as the travel distance, time, energy, distances to ob-

stacles, distance to the goal. Both $U_j$ and $C_l$ are function of the policy $\mathbf{a}_1,...,\mathbf{a}_N$. The weighting coefficients $\alpha_j$ and $\beta_l$ give a different impact to the two parts, and are chosen by the designer according to the task context. When the state at the goal configuration fully determines the rewards, $U_j$ and $C_l$ are computed based on this state only. When attention is paid to both the goal configuration and the intermediate time evolution, the terms $U_j$ and $C_l$ are functions of the robot state at different time steps $k$.

Criterion (3) is to be minimized with respect to the sequence of actions under *constraints* $\mathbf{c}(\mathbf{x}_1,...,\mathbf{x}_N,\mathbf{a}_1,...,\mathbf{a}_N) \leq \mathbf{c}_{thr}$, where $\mathbf{c}$ is a vector of physical variables that cannot exceed some threshold values $\mathbf{c}_{thr}$, e.g. maximum allowed velocities and acceleration.

### *2.1. Action Sequence*

The description of the actions $\mathbf{a}_1,\ldots,\mathbf{a}_N$ can be given in the configuration space in different ways and has a major impact on the optimization problem that will be solved afterwards (Section 3). Two major groups of methods can be distinguished: with *parameterized* and *nonparameterized* sequence of motions. Within the framework of the *parameterized* sequence of actions the AS problem is reduced to a finite-dimensional parameter optimization. The robot trajectory is considered as composed of primitives (finite sine/cosine series [12], elliptic or other functions with appealing properties) the parameters of which are searched for in order to satisfy an optimality criterion. The choice of "where to look next" can be treated as a case of an optimal experiment design [6].

The methods for *nonparameterized* actions generate a sequence of freely chosen actions that are *not* restricted to a certain form of trajectory [18]. Constraints, such as maximum acceleration and maximum velocity, can be added to produce executable trajectories. A general framework for dealing with uncertainties involves optimization problems using *Markov decision processes (MDPs)* and *partially observable MDPs (POMDPs)* [19,20]. Often used are probabilistic metrics as a decision making criterion, e.g. the Shannon entropy [21]. However, computational complexity makes the POMDPs intractable for systems with many states. A mobile robot operating in the real world may have millions of possible states. Hence, exact solutions can only be found for (PO)MDPs with a small number of states. Larger problems require *approximate solutions*, like [22], and hierarchical POMDPs.

Both parameterized and nonparameterized methods can generate *locally* and *globally* optimal trajectories depending on the length of the path along which the optimization is performed, i.e. the trajectory is optimal in some segments with respect to the performance criterion or along the whole path. Optimal is the action that continuously directs the vehicle towards the maximum increase of information.

### 2.2. Performance Criteria Related to Uncertainty

Considered in a stochastic framework, the outcome of an action is a random change in the robot state. The outcome of an action can be characterized by the terms $U_j$ that represent: the expected uncertainty about the state or this uncertainty compared to the accuracy needed for the task completion. Due to different uncertainties, a natural tool for AS is the Bayesian framework, in which the characterization of the accuracy of the estimate is based on a scalar loss function of its probability density function. Different stochastic estimators can be applied for calculating the term $U_j$, such as the standard Kalman filter (KF), extended or iterated KFs [23,6], the unscented KF [24], and Monte Carlo techniques [25]. Since no scalar function can capture all aspects of the information extraction, no function suits the needs of every experiment. Commonly used functions are ***based on the covariance matrix*** of a stochastic estimator. The covariance matrix $\mathbf{P}$ of the state vector $\mathbf{x}$ is a measure for the uncertainty of the estimate. AS is looking for actions which minimize the uncertainty, i.e. the covariance matrix $\mathbf{P}$ or the inverse of the *Fisher information matrix* $\mathbf{I} = \mathbf{P}^{-1}$ [26,27]. Several scalar functions of a covariance matrix can be used [28]: *(i) D-optimal design*: minimizes the matrix determinant, $det(\mathbf{P})$, or the logarithm of it, $log(det(\mathbf{P}))$. It is *invariant* to any nonlinear transformation of the state vector $\mathbf{x}$ with a non-singular Jacobian, but is not invariant to physical units (when the elements of $\mathbf{x}$ have different units: as meters for position, degrees for angles), nor is it suitable for verifying whether the task has been completed. $det(\mathbf{P})$ being smaller than a certain value does not guarantee that the covariances of the state variables will be smaller than their toleranced value. *(ii) A-optimal design*: minimizes the trace $tr(\mathbf{P})$. A-optimal design does not have the invariance property when states have inconsistent units, nor does this measure allow to verify task completion. *(iii) L-optimal design*: minimizes the weighted trace $tr(\mathbf{WP})$. A proper choice of the weighting matrix $\mathbf{W} = \mathbf{MN}$ can render the L-optimal design criterion *invariant* to transformations of the state vector $\mathbf{x}$ with a non-singular Jacobian. The covariance matrix $\mathbf{P}$ is normalized and scaled [6,29].

The product of the normalizing matrix $\mathbf{N}$ and $\mathbf{P}$ is invariant to physical units. The scaling matrix $\mathbf{M}$ scales the elements of the product within a preset range. *Tolerance-weighted L-optimal design* [6] proposes a natural choice of $\mathbf{W}$ depending on desired tolerances of task completion. *(iv) E-optimal design*: minimizes the maximum eigenvalue $\lambda_{max}(\mathbf{P})$. Like A-optimal design, this function is not invariant to transformations of $\mathbf{x}$, but it allows for the verification of task completion.

The second large group of information functions are ***based on a probability density function*** $p(\mathbf{x})$. The Shannon entropy [21]

$$H[p(\mathbf{x})] = -\int_{-\infty}^{\infty} p(\mathbf{x}) log(p(\mathbf{x})) dx \tag{4}$$

gives the average *information* or the *uncertainty* of a random variable. Introduced to quantify the transmission of information in communication channels [21], the entropy

is successfully applied in many fields, including robotics and computer vision. In vision, entropy is used to qualify the view of a scene and determines *the next best view* (the one that obtains maximum information of a scene) [30]. In light of the AS process, entropy can characterize the robot's knowledge about its location in the environment. Entropy based performance criteria are:

- the *entropy* of the posterior distribution $p_{post}(\mathbf{x})$: $E[-\log p_{post}(\mathbf{x})]$, where $E[.]$ indicates the expected value.
- the *change in entropy* between two distributions: $p_1(\mathbf{x})$, the prior and $p_2(\mathbf{x})$, the posterior: $E[-\log p_2(\mathbf{x})] - E[-\log p_1(\mathbf{x})]$.
- the *Kullback-Leibler distance* [31] (also called *relative entropy*) is a measure for the goodness of fit or closeness of two distributions: $E\left[\log(p_2(\mathbf{x}) / p_1(\mathbf{x}))\right]$. The expected value is calculated with respect to $p_2(\mathbf{x})$.

The relative entropy and the change in entropy are *different* measures. The *change in entropy* only quantifies how much the form of the probability distributions changes whereas the *relative entropy* also represents a measure of how much the distribution has moved. If $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ are the same distributions, translated by different mean values, the change in entropy is zero, while the relative entropy is not. The computation of the probability density and entropy can be performed by Monte Carlo sample-based stochastic estimators [25,13,14].

## 3. Optimization Algorithms for Active Sensing

Active sensing corresponds to a constrained optimization of $J$ with respect to the policy $\mathbf{a}_1,\ldots,\mathbf{a}_N$. Depending on the robot task, sensors and uncertainties, different *constrained optimization problems* arise.

If the sequence of actions $\mathbf{a}_1,\ldots,\mathbf{a}_N$ is restricted to a *parameterized trajectory*, the optimization can be expressed as: linear programming, constrained nonlinear least squares methods, convex optimization, etc. [32]. Examples of problems formulated as optimization with respect to a set of parameters are dynamical robot identification [27] and generation of a sinusoidal mobile robot trajectory [29], where the solution is searched for within the optimal experiment design.

If the sequence of actions $\mathbf{a}_1,\ldots,\mathbf{a}_N$ is *not restricted to a parameterized trajectory*, then the optimization problem has a different structure and is within the Markov Decision Processes (MDPs) framework [18]. MDPs serve as a background for solving complex problems with incomplete information about the robotic system. A *Markov decision process* can be described [18,22] as a tuple $\langle X, A, Pr, R\rangle$ where $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$ is a finite set of states of the system which evolves stochastically; $A$ is a finite set of actions; $Pr: X \times A \longrightarrow \Pi(X)$ is a state-transition function, mapping an action and a state to a probability distribution over $X$ for the

possible resulting state. The Markovian transition probability $Pr(\mathbf{x}' \mid \mathbf{x}, \mathbf{a})$ represents the probability of going from state $\mathbf{x}$ to state $\mathbf{x}'$ with an action $\mathbf{a}$. To judge the quality of an action, a reward function is introduced, $R: X \times A \longrightarrow R$. It gives the immediate reward obtained by the agent (decision maker) in state $\mathbf{x}$ after taking an action $\mathbf{a}$. The next state and the expected reward depend only on the previous state and the action taken (Markov property).

A *policy* for a MDP is a mapping $\pi: X \longrightarrow A$ that selects an action for each state. Given a policy, a finite-horizon *value function* of the state can be defined such that $V_n^{\pi}: X \longrightarrow R$, where $V_n^{\pi}(\mathbf{x})$ is the expected value of applying the policy $\pi$ for $n$ steps starting in $\mathbf{x}$. Then a value function can be inductively written with $V_0^{\pi}(\mathbf{x}) = R(\mathbf{x}, \pi(\mathbf{x}))$ and

$$V_m^{\pi}(\mathbf{x}) = R(\mathbf{x}, \pi(\mathbf{x})) + \sum_{\mathbf{x}' \in X} Pr(\mathbf{x}, \pi(\mathbf{x}), \mathbf{x}') V_{m-1}^{\pi}(\mathbf{x}').$$

A policy $\pi$ is considered to be better than a policy $\pi'$, if for all $\mathbf{x} \in X$, $V^{\pi}(\mathbf{x}) \geq V^{\pi'}(\mathbf{x})$, and for at least one $\mathbf{x} \in X$, $V^{\pi}(\mathbf{x}) > V^{\pi'}(\mathbf{x})$. This means that a policy is optimal if it is not dominated by another policy.

Consider the following optimization problems that require solving in the MDPs: a finite-horizon, i.e. over a fixed finite number of time steps ($N$ is finite), or an infinite-horizon problem ($N = \infty$). For every state it is rather straightforward to know the immediate reward being associated to every action (1 step policy). The goal however is to find the policy that maximizes the reward over the long-term (N steps). Different optimization procedures exist for these kinds of problems, the most popular of which are:

- *value iteration:* it is a *dynamic programming* algorithm, that recursively calculates the optimal value function and policy [33]. The optimization is formulated as a sequence of problems to be solved with only one of the N variables $a_i$;
- *policy iteration:* is an iteration technique [34] over policies for infinite horizon systems. The current policy is improved repeatedly. The initial policy is chosen at random, and the process terminates when no improvement can be achieved;
- *linear programming approach:* formulates and solves a MDP as a linear program. In practice, policy iteration tends to be faster than the linear programming approach;
- *state based search methods:* represent the system as a graph whose nodes correspond to states. *Tree search* algorithms then search for the optimal path in the graph and can handle finite and infinite horizon problems [22].

In the *fully observable* MDPs the agent accurately knows what state it is in at each instant. When the information about the system state is incomplete or noisy, the solution is searched for in the group of *partially observable* MDPs (POMDPs). At each time step the state of the system is not known, only a probability distribution over the

states can be calculated. An optimal policy for every possible probability distribution at each time step is needed. This constitutes the key problem of this representation. In most real applications the set of states is quite large and the (PO)MDPs representation is computationally expensive and not feasible. Making approximations is the only way to apply these algorithms to real systems. Exact solutions can only be found for (PO)MDPs with a small number of (discretized) states. For larger problems *approximate solutions* are needed, e.g. [22]. Solutions to POMDPs are usually obtained by applying dynamic programming or by solving directly the Bellman equation [33]. In the search for more elegant schemes, hierarchical POMDPs are proposed. *Bayesian networks* are other tools allowing compact representation of the transitions.

## 4. Examples

Results from a *covariance-based parameterized* approach for trajectory generation [29] and an *entropy-based* method [13] are presented.

*Example 1.* Distance and orientation sensing of a mobile robot to known beacons is addressed [29]. We consider the trajectory generation of a nonholonomic wheeled mobile robot (WMR), moving from a starting configuration $(x_s, y_s, \phi_s)^T$ (position and orientation) to a goal configuration $(x_g, y_g, \phi_g)^T$, around a known nominal reference trajectory $(x_{r,k}, y_{r,k}, \phi_{r,k})^T$. The vehicle motion is described by the model

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{pmatrix} = \begin{pmatrix} x_k + v_k \Delta T cos(\phi_k + \psi_k) \\ y_k + v_k \Delta T sin(\phi_k + \psi_k) \\ \phi_k + \frac{v_k \Delta T}{L} sin\psi_k \end{pmatrix} + \begin{pmatrix} \eta_{x,k} \\ \eta_{y,k} \\ \eta_{\phi,k} \end{pmatrix},$$

with $x_k$ and $y_k$ the WMR position coordinates relative to a fixed frame (Fig. 1 (a)), and $\phi_k$ the orientation angle with respect to the $x$ axis. They form the state vector $\mathbf{x}_k = (x_k, y_k, \phi_k)^T$. $L$ represents the wheelbase (the distance between the front steering wheel and the axis of the driving wheels), $\Delta T$ is the sampling interval, $\eta_k = (\eta_{x,k}, \eta_{x,k}, \eta_{\phi,k})^T$ is the process noise. The WMR is controlled through a desired velocity $v_k$ and a direction of travel $\psi_k$, written into the control vector $\mathbf{u}_k = (v_k, \psi_k)^T$. Due to physical constraints, $v_k$ and $\psi_k$ cannot exceed boundary values: $v_k \in [0, v_{max}]$, $\psi_k \in [-\psi_{max}, \psi_{max}]$ ($\psi_{max} \leq \frac{\pi}{2}$). The WMR can only perform forward motions. The vehicle is equipped with a sensor measuring the range $r_k$ and bearing $\theta_k$ to a beacon $B$, located at known coordinates, $(x_B, y_B)^T$. The observation equation for the beacon is

$$\begin{pmatrix} r_k \\ \theta_k \end{pmatrix} = \begin{pmatrix} \sqrt{(x_B - x_k)^2 + (y_B - y_k)^2} \\ \arctan(\frac{y_B - y_k}{x_B - x_k}) - \phi_k \end{pmatrix} + \begin{pmatrix} \xi_{r,k} \\ \xi_{\theta,k} \end{pmatrix},$$

where $\mathbf{z}_k = (r_k, \theta_k)^T$ is the measurement vector and $\xi_k = (\xi_{r,k}, \xi_{\theta,k})^T$ is the observation noise. $\eta_k$ and $\xi_k$ are assumed Gaussian, zero mean, mutually uncorrelated, with covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively.

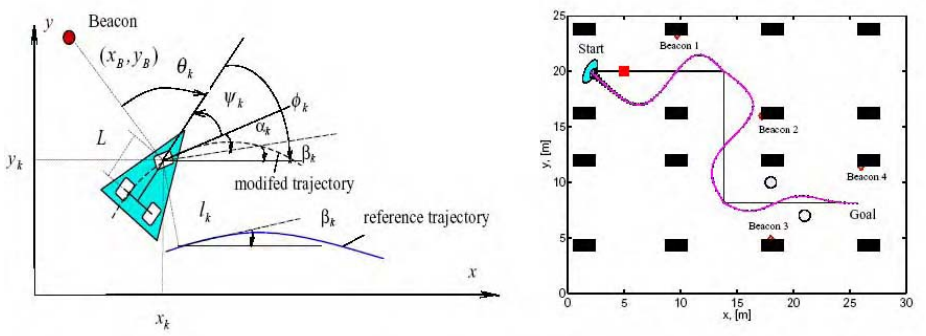



**Figure 1.** (a) WMR coordinates (b) Trajectory in the presence of multiple obstacles.

The beacon location with respect to the WMR is of paramount importance for the AS task, for the accuracy and informativeness of the data.

The optimal trajectory is searched [29] in the class $Q(\mathbf{p})$ of harmonic functions, where $\mathbf{p}$ is a vector of parameters obeying preset physical constraints. With $N$ the number of functions, the new (modified) robot trajectory is generated on the basis of a reference trajectory by the lateral deviation $l_k$ (*lateral* is the orthogonal robot motion deviation from the reference trajectory in $y$ direction) as a linear superposition

$$l_k = \sum_{i=1}^{N} A_i sin(i\pi \frac{s_{r,k}}{s_{r,total}}), \tag{5}$$

of sinusoids, with constant amplitudes $A_i$, $s_{r,k}$ is the reference path length up to instant $k$, $s_{r,total}$ is the total path length, and $r$ refers to the reference trajectory. In this formulation AS is a *global optimization problem* (on the whole robot trajectory) with a criterion

$$J = \min_{A_{i,k}} \{\alpha_1 U + \alpha_2 C\} \tag{6}$$

to be minimized under *constraints* (for the robot velocity, steering angle, orientation angle, distance to obstacles). $\alpha_1$ and $\alpha_2$ are dimensionless positive weighting coefficients. Here $U$ is in the form

$$U = tr(\mathbf{WP}), \tag{7}$$

where $\mathbf{P}$ is the covariance matrix of the estimated states (at the goal configuration), computed by an unscented Kalman filter [24]. $\mathbf{W}$ is a weighting matrix. The cost term $C$ is assumed to be the relative time

$$C = t_{total}/t_{r,total}, \tag{8}$$

where $t_{total}$ is the total time for reaching the goal configuration on the modified trajectory, and $t_{r,total}$ is the respective time over the reference trajectory. The weighting matrix $\mathbf{W}$ is presented as a product of a normalizing matrix $\mathbf{N}$, and a scaling matrix $\mathbf{M}$, $\mathbf{W} = \mathbf{MN}$, with the matrix $\mathbf{N} = diag\{1/\sigma_1^2, 1/\sigma_2^2, ..., \sigma_n^2\}$. $\sigma_i$, $i = 1, ..., n$, are assumed here to be the standard deviations at the goal configuration on the reference trajectory and $\mathbf{M}$ is the unit matrix.

Simulation results for a *straight line* reference trajectory, and the modified trajectory, generated with different number of sinusoids $N$ (in accordance with (5), and criterion (3) with terms (7) and (8)) together with the uncertainty ellipses are shown in Fig. 2 (a). The evolution of the weighted covariance trace is presented in Fig. 2 (b). The multisine approach gives higher accuracy, than the straight-line trajectory. As it is seen from Figures 2 (a), (b) the most accurate results at the goal configuration for $U$ are obtained with $N = 5$ sinusoids. Better accuracy is provided with bigger $N$, at the cost of increased computational load. The plot in Fig. 1 (b) gives a more complex robot trajectory in an environment with many obstacles and landmarks. The robot trajectory in this storehouse is generated as composed of different segments, three in this case (Fig. 1 (b)). Even complex and long trajectories can be reduced to a sequence of straight line segments. The end conditions of the first segment are initial conditions for the second segment and similarly for the other intermediate linking points. The criterion $U$ is in the form (7). In order to insure informative sensor measurements, each of the three segments is provided with two beacons. The minimum distance to obstacles should not be less than $0.5m$. The generated trajectory (with $N = 3$) and the $3\sigma$ uncertainty ellipses around it are plotted in Fig. 1 (b). The robot moves in the direction of increasing information (toward the beacons), thereby decreasing the uncertainties, which is demonstrated by the changing size of the uncertainty ellipses. The trajectories generated by the parameterized sequence of actions are *smooth*, always obeying the position constraints at the starting and goal configurations. The multisine approach is aimed at applications where enough freedom of motion is available, e.g. in large indoor and outdoor environments. No considerable gain will be reached in very constrained environments, such as very narrow corridors or very cluttered environments. The fewer

the constraints, the more effective the approach. This is obvious when we account for the fact that the lateral deviation is bounded by a threshold.

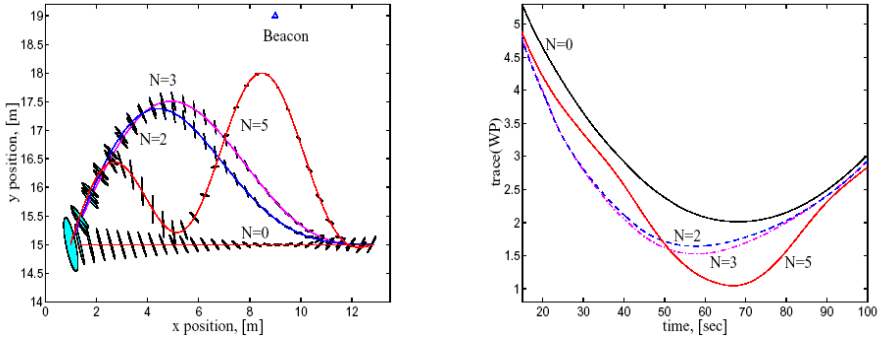



**Figure 2.** Robot trajectories and information criterion: a) Trajectories, generated with (5), b) Evolution of trace(WP) (8) in time, and different number N. Uncertainty ellipses are plotted around trajectories.

*Example 2.* This example illustrates AS based on entropy and cost minimization for the Minerva mobile robot [13], Fig. 3 (a), with positional uncertainty. The technique [13] is called *coastal* navigation by analogy to the navigation of ships. Ships often use the coastal parts of the land to determine where they are, when other advanced tools such as GPS systems are not available. Similarly, mobile robots need to position themselves in dynamic environments with changing obstacles such as people. The robot uses coast lines in the environment which contain enough information for accurate localization and moves close to areas of the map with high information content. The robot coastal planning technique comprises the steps: (*i*) the information content of the environment is modeled while accounting for the sensor features and possible dynamic obstacles; (*ii*) trajectories are generated by the information model of the environment and obstacle information in the map.

The sensors used to generate the map of the environment (the National Museum of American History) are laser range finders. Each cell of the map is characterized by information content, corresponding to the ability of the robot to localize itself.

The state vector $\mathbf{x}$ of the robot contains the position coordinate $(x, y)$, and the direction $\theta$. The robot acquires a range data $\mathbf{z}$ from a laser sensor. The method [13] generates a map of the environment that contains the information content of each robot position. The information of the robot's current position is characterized by the difference

$$U = E(H(p_{\mathbf{x}/\mathbf{z}})) - H(p_{\mathbf{x}}) \tag{9}$$

between the expected entropy of the positional probability conditioned on the sensor measurements, $E(H(p_{\mathbf{x}/\mathbf{z}}))$, and the entropy of the prior distribution $H(p_{\mathbf{x}})$.

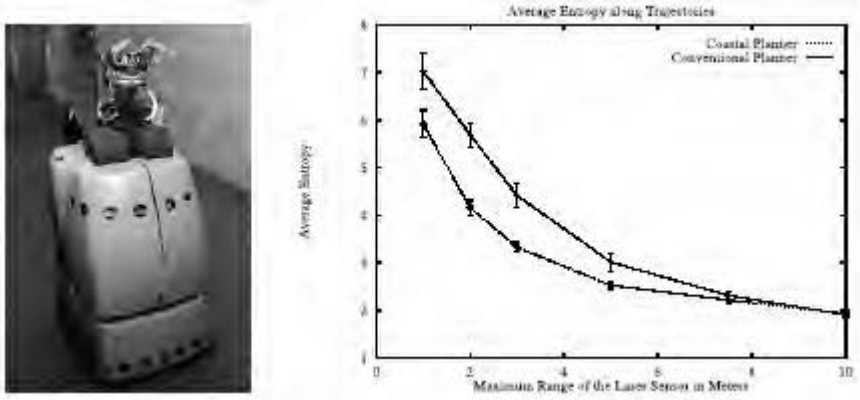



**Figure 3.** (a) Minerva robot (b) Average entropy over trajectories in the museum.

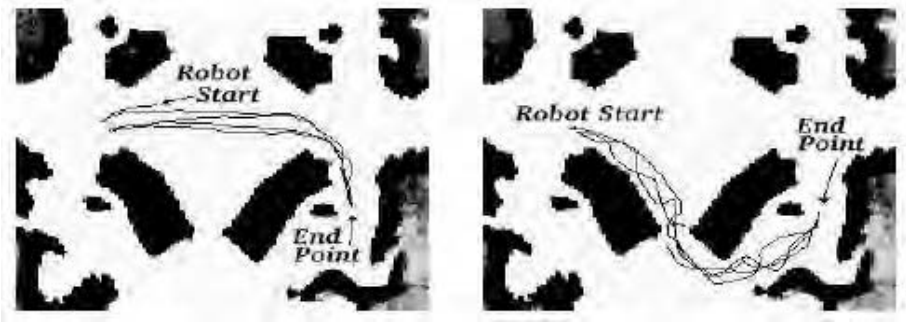



**Figure 4.** Robot trajectories (courtesy of W. Burgard, Univ. of Freiburg, Germany).

This term, together with a cost associated with travel, both weighted with coefficients, as in (3), are to be minimized in *coastal planning*. For *conventional planning*, only the cost term is accounted for. The cost term is formed by the probabilities of the map cells. The function to be minimized in the considered conventional planner is the cost of crossing cell $(x_i, y_i)$, the value of which increases when the probability that the cell is occupied is high.

Fig. 4 presents the robot trajectories generated by conventional and coastal planning for the same start and goal locations. The robot trajectory shown in Fig. 4 (a) is a line through the open space, whereas from Fig. 4 (b) it is seen that the robot does not travel directly through the open space but moves in such a way so as to maximize the information content. The black areas show obstacles and walls, the light grey areas are areas where no information is available to the sensors. The darker grey the area, the better the information gain from the sensors. As seen, the robot is following trajectories with a lower average entropy. Fig. 3 (b) presents the average entropy as a function of the maximum range of the laser sensor. It illustrates the uncertainty reduction during the sensing process with different sensor abilities, in a static environment, for both coastal and conventional sensing. When a range sensor with an increased range (up to 10 *m*) is used, the results from coastal and conventional sensing are comparable and in

this case the use of conventional navigation is recommended. In other outdoor conditions, coastal navigation might be more suitable.

The multisine approach has a *global* type of planning: it generates the whole trajectory and makes use of a sequential quadratic programming method for optimization. The approach from *Example 2* has a *local* type of planning (at each time instant the information criterion is computed and a decision for the robot movement is made) based on optimization via dynamic programming (the Vitterbi algorithm).

## 5. Conclusions

Active robotic sensing incorporates various aspects and has many applications such as active vision and autonomous robot navigation, among others. In this paper we present a probabilistic decision theoretic framework for making decisions. AS is considered as a multi-objective optimization process for determining whether the result of an action is better than the result of another. Frequently used statistical decision-making strategies are considered. Even though AS tasks are fairly different depending on the specific sensors, and applications, e.g. comparing force-controlled manipulation to that of autonomous mobile robot navigation, usually the optimality criteria are composed of two terms: a term characterizing the uncertainty minimization, i.e., maximization of information content, and a term for costs, such as traveled path or total time. Further investigations are directed toward AS tasks with one and multiple robots in a dynamic environment. AS of multiple robots requires coordination, synchronization, and collision avoidance. New solutions are needed in order to reach a reasonable balance between complexity and high performance.

## Acknowledgements

Financial support of the Fund for Scientific Research-Flanders (F.W.O–Vlaanderen) in Belgium, K. U. Leuven's Concerted Research Action GOA-99/04 and Center of Excellence BIS21/ICA1-2000-70016 are gratefully acknowledged.

## References

[1] Bajcsy, R., Real-time obstacle avoidance algorithm for visual navigation, Proc. of the 3rd Workshop on Computer Vision: Represent. and Control, pp. 55–59, 1985.
[2] Bajcsy, R., Active perception, Proc. of the IEEE, Vol. 76, pp. 996–1005, 1988.
[3] Aloimonous, Y., I. Weiss and A. Bandopadhay, Active vision, Intern. J. of Computer Vision, 1987, Vol. 1, No. 4, pp. 333–356.
[4] D. Ballard, Animate vision, Artificial Intelligence, Vol. 48, No. 1, pp. 57–86, 1991.
[5] Marchand, E., F. Chaumette, An autonomous active vision system for complete and accurate 3D scene reconstruction, Int. J. of Comp. Vision, pp. 171–194, 1999.
[6] De Geeter, J., J. De Schutter, H. Bruyninckx, H.V. Brussel, M. Decreton, Tolerance-weighted *L*-optimal experiment design: a new approach to task-directed sensing, Adv. Robotics, Vol. 13, No. 4, pp. 401–416, 1999.
[7] DeSouza, G., A. Kak, Vision for mobile robot navigation: A survey, IEEE Trans. PAMI, Vol. 24, No. 2, pp. 237–267, 2002.
[8] Denzler, J., C. Brown, Information theoretic sensor data selection for active object recognition and state estimation, IEEE Trans. PAMI, pp. 145–157, 2002.

[9] Zhang, H., J. Ostrowski, Visual motion planning for mobile robots, IEEE Trans. on Rob. and Aut., Vol. 18, No. 2, pp. 199–207, 2002.
[10] Madsen, C., H. Christensen, A viewpoint planning strategy for determining true angles on polyhedral objects by camera alignment, IEEE Trans. PAMI, Vol. 19, No. 2, pp. 158–163, 1997.
[11] Moon, I., J. Miura, Y. Shirai, On-line viewpoint and motion planning for efficient visual navigation under uncertainty, Rob. and Aut. Syst., pp. 237–248, 1999.
[12] Laumond, J.-P., Robot motion planning and control, Springer-Verlag, 1998.
[13] Roy, N., W. Burgard, D. Fox, S. Thrun, Coastal navigation – mobile robot navigation with uncertainty in dynamic environments, Proc. of IEEE Int. Conf. on Rob. and Aut., 1999.
[14] Burgard, W., D. Fox, S. Thrun, Active mobile robot localization by entropy minimization, Proc. of the 2nd Euromicro Workshop on Adv. Mobile Rob., 1997.
[15] Liu, S., and L. Holloway, Active sensing policies for stochastic systems, IEEE Trans. on AC, Vol. 47, No. 2, pp. 373–377, 2002.
[16] Lim, H.-L., L. Holloway, Active sensing for uncertain systems under bounded-uncertainty sensing goals, Prepr. of the 13th World Congr. of IFAC, USA, 1996.
[17] Hager, G., M. Mintz, Computational methods for task-directed sensor data fusion and sensor planning, Intern. J. Rob. Research, Vol. 10, No. 4, pp. 285–313, 1991.
[18] Kaelbling, L., M. Littman, A. Cassandra, Planning and acting in partially observable stochastic domains, Artif. Intell., Vol. 101, No. 1–2, pp. 99–134, 1998.
[19] Cassandra, A., L. Kaelbling, M. Littman, Acting optimally in partially observable stochastic domains, Proc. of the 12 Nat. Conf. on AI, pp. 1023–1028, 1994.
[20] Thrun, S., Monte Carlo POMDPs, Adv. in Neural Inf. Proc. Syst. 12, MIT Press, pp. 1064–1070, 1999.
[21] Shannon, C., A mathematical theory of communication, I and II, The Bell System Techn. Journ., Vol. 27, pp. 379–423 and pp. 623–656, 1948.
[22] Boutilier, C., T. Dean, and S. Hanks, Decision-theoretic planning: structural assumptions and computational leverage, J. of AI Res., Vol. 11, pp. 1–94, 1999.
[23] Bar-Shalom, Y., X.R. Li, Estimation and tracking: principles, techniques and software, Artech House, 1993.
[24] Julier, S., The scaled UT, Proc. of the Amer. Contr. Conf., pp. 4555–4559, 2002.
[25] Doucet, A., N. de Freitas, N. Gordon, Eds., Sequential Monte Carlo Methods in practice, Springer-Verlag, 2001.
[26] Fisher, R., On the mathematical foundations of theoretical statistics, Philosophical Trans. of the Royal Society of London – A, Vol. 222, pp. 309–368, 1922.
[27] Swevers, J., C. Ganseman, D. Bilgin, J. De Schutter, H.V. Brussel, Optimal robot excitation and identification, IEEE Trans. on AC, Vol. 13, pp. 730–740, 1997.
[28] Fedorov, V., Theory of optimal experiments, Acad. press, NY, 1972.
[29] Mihaylova, L., J. De Schutter, H. Bruyninckx, A multisine approach for trajectory optimization based on information gain, Rob. and Aut. Syst., pp. 231–243, 2003.
[30] Vázquez, P., M. Feixas, M. Sbert, W. Heidrich, Viewpoint selection using viewpoint entropy, T. Ertl, B. Girod, G.Greiner, H. Niemann, H.-P. Seidel (Eds.) Vision, Modeling, and Visualization 2001, pp. 273–280.
[31] Kullback, S., On information and sufficiency, Ann. Math. Stat., pp. 79–86, 1951.
[32] NEOS, Argonne National Laboratory and Northwestern University, Optimization Technology Center, 2002, *http://www-fp.mcs.anl.gov/otc/Guide/*.
[33] Bellman R., Dynamic Programming, Princeton Univ. Press, 1957, New Jersey.
[34] Howard, R., Dynamic Programming and Markov Processes, The MIT Press, 1960.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# A New Genetic Algorithm for Global Optimization of Resources in Naval Warfare

David BOILY and Hannah MICHALSKA
*Lockheed-Martin Canada, McGill University*

**Abstract.** This paper introduces a novel Genetic Algorithm (GA) for time efficient calculation of a solution to a resource management (RM) problem in the context of naval warfare. The novelty resides in the introduction of a new operator to correct the behavior observed in Steady State Genetic Algorithms (SSGA). The SSGA model differs from the traditional model in that it simulates the dynamics of a population reproducing in a semi-random way. It has been observed that genetic diversity is lost within a few generations when an SSGA is implemented using a small population [5]. The main purpose of the new operator is the diversification of a population. Its performance is evaluated according to a measure of a population's diversity (entropy). The RM problem is also examined in detail; it is formulated as a non-linear optimization problem. The GA has been implemented using a proprietary data-driven multi-agent system, developed by Lockheed Martin Canada. The advantage of this novel GA over previous methods (TABU search) has been empirically confirmed by extensive simulations.

**Keywords.** Genetic algorithm, resource management, TABU, non-linear optimization, NP-complete, multi-agent, data-driven, naval warfare

## 1. Introduction

Modern warfare is becoming ever more technically sophisticated; modern defense must therefore deal with a varied array of threats as efficiently as possible. This is primarily enabled by the increased volume, rate and complexity of information provided by modern sensors. It is thus necessary to develop fast and reliable information processing algorithms.

A vast compendium of artificial intelligence techniques has been put to successful use in solving the problem of what to do with all that information before it is depreciated. The key characteristic of modern warfare is an ever increasing operations speed.

Resource Management (RM) is a fundamental component of an integrated decision support system. It involves planning for the allocation and scheduling of resources. A previous attempt to solve a specific RM problem for naval warfare [1], uses a simple construction heuristic to build an initial passable solution and to subsequently improve that solution with more or less advanced search methods. In contrast, the results presented in this paper rely on a more systematic approach and the use of a Genetic Algorithm (GA). The use of this type of algorithm was not attempted before as it was thought to require prohibitively long execution times. However, as is demonstrated here, the structure and data flow of the GA can be tailored in such a way as to meet stringent time limitations.

In its final form, the GA is shown to offer greater flexibility than previous techniques and provides for more accurate results. In this way, the approach presented here provides a practical proof that genetic algorithms can be used as part of real time decision support systems.

The paper is organized as follows. The resource management problem is defined, explained and formulated in sections 2 and 3. An overview of the techniques used in [1] is also presented. The algorithmic complexity of the problem is assessed, as in [2], while considering an independent graph problem with estimation of the time required for computation of the cost function. Later sections of the paper are dedicated to the definitions of the specific GA operators which prove to be useful in the solution of the RM problem at hand. It is also explained how to streamline their use toward obtaining a real-time (on-line) solution to the problem. The efficiency of the algorithm is assessed using realistic, randomly generated scenarios, such as those introduced in [1]. Numerous simulations demonstrate superior efficiency of this technique over previous methods.

## 2. The Resource Management Problem

The problem considered here concerns the allocation of resources in the context of naval defense. Specifically, the following scenario is considered. A preset number of anti-ship missiles (ASMs) are homing in on a naval target (ship) that is in command of its own defense. The ASMs are assumed to travel at the same constant speed of 850 m/s, and their trajectories are assumed to be linear in the direction of the ship. The defense system on board the ship can use three types of weapons: for long engagements [Surface-to-air missiles (SAMs)], for medium engagements [56mm gun], and for close-range engagements [close-in weapon system (CIWS)]. At any instant of time, the current positions of the ASMs, whose knowledge is required for an engagement to take place, is determined through the use of illuminators. Although the CIWS has its own illuminator, the SAMs and the gun must share only two illuminators. Due to the fact that each weapon necessitates an illuminator, these illuminators must be assigned to the right weapon at the right time in order to execute any hypothetical engagement. In this context the optimal warfare resource allocation problem is that of maximizing the overall probability that the ship survives the encounter with the attackers. For more technical details on the defense weapons and associated systems see the appendix in [1]. It is a non-trivial assumption that all the above described actions happen almost instantaneously, i.e. the position of the missiles will change only linearly between the times of their detection and interception.

### *2.1. The Engagement List*

An example defense scenario is presented below. This scenario will be used to explain how the allocation of the aforementioned weapon systems is carried out. The goal will be to maximize the survival probability of the ship. The ASMs in this scenario are described in Table 1.

**Table 1.** Threat specification

| | x (km) | y (km) | range (km) | ship hit (sec) |
|---|---|---|---|---|
| threat 0 | –16.91 | 14.13 | 22.04 | 25.9252 |
| threat 1 | –5.30 | 8.48 | 10.00 | 11.7647 |
| threat 2 | 7.30 | –7.48 | 10.45 | 12.2963 |

To decide on a course of action, the whole array of possible engagements must be generated, this is done using the recursive algorithm described in [1]. In [1] the algorithm was referred to as the cue generation algorithm (Gencue). Here, it shall be referred to as the Build Engagement List (BEL) algorithm, as this is what it does. An engagement is a (weapon, illuminator, target) triplet paired with a fire time and a kill assessment (KA) time. In effect, an engagement triplet determines, at a given fire time: which weapon (SAM, Gun or CIWS) and which illuminator (STIR A, STIR B or CIWS) is used against which target (threats 0, 1 or 2). The success of such an operation is assessed in terms of the (KA) time. For example, engagement 9 in Table 2 simply states that a SAM will be launched at 9.196 seconds using the second STIR illuminator and the success of the operation will assessed at 12.25 seconds. The BEL algorithm is basically an exhaustive search of engagements that obey kinematic laws and the following engagement doctrine [1]:

- neither a SAM nor the gun can engage or re-engage a threat until a kill assessment (KA) for a prior engagement is completed ("shoot-look-shoot");
- a threat cannot be engaged or re-engaged with a SAM or the gun if it is engaged with the CIWS;
- a threat can be engaged with the CIWS even if it is already engaged with a SAM or the Gun;
- the CIWS fires at a single threat until the threat is either destroyed or has hit the ship.

For details on how this algorithm works and is implemented see [1]. Given the threat specification in Table 1, the BEL algorithm generates the Engagement List presented in Table 2 which comprises $N = 20$ entries.

The formulae to obtain the probability of kill for each engagement are given and explained in the Appendix of [1]. It is important to note that a probability of success is assigned for every engagement in the Engagement List. The number of engagements $N$ grows exponentially with the number of threats. The illuminators have blind zones and, of course, the closer the threat, the less possible actions can be taken towards eliminating it. Thus $N$ also depends, albeit linearly, on the azimuthal angle and range of those threats.

Table 3 shows the average of $N$ given 100 randomly generated scenarios (threat range of 5 km to 40 km and any azimuthal angle) for each number of threats. Also shown is the average CPU time taken by the BEL algorithm to generate each Engagement List. It grows exponentially with the number of threats.

**Table 2.** Engagement List

| engagement number | fire time (sec) | KA time (sec) | threat | Illuminator | weapon | probability of kill |
|---|---|---|---|---|---|---|
| 0 | 5.294 | 11.71 | 1 | ILL_CIWS | CIWS | 0.7355 |
| 1 | 5.826 | 12.25 | 2 | ILL_CIWS | CIWS | 0.7355 |
| 2 | 5.947 | 11.71 | 1 | STIR1_A | Gun-4 | 0.6246 |
| 3 | 6.478 | 12.25 | 2 | STIR2_B | Gun-4 | 0.6246 |
| 4 | 7.083 | 11.71 | 1 | STIR1_A | Gun-3 | 0.5204 |
| 5 | 7.614 | 12.25 | 2 | STIR2_B | Gun-3 | 0.5204 |
| 6 | 8.219 | 11.71 | 1 | STIR1_A | Gun-2 | 0.3873 |
| 7 | 8.665 | 11.71 | 1 | STIR1_A | SAM | 0.7500 |
| 8 | 8.751 | 12.25 | 2 | STIR2_B | Gun-2 | 0.3873 |
| 9 | 9.196 | 12.25 | 2 | STIR2_B | SAM | 0.7500 |
| 10 | 9.356 | 11.71 | 1 | STIR1_A | Gun-1 | 0.2172 |
| 11 | 9.887 | 12.25 | 2 | STIR2_B | Gun-1 | 0.2172 |
| 12 | 17.83 | 25.88 | 0 | STIR1_A | Gun-6 | 0.7700 |
| 13 | 18.97 | 25.88 | 0 | STIR1_A | Gun-5 | 0.7061 |
| 14 | 19.45 | 25.88 | 0 | ILL_CIWS | CIWS | 0.7355 |
| 15 | 20.11 | 25.88 | 0 | STIR1_A | Gun-4 | 0.6246 |
| 16 | 21.24 | 25.88 | 0 | STIR1_A | Gun-3 | 0.5204 |
| 17 | 22.38 | 25.88 | 0 | STIR1_A | Gun-2 | 0.3873 |
| 18 | 22.83 | 25.88 | 0 | STIR1_A | SAM | 0.7500 |
| 19 | 23.52 | 25.88 | 0 | STIR1_A | Gun-1 | 0.2172 |

**Table 3.** Number of engagements vs. number of threats

| threats | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| *N* | 8.29 | 25.38 | 58.05 | 99.60 | 204.56 | 370.20 | 562.88 | 955.03 |
| cpu (ms) | 1.5 | 3.4 | 6.8 | 22.1 | 91.2 | 299.6 | 980.0 | 2904.6 |

### *2.2. Contingency Plan*

The Engagement List (EL) presented above is not an attack plan in itself, but an array of possible choices from which an attack plan must be constructed. Such a plan will take the form of a Contingency Plan (CP).

A CP includes all the possible outcomes of a particular set of actions, be it a chosen engagement, or an ASM hitting the ship. A CP is represented by a binary tree as there are only two possible outcomes to any given action: a threat is killed (K), or it is not killed (NK), and the ship is destroyed (D) or else survives (ND).

Figure 1 shows an example of a CP that includes engagements 0, 1 and 13. Every path shows a possible evolution of the battle. The probabilities associated with these events are given by the probability of kill attached to the respective engagement it represents or is set to 0.5 in the case of an ASM hitting the ship. Each node represents an action (fire on a threat, assess if a threat was killed, and assess the destruction of the ship) and the leaves of the tree indicate possible outcomes (the ship survived or was destroyed). The survival probability of the ship at a certain node $v$, also called the utility $u_v$ of node $v$, is recursively computed from its two children. Given $u_1$ and $u_2$, the utility of the two children nodes and $p$, the probability associated with the event at node $v$ (the probability that the ship is destroyed by an ASM in the case of a destruction assessment node or the probability that a threat is killed in the case of a kill assessment node), is given by:

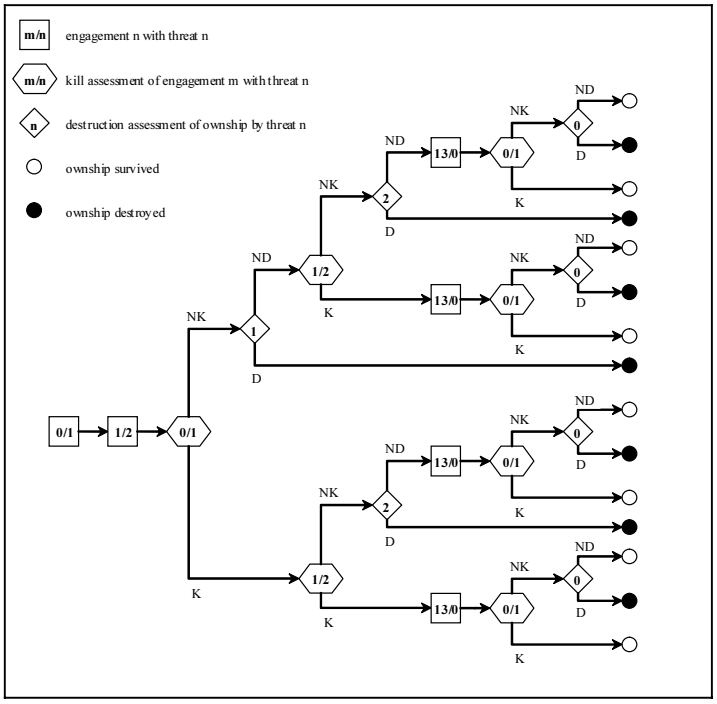



**Figure 1.** Contingency plan.

$$u_v = pu_1 + (1-p)u_2 \tag{1}$$

The utility of a leaf is set at 0 if the ship is destroyed or is set to 1 if the ship survives. The probability of survival for a given CP is determined by the utility of its root node. This is the value which must be optimized. For example, the utility of the ship hit node in the upper right corner of Figure 1 is computed as follows: the utilities of its two children, the ship destroyed (D) and not destroyed (ND) leaves, are $u_1 = 0$ and $u_2 = 1$. Here, as stated before, $p = 0.5$. Therefore $u_v = 0.5$. Moving up the CP tree, the utility of the KA node in the upper right corner is computed as follows: the utilities of its two children, the ship not destroyed (ND) and the ship hit by threat 0 nodes, are $u_1 = 1$ and $u_2 = 0.5$. The probability attached to the present action is given by the probability of kill of engagement 0, which is to say $p = 0.7355$. Therefore $u_v = 0.7355 \times 1 + 0.2645 \times 0.5 = 0.86775$. This process is recursively applied until the utility of the root node is obtained. This utility is then taken to be the probability of survival for the CP.

### *2.3. The Incompatibility Graph*

The Engagement List is constructed in order to produce a discrete solution space $S$ of all possible CPs. A CP cannot include the whole set of engagements made available by the EL because some engagements are incompatible with each other. The conditions under which two engagements are incompatible are:

– two threats, each of which can only be seen by the same STIR, cannot simultaneously use that STIR;
– there cannot be multiple STIR allocations to the same threat;
– the general shoot-look-shoot doctrine is adhered to.

These conditions are applied to the EL constructed by the BEL algorithm to generate an Incompatibility Graph (IG).

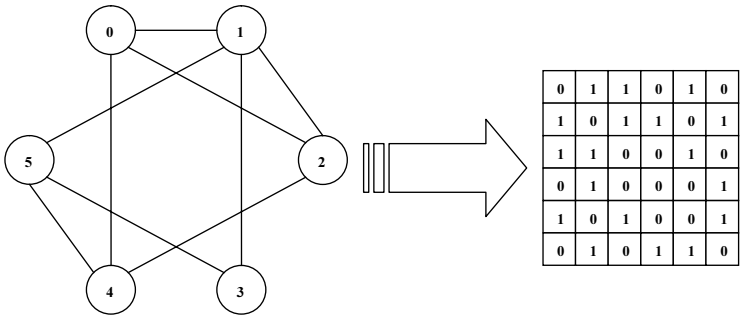



**Figure 2.** Incompatibility graph.

The IG is an undirected graph containing as many nodes as there are engagements in the EL (e.g. Figure 2). The graph can therefore be represented by a symmetrical $N \times N$ binary matrix. Two nodes are connected if the respective represented engagements are incompatible. A realizable CP cannot contain any such pairs.

The IG reduces the size of $S$ because all incompatible CPs ($S_I$) can be disregarded. Only compatible CPs ($S_C$) need to be considered. The size of $S_C$ is given by the following equation:

$$card\left(S_C\right) = 2^N - 1 - card\left(S_I\right) \tag{2}$$

because:

$$S = S_I \cup S_C \quad \text{and} \quad S_I \cap S_C = \phi \tag{3}$$

Here $N$ is the number of engagements in the EL. When the incompatibility graph is fully connected, the number of possible CPs is equal to $N$ because every possible CP is a singleton containing only one engagement. A fully connected graph is an extreme case; in the aforementioned randomly generated scenarios the graph is on average 30% to 50% connected (see Table 4). Even though the solution space is drastically reduced, it is still exponential in respect to $N$, which implies a double exponential with respect to the number of threats.

**Table 4.** Incompatibility graph connectivity

| threats | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $N^2$ | 76 | 779 | 4274 | 12803 | 53477 | 179723 | 411870 | 1079759 |
| arcs | 40 | 343 | 1623 | 5634 | 20313 | 59462 | 118131 | 339886 |
| percent | 53% | 44% | 38% | 44% | 38% | 33% | 29% | 31% |

However, what is gained in reducing the size of $S$ is lost when it comes to complexity. It can be verified that the resulting solution space contains a large number of local optima (defined using the metric described in 6.3).

## 3. The RM Problem Statement

The next section presents a rigorous mathematical formulation of the problem for the purpose of optimization.

### *3.1. The Cost Function*

The cost function is defined as the probability of survival for a given CP, i.e.:

$$S \triangleq \{0,1\}^N \qquad \begin{array}{rccc} f: & S & \to & [0,1] \\ & x & \mapsto & u_x \end{array} \tag{4}$$

The solution space $S$ of possible CPs is represented as the discrete space of binary strings containing $N$ bits. In the binary representation of a CP, $x_i$ =1 implies that engagement $i$ is part of that CP. Here $u_x$ is the utility of the CP $x$. From the recursive formula (1) when applied to the CP tree containing the entire EL, the following cost function is obtained, in which the active engagements (i.e. the ones that correspond to $x_i$ = 1), yield the correct formula to obtain the utility of any CP:

$$\begin{aligned} & f(x_1,...,x_N) = \sum_I \alpha_I X_I \quad I \triangleq \{i_1,...,i_k\} \quad k = 0,...,N \\ & \qquad\qquad\qquad\qquad i_n \in \{1,...,N\} \quad n \neq m \Rightarrow i_n \neq i_m \\ & X_I = x_{i_1} \cdot \ldots \cdot x_{i_k} \\ & X_\phi \triangleq 1 \end{aligned} \tag{5}$$

As in the sum of every possible monomial in $x_i$ multiplied by an $\alpha$ which depends on the probabilities assigned to the engagements represented in the multi-index $I$. Here is an example of what a part of that sum could look like:

$$f\left(x_1,\ldots,x_7\right) = \ldots + 0.121 x_1 x_2 x_4 + 0.091 x_1 x_2 x_4 x_5 + \ldots \tag{6}$$

Here is an example to clarify. Consider the following scenario:

**Table 5.** Threat specifications

| threat number | ship hit time | probability of destruction |
|---|---|---|
| 1 | 3 | $\beta_1$ |
| 2 | 6 | $\beta_2$ |

**Table 6.** Engagement list

| engagement number | KA time | Threat | probability of kill |
|---|---|---|---|
| 1 | 1 | 1 | $\alpha_1$ |
| 2 | 2 | 1 | $\alpha_2$ |
| 3 | 4 | 2 | $\alpha_3$ |
| 4 | 5 | 2 | $\alpha_4$ |

The tree to consider, containing 6 events (fire time need not be considered when computing the utility of a CP), is presented in Figure 3.

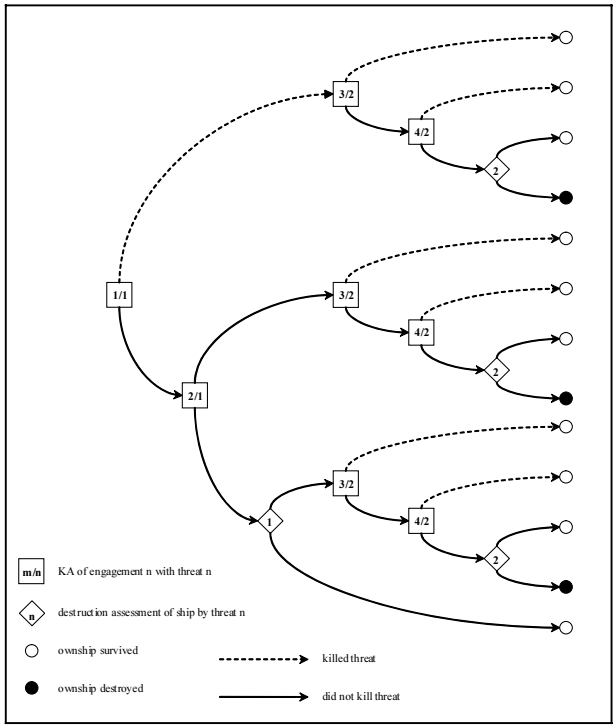



**Figure 3.** The scenario tree.

Using the recursive formula (1) given in 2.2 yields the following representation of the cost function:

$$
\begin{aligned}
&f(x_1,x_2,x_3,x_4) = \alpha_1\alpha_2\alpha_3\alpha_4(1-\beta_1)(1-\beta_2)x_1x_2x_3x_4 \\
&-\alpha_1\alpha_2\alpha_3(1-\beta_1)(1-\beta_2)x_1x_2x_3 - \alpha_1\alpha_2\alpha_4(1-\beta_1)(1-\beta_2)x_1x_2x_4 \\
&-\alpha_1\alpha_3\alpha_4(1-\beta_1)(1-\beta_2)x_1x_3x_4 - \alpha_2\alpha_3\alpha_4(1-\beta_1)(1-\beta_2)x_2x_3x_4 \\
&-\alpha_1\alpha_2(1-\beta_1)\beta_2x_1x_2 + \alpha_1\alpha_3(1-\beta_1)(1-\beta_2)x_1x_3 \\
&+\alpha_1\alpha_4(1-\beta_1)(1-\beta_2)x_1x_4 + \alpha_2\alpha_3(1-\beta_1)(1-\beta_2)x_2x_3 \\
&+\alpha_2\alpha_4(1-\beta_1)(1-\beta_2)x_2x_4 - \alpha_3\alpha_4\beta_1(1-\beta_2)x_3x_4 \\
&+\alpha_1(1-\beta_1)\beta_2x_1 + \alpha_2(1-\beta_1)\beta_2x_2 + \alpha_3\beta_1(1-\beta_2)x_3 + \alpha_4\beta_1(1-\beta_2)x_4
\end{aligned}
\tag{7}
$$

The CP containing engagements 1 and 3 would then give a probability of survival equal to $f(1,0,1,0)$. The size of this function increases exponentially with respect to the number of engagements in the EL. It could be simplified somewhat by removing the terms containing incompatible engagements, but that would not change its exponential nature.

### *3.2. The RM Problem as a Maximization Problem*

The problem statement can now be presented as a maximization problem. $A=(a_{ij})$ is taken to be the matrix representation of the IG.

$$
\max\{f(x_1,...,x_N) \mid (x_1,...,x_N) \in X\} \tag{8}
$$

where the constraint set $X$ is:

$$
\begin{aligned}
&X \triangleq \{(x_1,\ldots,x_N) \in \{0,1\}^N \mid g_i(x_1,\ldots,x_N) = 0, i = 1,\ldots,N\} \\
&g_i(x_1,\ldots,x_N) \triangleq x_i(x_1a_{i1} + x_1a_{i1} + \cdots + x_Na_{iN})
\end{aligned}
\tag{9}
$$

Section 4 will address the way this maximization problem should be approached. Before presenting a possible solution, the complexity of this problem should be made clear. The problem will be redefined as a decision problem where there are but two possible answers, yes or no. This is the class of problems studied in complexity theory.

### *3.3. NP-Completeness of the RM Problem*

The RM problem is defined as follows:

$$
\{\langle IG,k\rangle \mid \exists s \text{ a set of } k \text{ nodes with } (a,b) \in IG \Rightarrow a \notin s \text{ or } b \notin s\} \tag{10}
$$

Where *IG* is the incompatibility graph. The problem is therefore to decide whether or not there exists an independent set of $k$ nodes, i.e. to find a compatible CP containing $k$ engagements. This problem (known as the Independent Set Problem (ISP)) has been shown to be NP-Complete, as it is possible to reduce it to 3-SAT (3-Satisfiability) [2]. This is a simplification of the actual problem in which the independent set that is associated to the CP with the maximum utility must be found. Thus, the actual problem

does not only require that the ISP be solved but also that the cost function be maximized as well. If the cost function were trivial it would not add to the complexity of the problem. However, computing the utility of a CP takes exponential time relative to the number of engagements in the CP. The next section explains how to tackle this problem using a simple heuristic and two metaheuristic methods (TABU search and a Genetic Algorithm).

## 4. A Proposed Solution

This section presents the methods used and studied by the authors of [1]. The first method is a simple, albeit quite effective, construction heuristic that uses knowledge about the warfare environment and common sense. The second method is a metaheuristic in which a solution is improved through an iterative neighborhood search. The construction heuristic is used to initialize the metaheuristic making it a complete system to solve the RM problem.

### *4.1. The Construction Heuristic*

Both metaheuristic methods studied here (TABU search and Genetic Algorithms) need one or several initial solutions. In most cases it is more useful to start with a relatively good solution as opposed to a randomly generated one. Knowing something about the problem and the environment in which it evolves is helpful; some principles are listed below:

- the closest threats are engaged first;
- a SAM has priority over the gun to engage a threat;
- the CIWS engages whenever possible;
- the number of re-engagements of a threat is maximized.

These principles suggest an algorithm. The threats are ranked based on their distance from the ship; the closest threat has rank 1. The algorithm is initialized with $r = 1$ and an empty CP. While the solution is not saturated, as in there exist engagements in the EL that are compatible with the present state of the CP, scan the EL until one of the following is found (in order): a SAM engages a threat with rank $r$, a Gun engages a threat with rank $r$, the CIWS engages a threat with rank $r$. If an engagement is found then add it to the CP, otherwise increment $r$. This is done until all threats have been processed, in other words while $r$ is smaller or equal than the number of threats [1]. This algorithm produces effective solutions to the RM problem that could be used without any additional computations. As will be shown, however, the solutions produced by this algorithm are not globally optimal. In subsequent sections this algorithm will be referred to as Init Solution (IS).

### *4.2. TABU Search*

TABU search is basically an iterative neighborhood search method. A modification of the solution is called a move. Sometimes a detrimental move will be allowed in order to facilitate the escape from a bad local optima, as opposed to a pure local search approach. To avoid cycling, a list of previously visited locations is kept in memory. It is then forbidden (“tabu”) to go back to these solutions for a certain number of iterations.

The algorithm can be summarized as follows. Generate an initial solution using the IS algorithm. While the stopping criterion (a maximum number of iterations without improvement) is not met, generate a neighborhood of the current solution through non tabu moves that add an engagement, and select the best solution in this neighborhood. If this solution is better than the current solution it becomes the new current solution. If it is not, then an engagement is removed. The removed engagement is then made tabu for a number of iterations which is randomly chosen between two preset bounds. When the stopping criterion is met the best solution reached while involved in the search is then returned.

Since removing or adding an engagement results in a far from trivial change in the probability of survival of a CP, it would take too much time to compute an exact value for every possible move when executing a neighborhood search. Thus, some approximations were used to compute the impact of adding to, or removing, an engagement from a CP. These will not be presented here, for further explanations about the specifics of the TABU search method used to solve the RM problem see [1].

## 5. Genetic Operators to Solve the RM Problem

To implement a Genetic Algorithm (GA) in order to solve a problem, it is imperative to find an appropriate method for coding a solution as well as an efficient method of evaluating such coded solutions. A GA's performance heavily relies on the genotype/phenotype relation that is chosen. An evaluation procedure is usually called a Fitness Function (an allusion to the idiom "survival of the fittest"). Thereafter a method in which two solutions can be successfully combined to form a new solution must be found. This Crossover method must have a certain degree of heredity (i.e. the child solution must have a fitness that is in the neighborhood of its parent's fitness with a high enough probability) or else the entire process is aimless and random, as there is no steady climb towards any optimum.

It is also very useful to be able to perturb any given solution with what is usually called a mutation operator. Such an operator can be random (usually with a very low probability of improving a solution) or heuristically driven (slower but guaranteed to improve almost any solution). A GA with a heuristically driven mutation operator is sometimes called a Memetic Algorithm [3].

The phenotype/genotype relation, a method for evaluating a possible solution, as well as the crossover and mutation operators to solve the RM problem are the objects of this section.

### *5.1. The Genetic Code*

The genetic code of a possible solution, i.e. a Contingency Plan, is quite simple. The genotype of a CP is an array of integers ranging from 1 to $N$ that represent the actual engagements contained therein. There is no need to manipulate large tree structures as the genotype chosen (the array of integers) contains all the information needed to reconstruct the phenotype (the CP tree).

The difference of this coding scheme with the usual methods used in GAs is that the length of the code is dynamic. Of course, in keeping with tradition, the genotype could have been represented as an array of $N$ bits, producing a clear isomorphism with the solution space definition given above.

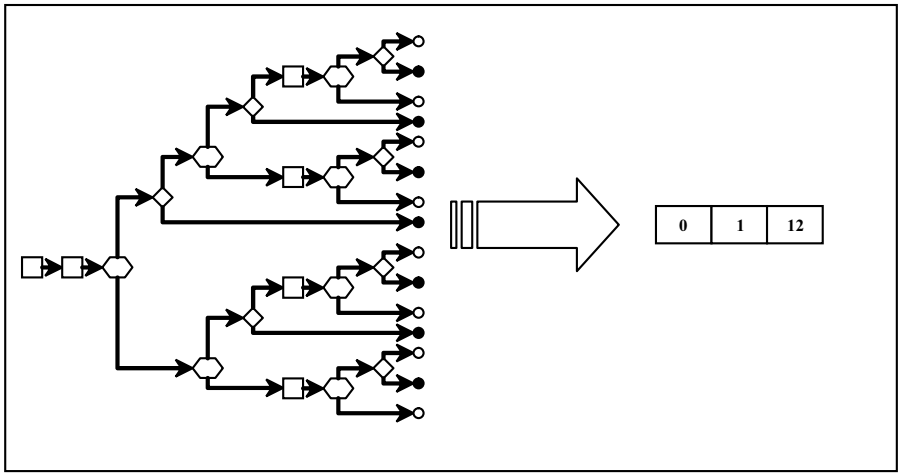



**Figure 4.** The Code (genotype) of a CP.

GAs were first implemented using such arrays of bits because they were simple and efficient to manipulate. Both approaches are nowadays almost equivalent when it comes to CPU resources and memory. The binary approach necessitates less memory management and has been adopted in the computer implementation of the algorithm, but for the sake of simplicity the genetic operators are presented here using the former approach. The choice of genotype presented here is in accordance with both the principle of meaningful building blocks and the principle of minimal alphabets [4]. It is hard to imagine a simpler and more meaningful way of expressing a CP. It is important to note that the time required for the genotype to phenotype transformation is negligible when compared to the time required to compute the Fitness (utility) of the phenotype (CP).

Table 7 illustrates the average size (over the 100 aforementioned random scenarios per number of threats) of the best solution found (usually representative of the maximum independent set size). The length of the resulting code in the case of the IS construction heuristic and TABU search are also presented.

**Table 7.** Size of Best Solution

| Threats | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| GA | 4.44 | 8.22 | 11.10 | 13.95 | 16.42 | 18.77 | 22.69 | 25.41 |
| IS | 4.44 | 8.23 | 11.23 | 14.09 | 16.73 | 19.01 | 22.97 | 25.62 |
| TABU | 4.44 | 8.03 | 10.91 | 13.25 | 15.50 | 18.08 | 22.69 | 25.36 |

### *5.2. The Fitness Function*

The fitness of a CP is naturally its associated utility. The Fitness Function (FF) is therefore the same as the cost function described above in the rigorous exposition of the problem. Unfortunately, obtaining an exhaustive expression of this function would be prohibitive beyond trivial scenarios in which the EL contains but a few engagements. For example, a scenario consisting of 5 threats and an EL of size 300 would necessitate the evaluation of $2^{305}$ terms. A more practical approach must be developed if we are to implement a sensible GA scheme for the RM problem. Since every compatible CP can

only contain but a small fraction of the engagements in the EL, the FF builds, every time it is called, a reduced cost function. Only the terms that contain the CP's engagements are created using a recursive function that works along the principle presented in section 2 (how to compute the utility of a CP using its tree structure). Although this computation takes place in exponential time with respect to the size of the CP, it proved to be very efficient in all but the most extreme cases.

*5.2.1. The Fast Fitness Function*

In extreme scenarios, where it would be prohibitively long to compute the exact utility of a CP, an approximation of the utility was used. The Fast Fitness Function (FFF) is the following linear approximation of the cost function:

$$FFF(CP) = \sum_{eng \in CP} eng(p_k) \tag{11}$$

where $p_k$ is the probability of kill associated with engagement $k$. This approximation takes linear time with respect to the size of the CP. It involves a certain degree of error in the ordering and selection of solutions but still yields very good results when used only for extreme cases. Such extreme cases usually have an IG with a low connectivity factor and therefore it is relatively easy to generate close to optimal solutions. Indeed, the solutions found using the FFF were almost as good as those found using the slower FF (as the Hybrid column demonstrates in Table 9). Table 8 shows the percentage of scenarios in which the FFF was used in the Hybrid simulations. Table 9 shows the average time as well as the utility over 100 random scenarios (for numbers of threats ranging from 1 to 8).

**Table 8.** Percentage of scenarios in which the FFF was used

| threats | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| FFF | 0% | 0% | 0% | 4% | 6% | 12% | 29% | 42% |

**Table 9.** Comparisons of FF and FFF

| | CPU time (ms) | | | Fitness | | |
|---|---|---|---|---|---|---|
| threats | FF | FFF | hybrid | FF | FFF | Hybrid |
| 1 | 46.4 | 52.5 | 59.0 | 0.9832 | 0.9832 | 0.9832 |
| 2 | 87.3 | 61.4 | 92.2 | 0.9513 | 0.9401 | 0.9514 |
| 3 | 236.8 | 110.2 | 216.1 | 0.9006 | 0.8838 | 0.9002 |
| 4 | 814.7 | 150.4 | 466.2 | 0.8233 | 0.7663 | 0.8224 |
| 5 | 2670.4 | 254.1 | 1527.2 | 0.7448 | 0.6952 | 0.7435 |
| 6 | 4428.7 | 344.5 | 1775.4 | 0.6536 | 0.5971 | 0.6477 |
| 7 | 33160.8 | 404.5 | 3243.5 | 0.5958 | 0.5385 | 0.5724 |
| 8 | 68554.0 | 552.5 | 3151.4 | 0.4845 | 0.4490 | 0.4569 |

Although very fast, the simulations that only used the FFF gave the worst results. The simulations that only used the normal FF gave the best utilities, but took longer. The computation time averaged around a minute when 8 threats were involved, and 30 seconds when 7 were involved. This is incompatible with this particular application because the ASMs would have hit the ship by the time a solution was arrived at. The Hybrid simulation delivers what seems to be the best trade-off between precision in

solution and speed of computation. The Hybrid system is explained in more detail in section 7.

### 5.3. Initialization: The Random Code Generator

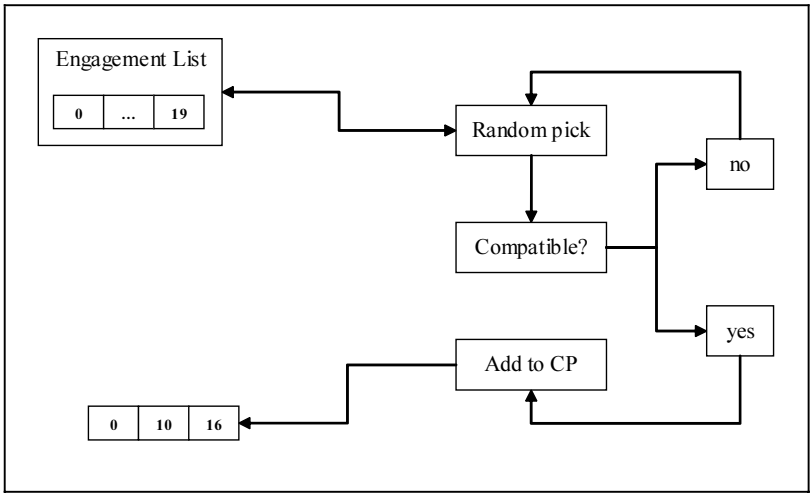



**Figure 5.** Random Code Generator.

To initialize the GA, a simple and effective way of generating random CPs to form an initial population with enough variety is needed. This was done using the Random Code Generator (RCG) shown in Figure 5. Simply put, the RCG randomly picks an engagement from the EL and adds it to the solution if it is compatible with all previously chosen engagements. The procedure is repeated for a fixed number of iterations, which is chosen according to the size of the EL. The same number of iterations is used in the Mutation and Crossover operators, both of which are described below. There is no need to verify whether the CPs so generated are saturated or not. A saturation check would only slow down the system, and more importantly would reduce the randomness of the generated CPs. These randomly generated solutions are expected to form only partial CPs which the GA will thereafter complete.

Table 10 shows the average performance of the RCG over 30 random scenarios per number of threats. In each scenario 100 CPs were randomly generated. The performance of the IS construction heuristic is also presented.

**Table 10.** Performance of the Random Code Generator

| threats | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Avg | 0.9404 | 0.8268 | 0.6418 | 0.4320 | 0.3553 | 0.2382 | 0.1863 | 0.1357 |
| Min | 0.7439 | 0.5588 | 0.4167 | 0.2404 | 0.1896 | 0.1077 | 0.0834 | 0.0605 |
| Max | 0.9811 | 0.9497 | 0.8728 | 0.6840 | 0.6159 | 0.4375 | 0.3540 | 0.2736 |
| IS | 0.9827 | 0.9319 | 0.8581 | 0.7282 | 0.6300 | 0.5131 | 0.4831 | 0.3886 |

### 5.4. Crossover

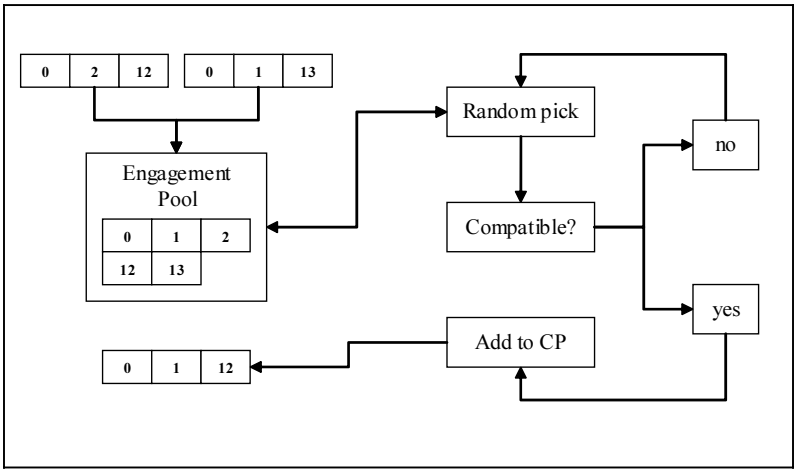



**Figure 6.** Crossover of two CPs.

The crossover of two CPs can be seen as an operator that goes from the Cartesian product of $S$ to $S$:

$$\begin{aligned} C: \quad S \times S \quad &\to \quad S \\ (x, y) \quad &\mapsto \quad C(x, y) \end{aligned} \tag{12}$$

The crossover of two CPs is shown in Figure 6. First, an engagement pool containing all the engagements of both parent CPs is built, then a random engagement from this pool is chosen and added to the child CP. Every time an engagement is picked, a compatibility check is performed to see if the newly picked engagement is compatible with the previously chosen engagements. The procedure is repeated for the same number of iterations mentioned above.

The crossover operator possesses good heredity because the engagements present in the engagement pool tend to be, on average, more compatible than those in the EL. The following measures of heredity were devised so that this could be verified empirically. Good Heredity and Elite Heredity are defined as follows:

$$\begin{aligned} &\text{Good Heredity} \triangleq P[f(C(x, y)) \geq f(x) \text{ or } f(C(x, y)) \geq f(y)] \\ &\text{Elite Heredity} \triangleq P[f(C(x, y)) > f(x) \text{ and } f(C(x, y)) > f(y)] \end{aligned} \tag{13}$$

i.e. Good Heredity is the probability that the resulting child is at least as good as one of its parents and Elite Heredity is the probability that a child is strictly better than both its parents. Table 11 shows the results over 30 randomly generated scenarios for each number of threats.

**Table 11.** Heredity Measures for RM Crossover

| threats | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| good | 78.00 | 68.01 | 58.50 | 52.72 | 54.41 | 59.89 | 58.32 | 65.60 |
| elite | 6.16 | 21.56 | 13.78 | 6.96 | 6.98 | 8.24 | 8.00 | 9.07 |

*5.5. Mutation*

To mutate a CP, an engagement is randomly picked from the EL and added to the CP if it is compatible. The procedure is repeated for the same number of iterations as for the RCG and Crossover operators.

$$\begin{array}{rccc} M: & S & \to & S \\ & x & \mapsto & M(x) \end{array} \tag{14}$$

The mutation operator here is not heuristically driven, but its success ensures that a better solution is produced because adding an engagement can only improve a solution. If no engagements were added, an engagement is removed as the solution is judged to be saturated. If an engagement is removed, two copies of the CP are produced: one with the removed engagement and one without, because the saturated solution might be the optimal solution. The process is illustrated in Figure 7.

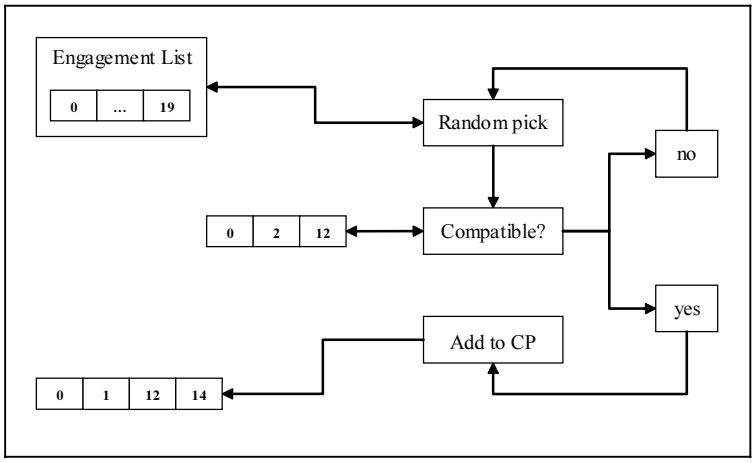



**Figure 7.** Mutation of a CP.

Table 12 shows the ratio of successful to unsuccessful mutations for 100 random scenarios per number of threats. Except for easily saturated solutions (when only one threat is involved) the operator can with high probability add an engagement. The high heredity of the Crossover operator coupled with the high success rate of the Mutation operator make for an aggressive and effective search of the solution space.

**Table 12.** Mutation Success

| threats | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| added | 21.76 | 220.01 | 477.78 | 577.44 | 651.54 | 625.10 | 651.80 | 682.18 |
| removed | 282.01 | 110.80 | 117.62 | 122.82 | 118.33 | 106.36 | 114.57 | 141.89 |
| success | 7.2% | 66.5% | 80.2% | 82.5% | 84.6% | 85.5% | 85.1% | 82.3% |

## 6. An Any Time Solution System

"Genetic Algorithms require a pool of solutions, where each solution evolves concurrently through information exchange. Although attractive, these ideas also imply large computation times, which are incompatible with our application." [1]

The operators described in section 5 can be used to build a GA. Traditionally a GA has the following data flow. First, an initial population is created using the RCG. The size of the population is preset. Then a means of selection must be chosen. For example the best half could be chosen for reproduction. The surviving half is then randomly coupled twice so that they may produce as many offspring as needed to maintain the population size. The procedure is then repeated until stopped. A number of other selection criteria could be used, but the basic mechanism is always the same. One generation at a time is processed, and one best solution is given at every generation. This is hardly compatible with the current application.

In the context of the current problem, a user is likely to require a solution at any possible time. The genetic operators presented above could be used to build an anytime solution system. With data-driven agent programming such a system was built. It is the subject of the following section. The resulting GA is part of a class of GAs sometimes called Steady State Genetic Algorithms. It is not bound by any iterative process but instead follows the dynamics of a natural population reproducing in a semi-random way.

### *6.1. Adjusting the Flow of Information*

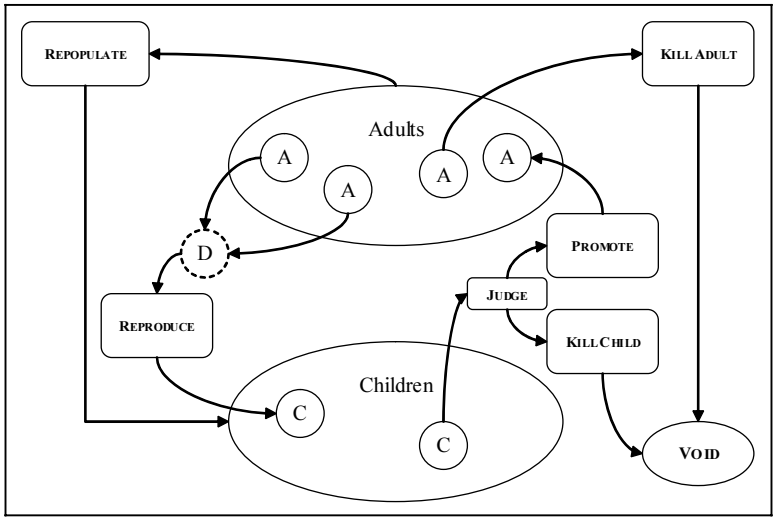



**Figure 8.** Steady State GA Data Flow.

The principles of reactive data-driven agent programming should be explained before the data-flow chart can be understood. The system used (*Cortex*) was developed by Lockheed Martin Canada (LMC). It comprises of a blackboard where the data is stored and a controller which directs the data to the appropriate agents. Each agent is triggered by a certain type of data. When data is put on the Blackboard, the agents that are triggered by it place a demand which is then inserted in an execution queue by the controller. If several agents have made a demand on a single data instance simultaneously then a context function is used to decide which of these agents will take precedence. An agent which acts on a set of data at one time is called a meta-agent and is triggered not by a single instance of data but by the properties of a set of data.

All of these terms and principles will be made clearer once put into the context of an example, namely a Genetic Algorithm using the aforementioned operators to solve the RM problem (Figure 8). The resulting GA is not specialized to solve the RM problem, but is an alternative method of implementing a GA. The added bonus is that the solution space is no longer explored on a generation-by-generation basis, and a near optimal solution is found without having to waste precious CPU time computing the fitness of too many sub-optimal solutions. For example, it would be possible to find a generation 10 solution without having to go through the complete manipulation and evaluation of the precedent 9 generations.

### *6.1.1. The Data*

There are three types of data on the blackboard, Adults (A), Children (C) and Demands (D). Adults and Children contain the same type of information, the genotype of a solution and its fitness value. They belong to the animal class. They also contain a tag, an integer value that is incremented every time a new animal is created, and a generation number. The generation of an animal is determined by the generation of its parents. The difference between Adults and Children is that Adults have the ability to reproduce and Children do not. Children have to be judged worthy of reproduction before they are promoted to the Adult class. When two Adults are on the blackboard simultaneously a Demand is automatically spawned. The Demand data type is virtual, it contains no data; it is but a flag. Virtual data is part of the Cortex architecture. In the Genetic Algorithm architecture the Demand virtual data is essentially used to signal the presence of two Adults.

### *6.1.2. The Simple Agents*

There are four simple agents (agents triggered by a single instance of a data), Reproduce, Promote, Kill Child and Kill Adult. The Reproduce agent is triggered by the presence of a Demand on the blackboard. When activated, it proceeds to create two Children from the genetic code of the two Adults who instigated the Demand. It applies the crossover operator $C$ to the two parents and then the mutation operator $M$ to the resulting child.

$$child = M\left(C\left(parent_1, parent_2\right)\right) \tag{15}$$

The Promote agent is triggered when a Child is present on the blackboard and has been judged to have a high enough fitness. Its only function is to take a Child and promote him to the Adult class. On the other hand, if a Child is not fit enough (see 6.2.1 for further explanation as to how selection is accomplished) it triggers the Kill Child agent which then proceeds to delete it. The Kill Adult agent is triggered when an unfit Adult is present on the blackboard, using the same criterion as for Children, the unfit Adult is deleted.

### *6.1.3. The Flow*

To initialize the algorithm, a population of children is created using the RCG. The Children are, one by one, either promoted or deleted. As pairs of Adults appear on the blackboard, so do Demands. The Reproduce agent is triggered and new Children are created. These are then judged fit or unfit and are dealt with accordingly by the Promote and Kill Child agents. The Adult population becomes progressively more fit as a result. The Kill Adult agent then intervenes to rid the Adult population of older, less fit individuals. Each agent execution counts as one time unit. The algorithm then proceeds until a prescribed time limit is passed or the user stops it. A best solution is available at any time. The longer the algorithm is run the better the solution obtained. Of course, there is a point of diminishing return as it takes on average progressively more time to come up with a new best solution. The question as to when to stop the algorithm is left open and should be the object of further research.

## *6.2. Parameters*

A set of control parameters was used to control the behavior of the algorithm. These are not all presented here as they would obscure the main line of reasoning and are present for purely technical reasons. The parameters of interest here are the ones related to selection and population control. The size of the population tends to continuously grow if left unattended.

### *6.2.1. The Threshold*

Every GA implementation must have a selection scheme, a means of deciding who will survive and who will die. In a traditional GA, selection is applied to a whole population in a single context, in between two generations. Here selection is applied to a single animal at a time and the context is in constant flux. Each time an agent is executed the nature of the population is changed. The fate of an Animal is decided upon a measure depending on the population's fitness, called the threshold. There are many different ways to place a threshold on a population. Several were tried and the one below, a weighted median, was selected.

$$Threshold = \alpha(\max(fitness) - \min(fitness)) + \min(fitness) \quad \alpha \in ]0,1[ \qquad (16)$$

Numerous simulations were performed in order to find the optimal value of $\alpha$, for the RM problem a value of 0.8 was near optimal (for the TSP problem a value closer to 0.5 was near optimal). Other thresholds were tested. Some were a function of the average fitness of the population; these were inefficient because they depended on the distribution of fitness among the population and as a result the elites tended to take over

the population too quickly, which led to a premature convergence of the GA. Some even took into account the size of the population among their parameters in an effort to simulate the building selection pressure as a population grows in size. This also led to premature convergence, so other parameters were formulated in order to control the size of the population.

### *6.2.2. Population Control*

The size of the population was controlled by means of two parameters: the population cap and the minimum population. These control the activation status of the Reproduce agent. When the size of the population reaches the population cap the Reproduce agent deactivates itself. The algorithm enters a purely selective phase. When the size of the population drops below the minimum population the Reproduce agent is reactivated by either of the Kill agents, and a Repopulation flag is set on the blackboard (see 6.5 for further explanation about the Repopulate agent). Figure 9 shows the size of the population against time in an example run of the algorithm.

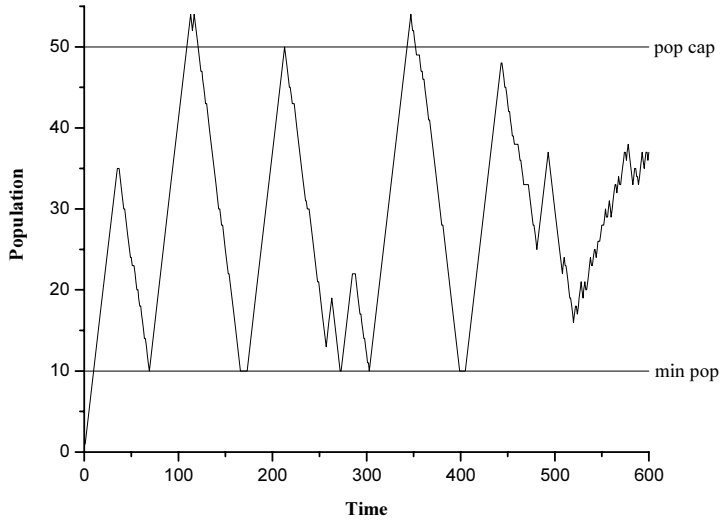



**Figure 9.** Population size.

### *6.3. Entropy of a Population*

Entropy, as presented here, is a useful measure of the variety in a population. In most problems it is often only necessary to look at the variance of the fitness in a population to assess its degree of variety or solution space coverage. For the RM problem, in which the solution space contains numerous local optima of similar and often equal value, a deeper measure is in order. This measure shall be called the entropy of a population for obvious reasons.

Before entropy is defined, a metric on the solution space $S=\{0, 1\}^N$ must be defined. The distance between two elements of $S$ is defined as follows:

$$\begin{aligned} d: \quad S \times S &\rightarrow \{0,1,\ldots,N\} \\ (x,y) &\mapsto \sum_{i=1}^{N} |x_i - y_i| \end{aligned} \tag{17}$$

This metric is known as the Hamming distance, essentially the sum of all the different respective bits of $x$ and $y$. With this metric in mind, the entropy of a population $P$ can now be defined. Let $P^* \subset P \times P$ be defined as follows:

$$\begin{aligned} &\text{if} \quad P = \{x_1,\ldots,x_n\} \\ &\text{then} \quad P^* = \{(x_i,x_j) \,|\, i < j \text{ and } j \leq n\} \end{aligned} \tag{18}$$

Now, the entropy of $P$ is defined as:

$$Entropy[P] = \frac{1}{|P^*|} \sum_{(x,y)\in P^*} d(x,y) \tag{19}$$

In essence, it is the average distance between every pair of solutions in the population. As a rule of thumb, it is wise to keep the entropy of a population higher than a certain level or else the algorithm will have the tendency to prematurely converge as the population loses its diversity.

### *6.4. Premature Convergence*

A steady-state GA can put too much selection pressure on its population, so that it loses its diversity as the genes of the elite spread too quickly [5]. Looking at the population size graph of Figure 9 one might think that from the ten or so survivors left after an extinction period, the resulting population, after multiple crossovers amongst a very uniform set of solutions, would occupy only a small region of the solution space. This is in fact true, and an example of it is shown in Figure 10 where the size, the entropy, the average fitness and maximum fitness of the population are superposed (the scenario chosen had 5 threats and the size of the EL was 199).

At time 201 a solution of fitness 0.6515 is found and thereafter, between time 201 and 240, the entropy of the population drops as the few elites and their descendents begin to dominate the population. Thereafter the population continuously grows because most of its members are above the selection threshold, the algorithm subsequently converges at time 398 (convergence is here defined when max fitness = min fitness). The same results were found, scenario after scenario, and not only for the RM problem, but also for the Traveling Salesman Problem which was used to benchmark and test the GA.

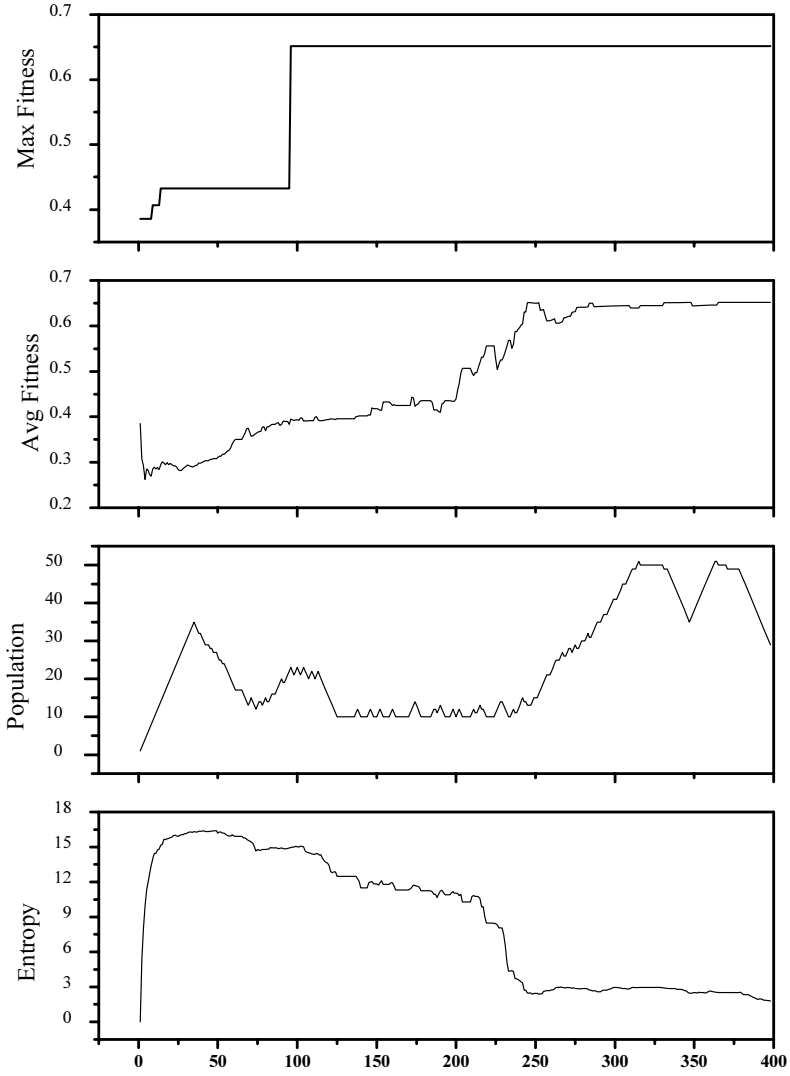



**Figure 10.** Scenario progression.

### *6.5. The Repopulate Agent*

To prevent premature convergence a new meta-operator was introduced, which took the form of the Repopulate meta-agent in the present GA implementation. The Agent operates on the entire population in contrast with the data-driven Reproduce agent. Its triggering is sensitive to the size of the population. The activation of this agent is the responsibility of the Kill Child and Kill Adult agents. When the population size falls below a certain threshold, the Kill Agent responsible for the crossing of this threshold sets a flag on the Blackboard to activate the Repopulate Agent. What it does is for every survivor a ghost is summoned using the RCG, the survivor then mates with it to produce one or two children (figure 11), the ghost is then disposed of, as in, it is not

admitted into the population. Since no selection must be applied to the summoned ghosts, there is no need to compute their fitness and therefore the repopulation is not costly in terms of CPU resources. This operator spreads the gene pool across the solution space in an efficient and often fruitful manner, as it often results in a series of new best solutions.

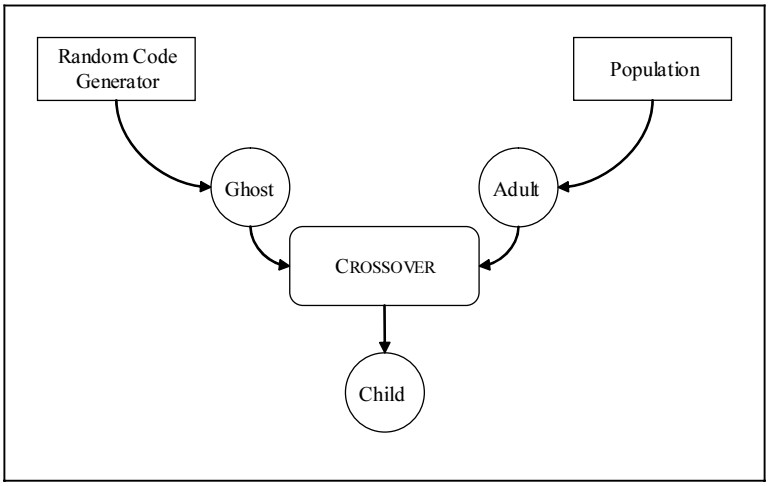



**Figure 11.** The Repopulate Agent.

The repopulate size parameter (number of ghosts per survivor), was found to be near optimal at 2 for the RM problem. Here is, for the same scenario described in section 6.4, the resulting behavior of the GA when the Repopulate agent was activated (figure 12).

Here, the entropy was kept high throughout the whole scenario and no convergence was detected. The aforementioned convergence effect, where the population grows continuously because all its surviving members are above the selection threshold, was avoided because the surviving elites were not allowed to reproduce amongst themselves but, instead, with a random ghost population. Selection was thus applied to a more diverse population. Between time 300 and 400 a series of new best solutions were found. The majority of those new solutions were the result of a crossing between one of the surviving elites and a ghost. The last best solution was found at time 398 and had a fitness of 0.7567.

Table 13 shows the average number of repopulations per number of threats. This number increases with the size of the scenario, which is due to the severe selection pressure encountered when the complexity of the scenario is high.

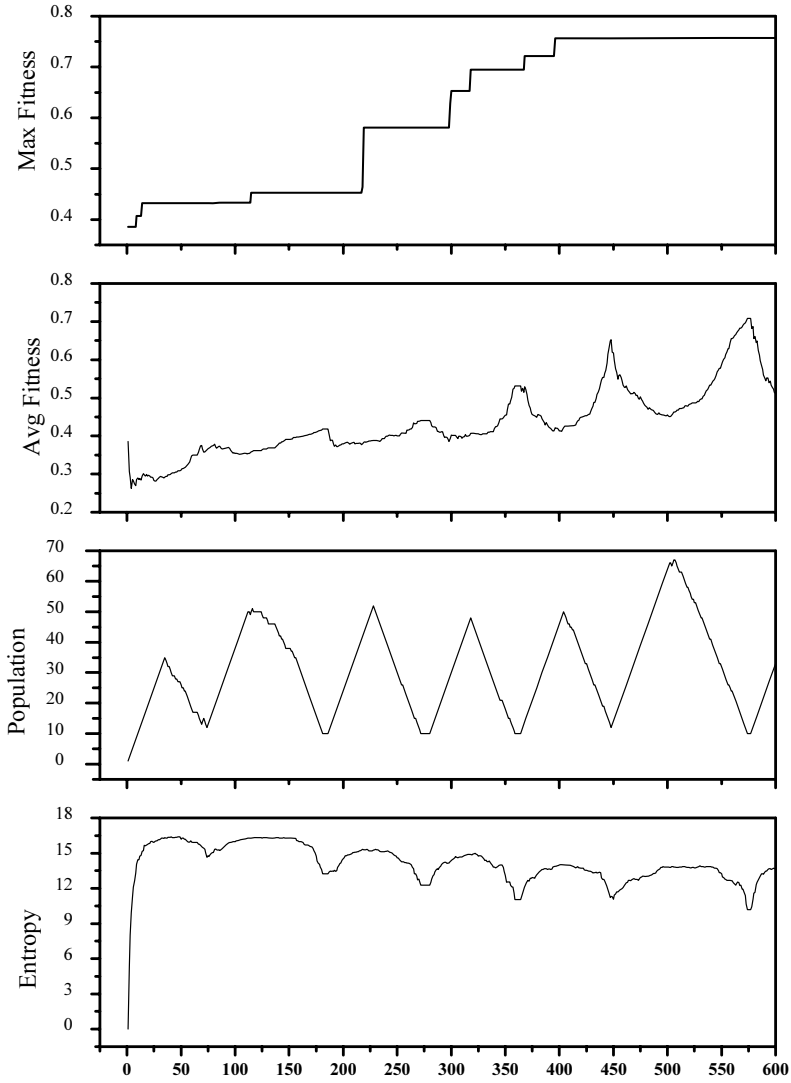



**Figure 12.** Scenario progression with Repopulation.

**Table 13.** Number of Repopulations

| threats | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| repopulations | 2.22 | 2.70 | 4.97 | 6.76 | 9.16 | 9.91 | 12.39 | 15.66 |

## 7. Comparisons and Performance

### *7.1. The Simulation*

A set of 800 randomly generated scenarios were created, 100 for each number of threats (1 to 8). The threats were randomly positioned from 5 to 40 km away from the

ship and at any azimuthal angle. This is basically the same testing procedure as was used in [1].

### 7.2. Results

The first test was performed with the normal recursive FF. The Random Code Generator was used to build an initial population of 35 solutions; the population cap was set at 50 with a minimum population of 10, and a threshold $\alpha$ of 0.8. Tables 14 and 15 show, respectively, the CPU time in milliseconds and the utility of the best solution found. The column containing the first test results is labeled GA. The column labeled IS shows the performance of the construction heuristic, and the column labeled TABU shows the TABU search performance when initialized with the output of the construction heuristic. The pure GA outperforms both in terms of utility, but for a number of threats higher than 6, takes considerably longer to simulate, taking an average of 68 sec for 8 threats. This is interesting from a theoretical point of view but incompatible with the current application.

The second test was identical to the first in all but one aspect: the output from the construction heuristic was injected into the initial population. This was done in the hope that a highly fit individual present in the initial population would act as a guide for the search towards a better optimum. The results, shown in the column labeled GAIS, do in fact substantiate that conjecture. The GAIS algorithm gave the best utilities of all. Since the recursive FF was used, the CPU times were still unacceptable.

The third test was identical to the second except that the linear FFF was used for all of the simulations. This was done to evaluate the degree of error introduced into the selection scheme when the FFF was used to approximate the utility of a solution. The results are shown in the column labeled $GAF^3$ (the utility presented is not an approximation). The CPU times demonstrate that the algorithm is significantly faster than all the other meta-heuristic methods investigated. As expected, the utilities show a slight decrease when compared to the GAIS algorithm, but are still higher than those of TABU search. The $GAF^3$ algorithm is a sensible system in that both its CPU time and average utility are highly compatible with the current application.

The fourth and last test was identical to the third test except that both the FF and FFF were used. The FFF was automatically used when the system judged the complexity of the problem to be too high for the FF to perform within an acceptable time limit. The following criterion was used to ascertain the complexity of the problem:

$$\begin{aligned} &\textit{if}\,(\textit{size}(\textit{IS solution}) + \textit{nb\_threats} > 34) && \textit{use FFF};\\ &\textit{else} && \textit{use FF}; \end{aligned} \tag{20}$$

The argument in favor of this criterion is that the size of the solution produced by the construction heuristic is approximately the size of the maximum independent set of the IG. Since the complexity of the FF is directly related to that size and the number of threats, several simulations were run until an optimal trade-off value (34) was found. The resulting algorithm's results are shown under the label GAH (Hybrid). The CPU times are practically equivalent to those of TABU search. The GAH delivers the best trade-off between precision in solution and speed of computation. The GAH is the final version of the algorithm proposed by this paper.

**Table 14.** Comparison of CPU time in milliseconds

| threats | IS | TABU | GAH | GAF[3] | GA | GAIS |
|---|---|---|---|---|---|---|
| 1 | 0 | 17 | 59 | 53 | 55 | 46 |
| 2 | 0 | 25 | 92 | 61 | 84 | 87 |
| 3 | 0 | 89 | 216 | 110 | 220 | 237 |
| 4 | 1 | 186 | 466 | 150 | 809 | 815 |
| 5 | 2 | 489 | 1527 | 254 | 2392 | 2670 |
| 6 | 2 | 1201 | 1775 | 345 | 4375 | 4429 |
| 7 | 7 | 2304 | 3244 | 405 | 32014 | 33161 |
| 8 | 14 | 4212 | 3151 | 553 | 66587 | 68550 |

**Table 15.** Comparison of Utility

| threats | IS | TABU | GAH | GAF[3] | GA | GAIS |
|---|---|---|---|---|---|---|
| 1 | 0.9827 | 0.9827 | 0.9832 | 0.9832 | 0.9832 | 0.9832 |
| 2 | 0.9319 | 0.9375 | 0.9514 | 0.9401 | 0.9514 | 0.9513 |
| 3 | 0.8581 | 0.8758 | 0.9002 | 0.8838 | 0.8987 | 0.9006 |
| 4 | 0.7282 | 0.7511 | 0.8224 | 0.7663 | 0.8131 | 0.8233 |
| 5 | 0.6300 | 0.6521 | 0.7435 | 0.6952 | 0.7230 | 0.7448 |
| 6 | 0.5131 | 0.5269 | 0.6477 | 0.5971 | 0.5944 | 0.6536 |
| 7 | 0.4831 | 0.4906 | 0.5724 | 0.5385 | 0.5133 | 0.5958 |
| 8 | 0.3886 | 0.3917 | 0.4569 | 0.4490 | 0.4060 | 0.4845 |

## *7.3. Genealogy*

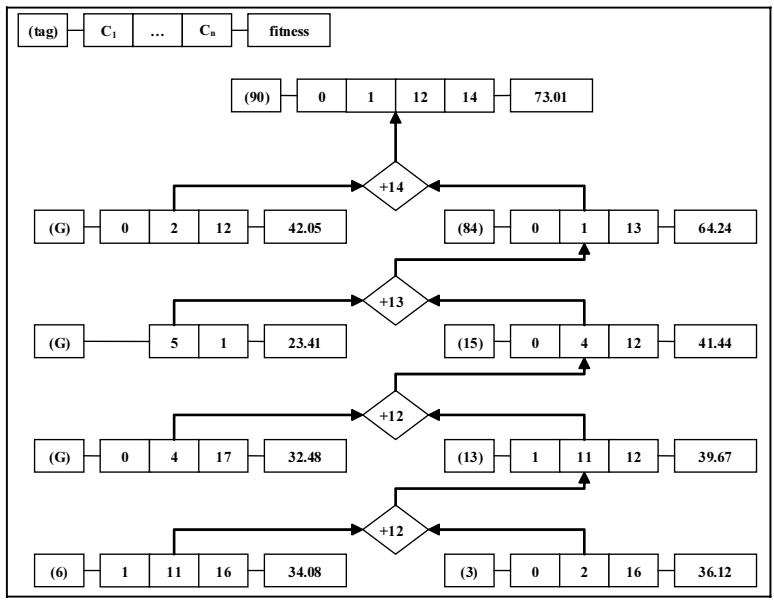



**Figure 13.** Genealogical tree of optimal solution.

For the scenario presented in section 2 (3 threats, 20 possible engagements), the GA found the optimal solution twice (solution 67 and solution 90), both of which with completely different genealogical trees.

Figure 13 shows the genealogy of solution 90. Solutions 6 and 3 are the product of the RCG. Solution 13 is the product of a crossover between 6 and 3 and a mutation that added engagement 12. Solution 15 is the product of a crossover between 13 and a ghost and a mutation that added engagement 12. And so on until solution 90. It finds amongst its descendents 5 normal solutions and three randomly generated ghosts. Good heredity and a high percentage of successful mutations is key in the attainment of this solution. The Repopulate operator is also responsible for the speed with which this solution was found. The crossover of two solutions will produce a child that is often in a totally different region of the solution space, while the mutate operator optimizes locally.

## 8. Concluding Remarks

This paper describes a construction heuristic and compares two improvement heuristics, namely TABU search and a Genetic Algorithm, for generating contingency plans aimed at guiding the allocation and scheduling of different types of weapon resources against anti-ship missiles. The results show that the novel GA developed here allows substantially better results than the more rigid TABU search without sacrificing processing time. Also, the memory requirements of the GA are not prohibitive because the 50 possible solutions present in memory are represented by relatively small arrays of integers and not by large binary trees. The main advantage is that the GA performs a global search as opposed to the TABU neighborhood search.

Improvements on the problem model could include more realistic probabilities and trajectories for the missiles as well as a dynamically changing engagement list. Also, the fact that some engagements are impossible to implement when a solution is arrived at (e.g. an engagement with fire time of 2.5s being present in a CP that took 3s to construct) should be taken into account. Currently, a solution is produced over a single planning episode. Considering multiple planning episodes over time would have a substantial impact on the design of the system. For example, a part of the plan would typically be optimized while the first part is being executed. It was also assumed that an infinite amount of SAMs and rounds were available, this would not be the case in a more realistic setting. Such additions would mean, in particular, that inserting a new engagement in a plan would not necessarily improve its associated utility. Consequently, the behavior of the TABU search heuristic (oscillating at the frontier of a saturated plan) could prove to be inadequate [1]. After such considerations, it is clear that the flexibility of the proposed GA should be an essential part of any system for solving real-time RM problems, as it is much less affected by the landscape of the solution space than TABU search. This is crucial because in a realistic setting the solution space would dynamically change over time. The GA is able to adapt because it can easily find novel solutions in a relatively short time, on the other hand, TABU search could very well be sequestered in a region of the solution space which no longer contains any kind of optima (local or global). Thus, the novel GA developed here offers several advantages in application to realistic RM in naval warfare. Most importantly it allows for quick on-line solution of a problem and offers great mobility in its search as opposed to a hill climbing neighborhood search. It is also important to note that the two aspects of a GA that were incompatible with the present application (large computation times and memory allocation) are dealt with and that the resulting algorithm offers more promise for future endeavors than TABU search.

## Acknowledgements

The authors would like to thank Michel Mayrand, for his programming skills and indispensable advice, Jean Couture for his help with some of the proprietary algorithms (BEL, IS and TABU) used in this research, and Elisa Shahbazian for the idea of applying a GA to the RM problem.

## References

[1] "A Tabu Search Heuristic for Resource Management in Naval Warfare," Dale E. Blodgett, Michel Gendreau, Francois Guertin, Jean-Yves Potvin, Rene Seguin.
[2] "Computers and Intractability: A Guide to the Theory of NP-Completeness," M.R. Garey and D.S. Johnson, New York: W.H. Freeman, 1983.
[3] "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," P. Moscato, Tech. Rep. Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.
[4] "Genetic Algorithms in Search, Optimization, and Machine Learning," D.E. Goldberg,, Addison-Wesley, 1989.
[5] "Neural Network Training Using Genetic Algorithms," A.J.F. van Rooij, L.C. Jain, R.P. Johnson, World Scientific, Series in Machine Perception and Artificial Intelligence – Vol. 26, 1996.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Calculus of Variations and Data Fusion

## *Methods and Examples*

Paolo CORNA[a], Lorella FATONE[b] and Francesco ZIRILLI[c]
[a] *Via Silvio Pellico, 4, 20030 Seveso (MI), Italy, e-mail: paolo.corna@libero.it*
[b] *Dipartimento di Matematica Pura ed Applicata, Università di Modena e Reggio Emilia,Via Campi 213/b, 41100 Modena (MO), Italy, e-mail: fatone.lorella@unimo.it*
[c] *Dipartimento di Matematica "G. Castelnuovo," Università di Roma "La Sapienza," Piazzale Aldo Moro 2, 00185 Roma, Italy, e-mail: f.zirilli@caspur.it*

**Abstract.** We present some data fusion procedures based on the use of partial differential equations, optimization problems and some ideas drawn from the calculus of variations. We present two real situations in which these fusion procedures have been successfully applied: the fusion of remote sensed SAR and optical images, and the fusion of multifrequency electromagnetic scattering data obtained in a laboratory experiment. Some examples of fusion results obtained through the use of real data are shown at the end of the paper.

**Keywords.** Data fusion, optimization problems, calculus of variations

### 1. Introduction

This paper considers some aspects of data fusion. We restrict our attention to the problem of fusing two images referring to the same scene taken (in general) by two different sensors. We always assume that the two images to be fused not only refer to the same scene but that they are also coregistered. The two images considered, being taken with two different sensors, may be in some sense synergic or redundant, and may be somehow characterized by some peculiar features. A data fusion procedure is a method that tries to exploit the situation described above to achieve a better and more reliable understanding of the observed scene than one, which can be obtained by analyzing the two images one by one.

We can say that recovering the content of the scene from the analysis of a single image amounts to the solution of an inverse problem, while recovering the content of the scene through a data fusion procedure starting from two independent coregistered images of the scene under consideration, amounts to the joint solution of two inverse problems.

There are many situations where a data fusion procedure can be used. We mention only two of them: remote sensing, when the Earth surface is observed with several instruments traveling on board of airplanes or satellites (see, for example, [1]), and medical imaging, when the same part of a patient's body is observed with two different instruments (see, for example, [2]). Later we show two examples: the fusion of SAR (Synthetic Aperture Radar) and optical satellite data of the Earth surface, and the fusion of multifrequency electromagnetic scattering data obtained in a laboratory experiment.

Further references and details about data fusion and its applications can be found, for example, in [3–5] and in the references quoted there.

Let us note, that from a practical point of view, data fusion approaches based on mathematical models only recently became usable in real problems. This is due to the fact that most scientific data today are digitized (hence, for example, an image can be represented as a matrix of numbers) and, above all, to the fact that efficient numerical algorithms, exploiting the power of today's computers, are now available to solve data fusion problems of a certain complexity.

There is a great variety of approaches and methods currently used in data fusion. We limit our analysis to a couple of methods based on the use of the following two ideas: *i*) the two images to be fused, since they refer to the same scene and are coregistered, have something in common that we call "structure"; *ii*) in order to extract the "structure" from a given image we must filter the image and apply a segmentation procedure. Note that we base the filtering step on the elementary properties of some Partial Differential Equations (PDE). That is, fusing two images means to minimize an objective function that represents the difference between their structures subject to constraints implied by the data (i.e. the images) available. The choice of some of the objective functions appearing in the fusion procedures considered here is suggested by the basic facts of calculus of variations.

An exhaustive analysis of these ideas can be found in [6–10] and in the web-sites:

http://web.unicam.it/matinf/fatone/esrin.asp,
http://web.unicam.it/matinf/fatone/w1

The paper is organized as follows: in Section 2 some facts about optimization, calculus of variations, and partial differential equations that will be used later are recalled without emphasis on mathematical rigour. In Section 3 we write down several mathematical models that can legitimately be interpreted as fusion procedures. In Section 4 the models introduced in Section 3 are applied to the examples involving real data mentioned previously and some of the results obtained are shown.

## 2. Mathematical Preliminaries

Let $\mathcal{R}$ be the real line and let $\mathcal{R}^n$ be the $n$-dimensional real Euclidean space. Let $\langle\cdot,\cdot\rangle$ denote the Euclidean scalar product in $\mathcal{R}^n$ and let $\|\cdot\|$ be the corresponding vector norm.

Given a continuously differentiable function $f$: $\mathcal{R}^n \to \mathcal{R}$, called *objective function*, we consider the following problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}),\ \boldsymbol{x} \in \mathcal{R}^n, \tag{1}$$

that is the *unconstrained optimization (minimization) problem* associated to the function $f$. Usually (1) is solved with an iterative method of the following type (see [11]):

given an initial vector $\boldsymbol{x}^{(0)} \in \mathcal{R}^n$, for $k \geq 0$ compute until convergence:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}, \tag{2}$$

where $\boldsymbol{d}^{(k)}$ is a suitable direction and $\alpha_k$ is a positive parameter, called *stepsize,* that at each iteration measures the length of the step along the direction $\boldsymbol{d}^{(k)}$.

In particular, if we denote with $\nabla\cdot$ the gradient of the function $\cdot$, the direction $\boldsymbol{d}^{(k)}$ is called a *descent direction* at the point $\boldsymbol{x}^{(k)}$:

$$\langle \boldsymbol{d}^{(k)}, \nabla f(\boldsymbol{x}^{(k)})\rangle < 0, \quad \text{if} \quad \nabla f(\boldsymbol{x}^{(k)}) \neq \boldsymbol{0},$$
$$\boldsymbol{d}^{(k)} = \boldsymbol{0}, \quad \text{if} \quad \nabla f(\boldsymbol{x}^{(k)}) = \boldsymbol{0}. \tag{3}$$

A *descent method* is a method of type (2) such that for $k \geq 0$ the vector $\boldsymbol{d}^{(k)}$ is a descent direction at the point $\boldsymbol{x}^{(k)}$.

Since the Taylor's formula ensures that there exists $\xi^{(k)}$ belonging to the segment of end points $\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$ such that:

$$f(\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}) - f(\boldsymbol{x}^{(k)}) = \alpha_k \langle \nabla f(\xi^{(k)}), \boldsymbol{d}^{(k)}\rangle, \tag{4}$$

we can conclude that, if $\alpha_k$ is sufficiently small along a descent direction at the point $\boldsymbol{x}^{(k)}$, we have:

$$f(\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}) < f(\boldsymbol{x}^{(k)}), \tag{5}$$

and hence, if $\alpha_k$ is chosen sufficiently small, the function $f$ decreases at each iteration of method (2) when, for $k \geq 0$, the direction $\boldsymbol{d}^{(k)}$ is a descent direction at the point $\boldsymbol{x}^{(k)}$.

Different choices of the directions $\boldsymbol{d}^{(k)}$, $k \geq 0$ correspond to different methods. In particular, choosing $\boldsymbol{d}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$, $k \geq 0$ we obtain the *gradient decent method* also called *steepest descent method* which exploits the fact that the fastest rate of descent of the function $f$ at the point $\boldsymbol{x}^{(k)}$ is reached in the direction $-\nabla f(\boldsymbol{x}^{(k)})$.

For the choice of the step sizes $\alpha_k$, $k \geq 0$, we refer to [11] for many different alternatives. However the basic idea in the choice of $\alpha_k$ is that $\alpha_k$ must be determined in such way that (5) is fulfilled without resorting to excessively small stepsizes $\alpha_k$ to avoid iterative methods with a slow convergence rate.

If we suppose that the variable $\boldsymbol{x} \in \mathbb{R}^n$ depends on a real variable $t$, $t \geq 0$, i.e. $\boldsymbol{x} = \boldsymbol{x}(t)$, $t \geq 0$, choosing $\alpha_k = \Delta t > 0$ for $k \geq 0$, the method (2) as $\Delta t \to 0$ generates the steepest descent trajectory associated to the function $f$ with initial point $\boldsymbol{x}^{(0)}$, that is the solution of the following initial value problem:

$$\frac{d\boldsymbol{x}}{dt} = -\nabla f(\boldsymbol{x}), \quad t > 0, \tag{6}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}^{(0)}. \tag{7}$$

The real variable $t$ is often called time variable.

The minimization method based on (6), (7), that consists in following the trajectory implicitly defined by (6), (7), is called *continuous steepest descent method*.

Let $\boldsymbol{x}(t;\boldsymbol{x}^{(0)})$ be the solution of (6), (7), we have:

$$\frac{df(\mathbf{x}(t;\mathbf{x}^{(0)}))}{dt} = \left\langle \nabla f(\mathbf{x}(t;\mathbf{x}^{(0)})), \frac{d\mathbf{x}(t;\mathbf{x}^{(0)})}{dt} \right\rangle = \tag{8}$$

$$-\left\langle \nabla f(\mathbf{x}(t;\mathbf{x}^{(0)})), \nabla f(\mathbf{x}(t;\mathbf{x}^{(0)})) \right\rangle = -\left\| \nabla f(\mathbf{x}(t;\mathbf{x}^{(0)})) \right\|^2 \leq 0\,, \quad t>0,$$

and hence $f(\mathbf{x}(t;\mathbf{x}^{(0)}))$ is a non increasing function of $t$, $t \geq 0$.

Let us note that a possible approach to solve problem (1) consists in calculating the following limit:

$$\lim_{t\to+\infty} \boldsymbol{x}(t;\boldsymbol{x}^{(0)}) = \boldsymbol{x}^*\,. \tag{9}$$

The point $\boldsymbol{x}^*$, if it exists, is a candidate to be a minimizer of $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{R}^n$. Note that for a general $f$, $\boldsymbol{x}^*$ may not exist or, when it exists, may not be a minimizer of $f$. Moreover $\boldsymbol{x}^*$, when it exists, depends on $f$ and on $\boldsymbol{x}^{(0)}$. That is the behaviour of the continuous steepest descent method as a minimization method depends on $f$ and on $\boldsymbol{x}^{(0)}$.

Let us consider now the steepest descent method in a more general framework. Let $u(x)$, $x \in [0,1]$, be a real function belonging to a suitable class of functions $X$, let us consider the following unconstrained minimization problem:

$$\min_u J(u)\,, \quad u \in X, \tag{10}$$

where $J(u)$ is the following functional:

$$J(u) = \frac{1}{2}\int_0^1 \left(\frac{du}{dx}\right)^2 \mathrm{d}x\,, \tag{11}$$

which is called the Dirichlet integral functional. Note that the function space $X$ in (10) plays the role of $\mathcal{R}^n$ in (1).

Moreover let us assume that for $u \in X$ the following boundary conditions hold:

$$\frac{du}{dx}(0) = 0\,, \quad \frac{du}{dx}(1) = 0\,. \tag{12}$$

The boundary conditions (12) are called Neumann boundary conditions.

Note that $u(x)$ = constant, $x \in [0,1]$ minimizes (11) and satisfies (12). That is, problem (10) has not a unique minimizer. However it is easy to enforce uniqueness of the minimizer by imposing a condition, for example, that the functions $u \in X$ have mean zero.

Let $u \in X$ and let $\delta u$ be an increment such that $u + \delta u \in X$; we have:

$$J(u+\delta u)-J(u)=\frac{1}{2}\int_0^1\left[\left(\frac{d\delta u}{dx}\right)^2+2\frac{du}{dx}\frac{d\delta u}{dx}\right]dx. \tag{13}$$

Integrating by parts, we obtain:

$$J(u+\delta u)-J(u)=\frac{du}{dx}\delta u\Big|_0^1-\int_0^1\frac{d^2u}{dx^2}\,\delta u\;dx+O(\delta u^2),$$

$$\text{when } \delta u\to 0, \tag{14}$$

where $O(\cdot)$ is the Landau symbol, so that:

$$\frac{\delta J(u)}{\delta u}=-\frac{d^2u}{dx^2},\quad u\in X, \tag{15}$$

and $\frac{\delta\cdot}{\delta u}$ means functional derivative of $\cdot$.

If the real function $u$ depends on both the time variable $t$ and on the spatial variable $x$, i.e. $u(t,x)$, $x\in[0,1]$, $t\geq 0$, then we can consider for problem (10), (11) an analogous of the continuous steepest descent method, that is:

$$\frac{\partial u}{\partial t}=-\frac{\delta J(u)}{\delta u},\quad x\in(0,1),\quad t>0, \tag{16}$$

$$u(0,x)=u_0(x),\qquad x\in[0,1], \tag{17}$$

where $u_0$ is a given function defined in [0,1].

In this case, using a more familiar notation to write (16), (17), we can see that the continuous steepest descent trajectory associated to the functional $J$ is the solution of the following Neumann-type initial-boundary value problem:

$$\frac{\partial u}{\partial t}=\frac{\partial^2u}{\partial x^2},\qquad x\in(0,1),\quad t>0, \tag{18}$$

$$\frac{\partial u}{\partial x}(t,0)=0,\ \frac{\partial u}{\partial x}(t,1)=0,\ t>0, \tag{19}$$

$$u(0,x)=u_0(x),\qquad x\in[0,1]. \tag{20}$$

Note that the boundary conditions (19) correspond to the boundary conditions (12).

Equation (18) is known as a *heat equation* and can be regarded as a *diffusion equation* where the diffusion coefficient is a constant equal to one.

More in general we can consider the function $u$ defined on a two-dimensional region, let us say, for example, the region Q = [0,1]× [0,1], and therefore given the real function $u(x,y)$, $(x,y) \in$ Q, we can reconsider the optimization problem (10) where now the two-dimensional Dirichlet integral functional $J_1(u)$, $u \in X_1$, is defined as follows:

$$J_1(u) = \frac{1}{2} \int_{\mathrm{Q}} \| \nabla u \|^2 dxdy, \quad u \in X_1, \tag{21}$$

where $X_1$ is a suitable function space that enforces the Neumann boundary condition on the boundary of Q, and $\|\nabla \cdot \|$ is the Euclidean norm of the gradient of the function $\cdot$ with respect to the $(x,y)$ variables.

It is easy to see that in this case the continuous steepest descent trajectory analogous to the one defined by (18), (19), (20) associated to the functional (21) satisfies the following Neumann-type initial-boundary value problem:

$$\frac{\partial u}{\partial t} = \mathrm{div}(\nabla u) = \Delta u, \quad (x,y) \in \mathrm{Q}, \quad t > 0, \tag{22}$$

$$\frac{\partial u}{\partial \mathbf{n}}(t,x,y) = 0, \quad (x,y) \in \partial \mathrm{Q}, \quad t > 0, \tag{23}$$

$$u(0,x,y) = v_0(x,y), \quad (x,y) \in \mathrm{Q}, \tag{24}$$

where $\mathrm{div}(\cdot)$ and $\Delta \cdot$ are respectively the divergence and the Laplacian of $\cdot$ with respect to the $(x,y)$ variables, $\partial \mathrm{Q}$ denotes the boundary of the region Q, $\frac{\partial u}{\partial \mathbf{n}}(t,x,y)$ is the derivative of $u$ in the direction $\mathbf{n}$ of the exterior unit normal vector to Q in $(x,y) \in \partial \mathrm{Q}$, and $v_0(x,y)$, $(x,y) \in$ Q is a given function defined on Q.

Equation (22) is known as the *heat equation* in two space variables.

## 3. Some Mathematical Models of Data Fusion Problems

We present some mathematical models to describe some data fusion problems, or more specifically, some image fusion problems. These models involve the use of PDE, of optimization problems and of some ideas taken from the calculus of variations that have been reviewed in the previous section.

The basic assumptions of the image fusion procedures that we present are twofold. The first one consists in assuming each image to be fused made of a few subregions where the image is smooth separated by boundaries where the image changes abruptly, and in associating to each image something that we call “structure” of the image. This assumption is satisfied in many situations of practical interest such as, for example, in remote sensing applications to agriculture. The second assumption is that images referring to the same scene must have the same structure.

In this framework, fusing $N$ images, $N \geq 2$, consists roughly in minimizing the difference of the structures of the $N$ images subject to the constraints posed by the data. For simplicity we consider only the $N = 2$ case, but what follows can be easily extended to the case $N>2$.

The fusion procedures we propose consist of the two following steps:

1. segmentation and noise removal;
2. fusion.

Let $\hat{u}_1$, $\hat{u}_2$ be two images to be fused referring to the same scene. In particular, $\hat{u}_1$, $\hat{u}_2$ can be seen as real functions defined on a rectangular region $R \subset \mathcal{R}^2$. In addition we suppose that they are coregistered.

Given $\hat{u}_1$, $\hat{u}_2$, the goal of step 1 is to obtain two segmented and noise free images, that is two images decomposed in piecewise smooth regions separated by boundaries where the images change abruptly. This can be accomplished in many different ways, and, according to [12], we propose the following one.

For $i = 1,2$, let $u_i(t,x,y)$, $(x,y) \in R$, $t \geq 0$, be a real function solution of the initial-boundary value problem that follows:

$$\frac{\partial u_i}{\partial t} = \mathrm{div}(s_{a_i,b_i}(\|\nabla u_i\|)\nabla u_i), \qquad (x,y) \in R, \quad t > 0, \tag{25}$$

$$\frac{\partial u_i}{\partial \boldsymbol{n}}(t,x,y) = 0, \qquad (x,y) \in \partial R, \quad t > 0, \tag{26}$$

$$u_i(0,x,y) = \hat{u}_i, \qquad (x,y) \in R, \tag{27}$$

where $a_i$ and $b_i$ are suitable real parameters such that $a_i > 0$, $b_i \neq 0$, and the function $s_{a_i,b_i}(\eta)$, $\eta \geq 0$, is chosen as follows:

$$s_{a_i,b_i}(\eta) = \frac{a_i}{1+\eta^2/b_i^2}, \qquad \eta \geq 0. \tag{28}$$

Note that equations (25) are diffusion equations whose behaviour is similar to the behaviour of the heat equation. In the case of the heat equation (22) the diffusion coefficient is identically equal to one, while in (25) the diffusion coefficients are given by $s_{a_i,b_i}(\|\nabla u_i\|)$, $i = 1,2$.

More in detail, given $T_i > 0$, $i = 1,2$, the solutions $u_i(T_i,x,y)$, $(x,y) \in R$, of (25), (26), (27), $i = 1,2$, are noise free images. In fact, for $i = 1,2$, where $\|\nabla u_i\|$ is large, that is in correspondence of the edges of the image, the diffusion coefficient $s_{a_i,b_i}(\|\nabla u_i\|)$ is small and therefore the exact localization of the "edges" of the image $\hat{u}_i$ is kept in the evolution in $t$, while where $\|\nabla u_i\|$ is small, the diffusion coeffi-

cient $s_{a_i,b_i}(\|\nabla u_i\|)$ is large and therefore the noisy homogeneous regions of the image are smoothed in the evolution in $t$. Thus the choice of the parameters $T_i$, $a_i$, $b_i$, $i$ = 1,2, corresponds to a sort of calibration of the filtering procedure induced by (25), (26), (27).

In order to obtain segmented images, let us define the following function which is an approximation of the Heaviside function:

$$S_{\tau_1,\tau_2}(\phi)=\begin{cases}0, & \phi<\tau_1,\\ P_m(\phi) & \tau_1\le\phi<\tau_2,\\ 1 & \phi\ge\tau_2,\end{cases} \tag{29}$$

where $P_m$ is a suitable polynomial in one variable of degree $m>1$, and the threshold parameters $\tau_1,\tau_2$ with $0<\tau_1<\tau_2$ are chosen to guarantee the differentiability and monotonicity of (29). We refer to $S_{\tau_1,\tau_2}$ as the "structure" function.

For $i$ = 1,2, the edges of the noise free image $u_i(T_i,x,y)$, can be approximated by the set:

$$\{\ (x,y)\in R:\ S_{\tau_1,\tau_2}\ (\|\nabla u_i(T_i,x,y)\|)=1\ \}. \tag{30}$$

In this way, for $i$ = 1,2, we are able to associate to each image $\hat{u}_i$ its "structure," that is the characteristic function of the set (30).

For $i$ = 1,2, the segmented and noise free images $\hat{u}_i(x,y)$, $(x,y)\in$ R, are the restrictions of $u_i(T_i,x,y)$, $(x,y)\in$ R, to the regions defined by the edges found in (30). Note that we assume that the edges are described by a curve, so that they have (two dimensional) measure zero. For later convenience we indicate the segmented and noise free images corresponding to $\hat{u}_i(x,y)$, $(x,y)\in$ R, with $h_i$, $i$ = 1,2.

Let us note that the threshold parameters $\tau_1,\tau_2$ can be different for the two images and therefore, at least in principle, they should depend on the index $i$. For simplicity of notation we omit this dependence.

Proceeding as in the previous section, it is easy to see that for $i$ = 1,2, the trajectory $u_i(t,x,y)$, $(x,y)\in$ R, $t\ge 0$ defined implicitly for $t\ge 0$ by (25), (26), (27) is the continuous steepest descent trajectory with initial condition (27) associated to the minimization (with respect to $v_i$) of the following functional:

$$Y_i(v_i)=\frac{a_i b_i^2}{2}\int_R \ln\left(1+\frac{||\nabla v_i(x,y)||^2}{b_i^2}\right)dx\,dy,\quad i=1,2. \tag{31}$$

where $v_i$ is a function defined on R belonging to a suitable functional class, and $\ln(\cdot)$ means logarithm of $\cdot$.

Note that the functional (31) takes the place of the Dirichlet integral functional (21) and that (25), (26), (27) take the place of (22), (23), (24).

The goal of step 2 is to provide a method of fusing the images. We write down three different alternatives to perform step 2 that generalize the material presented in [13] and developed in [6–10].

The first fusion procedure that we propose consists in solving the following unconstrained minimization problem:

given a suitable norm $|||\cdot|||$, the "fused" images $U_1$, $U_2$ are obtained as the minimizer of the following problem:

$$\min_{w_1,w_2} \{ \; ||| S_{\tau_1,\tau_2}(|| \nabla w_1 ||) - S_{\tau_1,\tau_2}(|| \nabla w_2 ||) |||^2 +$$

$$\lambda_1 ||| w_1 - h_1 |||^2 + \lambda_2 ||| w_2 - h_2 |||^2 \; \}, \tag{32}$$

where $w_1, w_2$ are functions defined on $R$ and $\lambda_1$, $\lambda_2$ are positive penalization parameters.

The images $U_1$, $U_2$ solutions of problem (32) can be considered fused images since they are obtained changing the functions $w_1$, $w_2$ in order to obtain a small distance between their structures $S_{\tau_1,\tau_2}(|| \nabla w_1 ||)$, $S_{\tau_1,\tau_2}(|| \nabla w_2 ||)$ under the requirement that the images $w_1$ and $w_2$ remain close to the noise free data $h_1$ and $h_2$ respectively.

Note that the minimization problem (32) represents step 2 of the fusion procedure and that the presence of $h_i$, $i = 1,2$, in (32) represents step 1. That is, in the fusion procedure given by (32), the steps 1 and 2 are performed one after the other.

Behind the second fusion procedure proposed there is the attempt of improving the model used previously (i.e.: (32)) by taking into account more information about the nature of the images to be fused. In particular we assume that the images $\hat{u}_1$, $\hat{u}_2$ are the results of two physical experiments denoted generically with:

$$F_i(m_i) = \hat{u}_i, \quad \text{in } R, \; i = 1,2, \tag{33}$$

where $F_1$, $F_2$ are linear or non linear operators representing the physical experiments and $m_1(x,y)$, $m_2(x,y)$, $(x,y) \in R$ are the physical quantities measured in the experiments, that is they are the unknowns of the problem.

In the next section we give two examples of this framework using remote sensed data of the Earth surface and multifrequency electromagnetic scattering data taken in a laboratory experiment.

Let us note that for $i = 1,2$, given the data $\hat{u}_i$ and the operators $F_i$, the single sensor inverse problem consists in determining the unknown $m_i$ considering the two equations (33) separately, that is inverting individually the operators $F_i$, for $i = 1,2$.

Conversely a fusion approach tries to recover the unknowns $m_i$, $i$ = 1,2, considering the fact that the two images $\hat{u}_i$, $i$ = 1,2, refer to the same scene, and consequently the two equations (33) should be considered jointly.

The second fusion procedure we propose is the following:

given $\varepsilon_1 > 0$ and a suitable norm $|||\cdot|||$, the "fused" physical unknowns quantities $M_1$, $M_2$ are obtained as the minimizer of the following problem:

$$\min_{\eta_1,\eta_2} \left\{ |||\, S_{\tau_1,\tau_2}(\|\nabla\eta_1\|) - S_{\tau_1,\tau_2}(\|\nabla\eta_2\|)\,|||^2 \right\}, \tag{34}$$

subject to the constraint:

$$\kappa_1 |||\, F_1(\eta_1) - h_1\,||| \;\; + \kappa_2 |||\, F_2(\eta_2) - h_2\,||| \le \varepsilon_1, \tag{35}$$

where $\eta_1$, $\eta_2$ are functions defined on $R$ and $\kappa_1$, $\kappa_2$ are positive penalization parameters.

From the solution $M_1$, $M_2$ of problems (34), (35) the "fused" images $V_1$, $V_2$ can be computed bearing in mind (33), that is computing:

$$V_i = F_i(M_i), \quad \text{in } R, \; i = 1,2. \tag{36}$$

Note that the minimization problem (34), (35) represents step 2 of the fusion procedure and that the presence of $h_i$, $i$ = 1,2, in (35) represents step 1. That is also in the fusion procedure given by (34), (35) step 1 and 2 are performed one after the other.

The last fusion approach we propose (see [9]) performs together step 1 and 2, that is performs together the filtering step, based on the solution of problem (25), (26), (27), and the data fusion step, based on the solution of (34), (35), exploiting the basic facts about calculus of variations presented in Section 2.

That is the third fusion procedure we propose is the following:

given $\varepsilon_2 > 0$ and a suitable norm $|||\cdot|||$, the "fused" physical unknowns quantities $N_1$, $N_2$ are obtained as the minimizer of the following problem:

$$\min_{\mu_1,\mu_2} \left\{ |||\, S_{\tau_1,\tau_2}(\|\nabla\mu_1\|) - S_{\tau_1,\tau_2}(\|\nabla\mu_2\|)\,|||^2 + J(\mu_1,\mu_2) \right\}, \tag{37}$$

subject to the constraint:

$$\gamma_1 |||\, F_1(\mu_1) - \hat{u}_1\,||| \;\; + \gamma_2 |||\, F_2(\mu_2) - \hat{u}_2\,||| \le \varepsilon_2, \tag{38}$$

where $\mu_1$, $\mu_2$ are functions defined on $R$, $\gamma_1$, $\gamma_2$ are positive penalization parameters and $J(\mu_1,\mu_2)$ is the following functional:

$$J(\mu_1,\mu_2) = \mathrm{Y}_1(\mu_1) + \mathrm{Y}_2(\mu_2) \tag{39}$$

where $\mathrm{Y}_1$, $\mathrm{Y}_2$ are the functionals defined in (31).

To solve the minimization problem (37), (38) means to find two functions $N_1$, $N_2$ that make small $|||S_{\tau_1,\tau_2}(||\nabla\mu_1||) - S_{\tau_1,\tau_2}(||\nabla\mu_2||)|||$ (this corresponds to fuse the images) and that make small $J(\mu_1,\mu_2)$ (this corresponds to denoise the images) subject to the constraint (38) posed by the data.

Note that the minimization of the functional $J(\mu_1,\mu_2)$ corresponds to noise removal for the images with a procedure similar to the one defined by (25), (26), (27). In fact, it is easy to see that the trajectory ( $\hat{\mu}_1(t,x,y)$, $\hat{\mu}_2(t,x,y)$ ), $(x,y) \in R$, defined implicitly for $t > 0$ by (25), (26), (27), when we choose $\hat{u}_1$ as initial condition to generate $\hat{\mu}_1$ and $\hat{u}_2$ as initial condition to generate $\hat{\mu}_2$, is the continuous steepest descent trajectory associated to the minimization of the functional $J(\mu_1,\mu_2)$ associated to the initial condition ( $\hat{u}_1, \hat{u}_2$ ) (see comments about formula (31)).

From the solution $N_1$, $N_2$ of problem (37), (38), the "fused" images $W_1$, $W_2$ can be obtained as follows:

$$W_i = F_i(N_i), \quad \text{in } R,\ i = 1,2. \tag{40}$$

Note that the norm $|||\cdot|||$ that appears in the fusion procedures suggested above in our numerical experience has always been chosen as the Euclidean norm induced by the pixel structure of the images considered.

Let us conclude this section with two important observations about these fusion models. In most practical situations the discretization of the unknowns appearing in the fusion models leads to optimization problems involving a large number (the order of thousand or even millions) of independent variables. In fact in the most common situation the number of independent variables in the optimization problems is equal to the number of pixels contained in the images that must be fused. Furthermore, equations (33) are usually ill posed and, consequently, must be treated with care. We can conclude by asserting that the model problems proposed for image fusion are challenging optimization problems since they usually are high dimensional and involve ill-posed constraints.

## 4. Two Examples of Image Fusion

In this section we show how to apply the fusion procedures introduced in the previous section in two different real situations: the fusion of remote sensed SAR and optical

images of the Earth surface, and the fusion of multifrequency electromagnetic scattering data taken in a laboratory experiment. For a complete presentation and discussion of these items we refer the interested reader [6–10].

### *4.1. SAR/Optical Image Fusion*

As a first example we consider an application of the image fusion procedures introduced in Section 3 to the context of remote sensing.

We are interested in combining coregistered satellite images relative to the same scene on the Earth surface coming from two complementary sensors that provide repeated coverage of the planet on a regular basis: the SAR sensor of the ERS-ESA mission and an optical sensor called SPOT. The goal that we pursue is to obtain from the available data sets information of higher quality than those obtained when the data sets of the two sensors are considered separately.

There are several practical motivations behind this kind of fusion, such as, for example: Earth resources survey, water management, urban growth observation, study of agriculture, forestry, seashore analysis and fire alarm.

Referring to equations (33), if $\hat{u}_1$, $\hat{u}_2$ denote respectively the SAR and the optical image to be fused, we have that $F_1$ is the operator that models the SAR measurement process, while $F_2$ is the operator representing the optical measurement process. Moreover $m_1$ is the physical quantity measured by the SAR sensor, that is the reflectivity of the ground cover in the microwave spectrum (i.e. the backscattering coefficient), and $m_2$ is the physical quantity measured by the optical sensor, that is the reflectivity of the scene in the visible and near-infrared spectrum (i.e. the luminance).

Two simple mathematical models for $F_1$ and $F_2$ are discussed in [9]. Substantially these models construct the SAR and the optical images through a convolution equation between the average squared amplitude of the electromagnetic signal at the microwave and at the visible frequency, and the transfer function of the SAR and optical instrument respectively.

In particular let us denote with $A_1(\xi,\psi)$, $(\xi,\psi) \in \mathcal{R}^2$ the amplitude of the electromagnetic signal at the SAR frequency emitted from $(\xi,\psi) \in \mathcal{R}^2$. The quantity $A_1$ is modelled as a random variable. The SAR image $U_1$ is given by:

$$U_1(x,y) = \int_{\mathcal{R}^2} \left| h_S(\xi - x, \psi - y) \right|^2 E(A_1^2(\xi,\psi))\, d\xi\, d\psi, \tag{41}$$

where $E(\cdot)$ indicates the expected value of ·, and $h_S$ is the transfer function of the SAR image measurement simulator which takes into account the spatial resolution of the SAR image, and in general the functioning of the SAR instrument.

A model analogous to (41) holds also for the optical images. Let $A_2(\xi,\psi)$, $(\xi,\psi) \in \mathcal{R}^2$ be the amplitude of the electromagnetic signal at the optical frequency emitted from $(\xi,\psi) \in \mathcal{R}^2$. The quantity $A_2$ has been modelled as a random variable. The optical image $U_2$ is given by:

$$U_2(x,y) = \int_{\Re^2} \left| h_O(\xi - x, \psi - y) \right|^2 E(A_2^2(\xi,\psi))\ d\xi\ d\psi\,. \tag{42}$$

The function $h_O$ is the transfer function of the optical image measurement simulator. This function, besides the functioning of the instrument, models also the perturbation on the measured data induced by the presence of the atmosphere. This perturbation is not negligible when modeling optical measurements. A simple expression for $h_S$ and $h_O$ together with their characteristic parameters can be found in [9].

Let us note that the usual representation using pixels of the measured SAR and optical images, that in the notation of Section 3 have been denoted respectively with $\hat{u}_1$, $\hat{u}_2$, can be regarded as piecewise constant approximations of the functions $U_1$, $U_2$ given in (41), (42). In particular the value of $\hat{u}_1$, $\hat{u}_2$ in a pixel can be calculated as an average of $U_1$, $U_2$ respectively in the area contained in the pixel considered.

Moreover we remark that the two basic assumptions of the proposed fusion procedures, that is the assumption that the images are made of smooth regions separated by sharp boundaries, and the assumption that the changes of reflectivity and luminance in the scene occur in the same physical locations, and, consequently, that the SAR and the optical images have the same "structure," are reasonably fulfilled in the situation examined below where SAR and optical data relative to the surroundings of an urban area are analyzed.

As an example, we consider a pair of digital SAR and SPOT images representing a zone in the surroundings of Paris (France), see Figure 1(a), 1(b). Indeed, the SAR image is an ERS-SAR image with the following technical characteristics: range looks = azimuth looks = 1, while the optical image is one of the four SPOT-4 channels image. These two images have been pre-processed by SPOT Image in order to obtain, through a resampling, the same pixel size, i.e. the same spatial resolution, for the two images. Moreover the SPOT Image has coregistered the SAR and the SPOT images between themselves, taking into account the Digital Elevation Model (DEM) of the terrain shown in the images. At the end of this pre-processing step the spatial resolution of the two images is $20m \times 20m$ (where $m$ denotes meters).

The dimension of the images of Figure 1 is of $120 \times 120$ pixels. Let us note that on the whole there are $2 \times 120 \times 120$ pixels = 28˙800 pixels, and consequently the optimization problem corresponding to any one of the fusion procedures considered has 28˙800 independent variables, that is posed in 28˙800 dimensions. Note that an image of $120 \times 120$ pixels corresponds on the Earth surface to a square of $2.4km \times 2.4km$ (where $km$ denotes kilometers), that is a relatively small area. This confirms the fact that the optimization problems considered are usually in high dimension.

In all the images shown in Figure 1, the white color represents high values of the pixel variable (gray level = 255), while the black color represents low values of the pixel variable (gray level = 0). Moreover the unit in the $x$ and y axes is the edge of the pixel (i.e. $20m$).

The images of Figure 1(a), 1(b) have been fused with the fusion procedure (34), (35) and (37), (38). The corresponding fused radar reflectivity and fused luminance obtained with the fusion procedure (34), (35) are shown respectively in Figure 1(c), 1(d), while in Figure 1(e), 1(f) are shown respectively the fused radar reflectivity and luminance resulting from the fusion procedure (37), (38). We show only the numerical

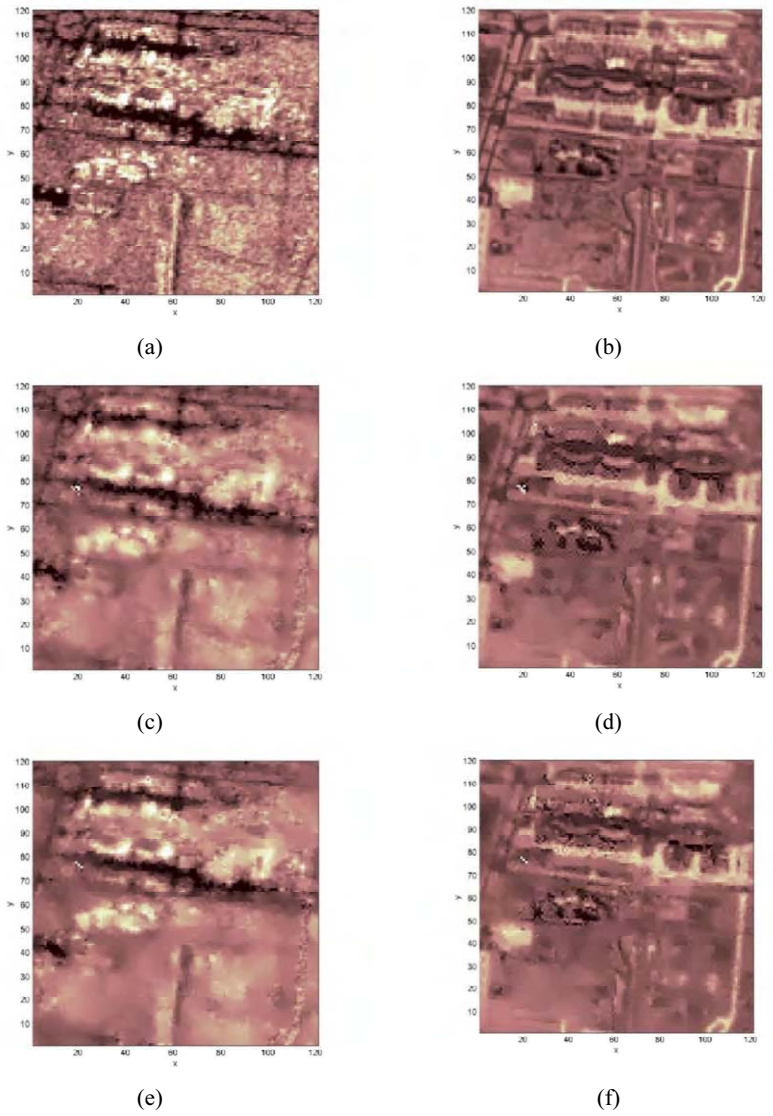


**Figure 1.** (a) Original ERS-SAR image; (b) original SPOT image; (c) fused radar reflectivity image (fusion procedure (34), (35)); (d) fused luminance image (fusion procedure (34), (35)); (e) fused radar reflectivity image (fusion procedure (37), (38)); (f) fused luminance image (fusion procedure (37), (38)).

results relative to the two fusion procedures (34), (35) and (37), (38) since they are intimately connected between themselves and are more "complete" than those obtained with the fusion procedure (32) since they involve the use of equations (33). Results obtained with the fusion procedure (32) can be seen in [6,8,10].

Note that in Figure 1 the fused images represent physical quantities different from those represented in the original images. This is due to the presence of the operators $F_1$, $F_2$ in the fusion procedures employed.

The optimization problem (34), (35) and (37), (38) have been discretized using the natural pixels structure of the images and using the finite differences approximations and the rectangular quadrature formulae to approximate respectively the derivatives and the integrals appearing in the optimization problems. The finite dimensional optimization problems obtained in this way have been solved using the optimization software package LANCELOT, see [14]. This package is particularly suited for the problems in high dimension characterized by many independent variables, and solves the optimization problems considered using a quasi-Newton method.

In both fusion experiments considered we have only used Euclidean norms and we have chosen $n = 5$ and $P_5(\phi) = 10\frac{(\phi-\tau_1)^3}{(\tau_2-\tau_1)^3} - 15\frac{(\phi-\tau_1)^4}{(\tau_2-\tau_1)^4} + 6\frac{(\phi-\tau_1)^5}{(\tau_2-\tau_1)^5}$ in (29). Moreover as for the choice of the parameters $\tau_1$, $\tau_2$, in order to avoid the difficulties arising from the highly non linear character of the function $S_{\tau_1,\tau_2}$ defined in (29), we have used an iterative process to reach appropriate values of $\tau_1$, $\tau_2$ with some kind of "continuation method" to link the different steps of the iteration; for more details we refer to [7,9].

Furthermore for the fusion procedure (34), (35) the following parameters have been chosen: $T_1 = 10$, $T_2 = 2$, $a_1 = 1$, $a_2 = 1$, $b_1 = 0.5$, $b_2 = 6$, $\kappa_1 = 1$, $\kappa_2 = 1$, $\varepsilon_1 = 0.1$, while for the fusion procedure (37), (38) the following choices have been made: $a_1 = 8\cdot10^{-6}$, $a_2 = 55\cdot10^{-10}$, $b_1 = 0.5$, $b_2 = 6$, $\gamma_1 = 1$, $\gamma_2 = 1$, $\varepsilon_2 = 0.1$.

Let us note that there are several other choices of the above parameters that lead to good results. In particular the value of these parameters must depend on the characteristics of the images to be processed and ad hoc calibrations are often needed. It is important to underline that the penalization parameters, that is $\lambda_1$, $\lambda_2$ in (32), $\kappa_1$, $\kappa_2$ in (35), and $\gamma_1$, $\gamma_2$ in (38), are those that more than the others determine the quality of the fusion products. In fact they "weigh" the contribution of each image in the fusion procedure and are able to balance the entire fusion procedure when, for some reason, the quality of the two images is uneven.

For example, from the choice of the above penalization parameters, it is possible to determine which one of the two images must "predominate" in the fusion procedure (and this can be obtained taking the penalization parameter corresponding to the "predominating" image bigger than the other penalization parameter), or it is possible to consider the two images as having the same "weight" in the fusion process (and this can be obtained taking the two penalization parameters of the same order of magnitude).

Some other numerical results concerning the fusion of SAR and optical images can be found in [6,8–10] and some interesting animations are shown in the web-sites:

http://web.unicam.it/matinf/fatone/esrin.asp,
http://web.unicam.it/matinf/fatone/w1.

Altogether the results obtained from the numerical experiments are satisfactory from both qualitative and quantitative points of view. In fact some qualitative aspects of the results of the fusion procedures can be seen observing the original and the fused images in Figure 1. For example, in Figure 1 it can be observed that the characteristics of both the SAR and the optical images are easily recognizable in all the fused images.

Moreover it is possible to note that the homogeneous zones of both the SAR and optical images emerge in the fused images and that the boundaries of these zones are more delineated and marked in the fused images than in the original ones.

A quantitative analysis of the quality improvement obtained with the fusion procedures is instead more demanding. In [10] we have tried to give a quantitative basis to the following statement: "the quality of the information obtained from the fused images is higher than the quality of the information provided by the original images taken one by one." This is done only in relatively simple circumstances adapting to our purposes some artificial vision techniques. Also in this case the results are quite satisfactory and they show that a fusion procedure may bring effective improvements in the analysis of the remote sensed images.

### *4.2. Multifrequency Electromagnetic Scattering Data Fusion*

The data fusion approaches presented in Section 3 have been applied in another context: the numerical inversion of multifrequency electromagnetic scattering data, that is electromagnetic scattering data relative to the same scene collected at different frequencies in a laboratory experiment. For a more detailed discussion of this problem we refer to [7] and to the references quoted there.

The scene considered is a region of the space containing an unknown inhomogeneity (i.e. the obstacle) used as a scatter in the electromagnetic experiment. When known electromagnetic waves hit the obstacle, scattered electromagnetic waves are generated. In this context we consider the following inverse problem: given some known incident electromagnetic fields, reconstruct the refraction index of a cylindrically symmetric (along the $z$-axis) inhomogeneity from some knowledge (i.e. the "far fields") of the corresponding scattered electromagnetic fields.

We suppose that the incident electromagnetic waves are time-harmonic plane waves linearly polarized along the $z$-axis, that is the axis of symmetry of the inhomogeneity. Moreover, if we assume that the nonhomogeneity is contained in a vertical cylinder with compact cross section $B \subset \mathbb{R}^2$, the electromagnetic scattering problem can be reduced to a two-dimensional problem for a scalar field. That is the scene of the experiment reduces to a compact region of the plane, i.e. B, the cross section of a rectangular cylinder containing the inhomogeneity.

We assume that the refraction indices of the scene considered are functions of the location in the scene and depend on the frequency $f$ of the incoming electromagnetic waves used in the experiment. From the knowledge of the incident electromagnetic fields and of the corresponding scattered data at different frequencies, we want to reconstruct the refraction indices of the scene at the frequencies used in the experiment.

We limit our attention to experiments involving data at two different frequencies and we approach the problem of reconstructing the refraction indices with the fusion procedures presented in Section 3. In this case the basic assumptions of the proposed fusion procedures are that the refraction indices of the scene at the different frequencies are piecewise constant functions and that their gradients change abruptly in the same physical locations. These assumptions are very reasonable in the data analysed in the laboratory experiment.

For $i = 1,2$, let $n_{k_i}(x,y)$ be the refraction index of the scene considered as a function of the location $(x,y) \in B$ in the scene and of the wave number $k_i$. Let us note that,

for $i = 1,2$, the wave number $k_i$ depends on the frequency $f = f_i$ considered in the experiment (see [7]). Let us define $m_{k_i}(x,y) = 1 - n_{k_i}(x,y)$, $(x,y) \in$ B, $i = 1,2$.

For $i = 1,2$, let $u^i_{k_i}$ be the $z$-component of the incident electric field, and let $u_{k_i}$ be the $z$-component of the "far field" of the scattered electric field generated by the inhomogeneity when hit by $u^i_{k_i}$. Moreover let $S_1 = \{ (x,y) \in \mathcal{R}^2 : \|(x,y)\| = 1 \}$ be the sphere of $\mathcal{R}^2$ of center the origin and radius one.

For $i = 1,2$, in the Born approximation, the operator $F_{k_i}$ that models the measurement process is given by:

$$u_{k_i}(\hat{x},\hat{y},\hat{\alpha}_1,\hat{\alpha}_2) =$$

$$-\frac{k_i^{3/2}}{\sqrt{8\pi}}\, e^{i\pi/4} \int_B m_{k_i}(\xi,\psi) e^{-ik_i\langle (\hat{x},\hat{y}),(\xi,\psi)\rangle} u^i_{k_i}(\xi,\psi,\hat{\alpha}_1,\hat{\alpha}_2)\; d\xi\, d\psi\,,$$

$$(\hat{x},\hat{y}), (\hat{\alpha}_1,\hat{\alpha}_2) \in S_1,\ i = 1,2, \tag{43}$$

where i is the imaginary unit, $(\hat{x},\hat{y}) = \dfrac{(x,y)}{\|(x,y)\|}$, $(x,y) \neq (0,0)$ and $(\hat{\alpha}_1,\hat{\alpha}_2) \in \mathcal{R}^2$, with $\|(\hat{\alpha}_1,\hat{\alpha}_2)\| = 1$, is the propagation direction of $u^i_{k_i}$.

Let us note that the integral equations (43) in the unknowns $m_{k_i}$, $i = 1,2$, that model the measurement processes are Fredholm integral equations of the first kind and that they are ill posed.

In [7] the fusion procedure (34), (35) with the choice (43) for the operators representing the physical experiments, has been used with satisfactory results to treat the multifrequency electromagnetic scattering data taken at the Institute Fresnel, CNRS (Marseille, France) in a laboratory experiment. The interested reader can find in [7] the numerical results omitted here for reasons of brevity.

## Acknowledgements

The authors thank ESA (Esrin, Italy), SPOT Image (France) and the Institute Fresnel (France) for making available the data used in the numerical experiments.

Finally the authors thank A.R. Conn, N.I.M. Gould and P.L. Toint for making available, free of charge, the optimization package LANCELOT.

## References

[1] Pohl, C. and Genderen, J.L. van (1998). "Multisensor image fusion in remote sensing: concepts, methods and applications," International Journal of Remote Sensing, 19, 823–854.

[2] Townsend, D.W. and Cherry S.R. (2001). "Combining anatomy and function: the path to true image fusion," European Radiology, 11, 1968–1974.

[3] Wald, L. (1999). "Some terms of reference in data fusion," IEEE Transactions on Geoscience and Remote Sensing, 37(3), 1190–1193.
[4] Waltz, E. and Llinas, J. (1990). *Multisensor Data Fusion,* Artech House, Boston, Mass.
[5] (1997) Special issue on data fusion, Proceedings of the IEEE, 85, 1–208.
[6] Fatone, L., Maponi, P. and Zirilli, F. (2001). "Fusion of SAR/Optical images to detect urban areas," in "Proceedings of the IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, Roma, Italy," IEEE Publisher, Piscataway N.J., 217–221.
[7] Fatone, L., Maponi, P. and Zirilli, F. (2001). "An image fusion approach to the numerical inversion of multifrequency electromagnetic scattering data," Inverse Problems 17, 1689–1702.
[8] Fatone, L., Maponi, P. and Zirilli, F. (2002). "Data fusion and nonlinear optimization," SIAM News 35(1), 4; 10.
[9] Fatone, L., Maponi, P. and Zirilli, F. (2003). "Data fusion and filtering via calculus of variations: an application to SAR/optical data and urban areas detection," submitted to ISPRS Journal of Photogrammetry and Remote Sensing.
[10] Corna, P., Fatone, L. and Zirilli, F. (2003). "Data fusion and quality assessment of fusion products: methods and examples," submitted to Electronics & Communication Engineering Journal.
[11] Gill, P., Murray, W. and Wright, M. (1981). *Practical Optimization*, Academic Press, New York.
[12] Perona, P. and Malik, J. (1990). "Scale space and edge detection using anisotropic diffusion," IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7), 629–639.
[13] Haber, E. and Oldenburg, D. (1997). "Joint inversion: a structural approach," Inverse Problems, 13, 63–77.
[14] Conn, A.R., Gould, N.I.M. and Toint Ph.L. (1992). *LANCELOT: A Fortran Package for Large Scale Nonlinear Optimization* (*Release A*), Springer-Verlag, Berlin.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Reliability in Multiple Hypotheses Testing and Identification Problems

Evgueni HAROUTUNIAN
*Tsakhkadzor, August 26, 2003, ASI Session*

**Abstract.** The procedures of the identification of probability distributions for $K(\geq 1)$ random objects, each having one from the known set of $M$ distributions, are studied. $K$ sequences of discrete independent random variables represent results of $N$ observations of each of these objects. The exponential decrease of test error probabilities is considered. The reliability matrices of logarithmically asymptotically optimal tests are investigated for some models. These models are determined by conditions of dependence or independence of objects and by the formulation of an identification problem. The optimal subsets of reliabilities which may be given beforehand and conditions of positiveness of all of the reliabilities are investigated.

**Keywords.** Statistical hypothesis testing, decision making, error probabilities, asymptotical optimal tests, identification problem

## 1. Introduction

This paper introduces the readers to an example of specific problems related to decision making, data classification, and identification which mathematicians are currently considering.

## 2. Problem Statement

Let $\boldsymbol{X}_k = (X_{k,n}, n \in [N])$, $k \in [K]$, are $K$ sequences each of $N$ discrete independent identically distributed random variables representing possible results of $N$ observations, respectively, each of $K$ randomly functioning objects.

For $k \in [K]$, $n \in [N]$, $X_{k,n}$ assumes values $x_{k,n}$ in the finite set $X$ of cardinality $|X|$. Let $P(X)$ be the space of all possible distributions on $X$. $M \geq K$ probability distributions $G_1, \ldots, G_M$ from $P(X)$ are given, some of which are assigned to vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K$. This assignment is unknown and must be decided on the base of results of $N$ independent observations $\mathbf{x}_k = (x_{k,1}, \ldots, x_{k,N})$, where $x_{k,n}$ is a result of the $n$-th observation of the $k$-th object.

When $M = K$ and all objects are different (any two objects cannot have the same distribution), there are $K!$ possible decisions. When objects are independent there are $M^K$ possible combinations.

Sequential multiple-decision procedures are presented in [1]. Chapter 10 of the book, "Search Problems" [2] by Ahlswede and Wegener, is devoted to statistical identification and ranking problems.

We study models considered in [1] and [2] and variations of these models inspired by the pioneering papers of Ahlswede and Dueck [3,4], and apply the concept of optimality developed for these models with $K = 1$ [5–10].

Let us consider the following family of error probabilities of the test

$$\alpha^{(N)}_{m_1,m_2,...,m_K|l_1,l_2,...,l_K},\ (m_1,m_2,...,m_K) \neq (l_1,l_2,...,l_K),$$
$$m_k, l_k \in [M],\quad k \in [K]$$

which are the probabilities of decisions $(l_1,l_2,...,l_K)$ when in reality distributions were $(m_1,m_2,...,m_K)$.

Also let us define the probability of rejecting distributions $(m_1,m_2,...,m_K)$ when they are true as follows:

$$\alpha^{(N)}_{m_1,m_2,...,m_K|m_1,m_2,...,m_K} = \sum_{\substack{(l_1,l_2,...,l_K)\neq \\ (m_1,m_2,...,m_K)}} \alpha^{(N)}_{m_1,m_2,...,m_K|l_1,l_2,...,l_K}. \tag{1}$$

We study the exponential decrease of error probabilities and define reliabilities:

$$\overline{\lim_{N\to\infty}} -\frac{1}{N}\log \alpha^{(N)}_{m_1,m_2,...,m_K|l_1,l_2,...,l_K} = E_{m_1,m_2,...,m_K|l_1,l_2,...,l_K} \geq 0 \tag{2}$$

The exponents of error probabilities will be called reliabilities.

It is useful to note that from (1) it follows that for any $(m_1,m_2,...,m_K)$ we have

$$E_{m_1,m_2,...,m_K|m_1,m_2,...,m_K} = \min_{\substack{(l_1,l_2,...,l_K)\neq \\ (m_1,m_2,...,m_K)}} E_{m_1,m_2,...,m_K|l_1,l_2,...,l_K}$$

Our criterion of optimality is the following. Given *M, K*, and values of a subset of the reliabilities, we are looking for the best (the largest) values for the unknown reliabilities. In addition, it is necessary to describe the conditions under which all these reliabilities are positive. The procedure that realizes such testing is identification, which following Birgé [9] we call "logarithmically asymptotically optimal" (LAO).

Let $N(x \mid \mathbf{x})$ be the number of repetitions of the element $x \in X$ in the vector $\mathbf{x} \in X^N$, and let

$$Q = \{Q(x) = N(x \mid \mathbf{x})/N, \quad x \in X\}$$

be the distribution, which is called "empirical distribution" of the vector $\mathbf{x}$ in statistics, "the type" [11,12] in information theory, and "the composition" in the algebraic literature.

Let us denote the space of all empirical distributions for a given $N$ by $P^{(N)}(X)$ and the set of all vectors of the type $Q \in P^{(N)}(X)$ by $T_Q^{(N)}$.

Consider for $k \in [K]$, $m \in [M]$, divergences

$$D(Q_k \| G_m) = \sum_{x \in X} Q_k(x) \log \frac{Q_k(x)}{G_m(x)}$$

and entropies

$$H(Q_k) = -\sum_{x \in X} Q_k(x) \log Q_k(x)$$

We shall use the following representations for $m_k \in [M]$, $k \in [K]$, when $G_{m_k}$ is the distribution of the $k$-th object:

$$P^{(N)}_{m_1, m_2, \ldots, m_K}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K) = \exp\{-N[\sum_{k=1}^{K} D(Q_k \| G_{m_k}) + H(Q_k)]\} \qquad (3)$$

This representation follows from the independence of the $N$ observations and of the $K$ objects and from the definitions of divergences and entropies.

Notice that the equality (3) is valid even when the left part is equal to $0$. In that case for one of $\mathbf{x}_k$ the distribution $Q_k$ is not absolutely continuous relative to $G_{m_k}$ and $D(Q_k \| G_{m_k}) = \infty$.

Our arguments will be based on the following fact: the "maximal likelihood" test accepts as solution values $m_1, m_2, \ldots, m_k$, which maximize the probability $P^{(N)}_{m_1, m_2, \ldots, m_K}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K)$, but from (3) we see that the same solution can be obtained by minimization of the sum

$$\sum_{k=1}^{K} [D(Q_k \| G_{m_k}) + H(Q_k)].$$

We consider the following models.

1. $K$ objects are different, they have different distributions among $M \geq K$ possibilities. The identification problem in the formulations of books [1] and [2] is considered. We restrict ourselves to the case $K = 2,\ M = 2$.
2. $K$ objects are independent, that is, some of them may have the same distributions. It is surprising that this model was not considered earlier in the literature. We examine here an example for $K, M = 2$.
3. We investigate one object, $K = 1$, and $M(\geq 2)$ possible probability distributions. The question is whether or not the $m$-th distribution occurred. This is the problem of distribution identification in the spirit of paper [3].
4. Ranking, or ordering problem [4]. We have one vector of observations $\mathbf{X} = (X_1, X_2, \ldots X_N)$ and $M$ hypothetic distributions. The receiver wants to know whether the index of the true distribution of the object is in $\{1, 2, ..., r\}$ or in $\{r+1, ..., M\}$.
5. $S$-identification of distribution [4]. Again $K = 1$. One wants to identify the observed object as a member of the subset $S$ of $[M]$, or of its complement.

## 3. Background

The study of interdependence of exponential rates of decrease, as the sample size $N$ goes to infinity, of the error probabilities $\alpha_{1|2}^{(N)}$ of the "first type" and $\alpha_{2|1}^{(N)}$ of the "second type" was initiated by the works of Hoeffding [5], Csiszár and Longo [6], Tusnády [7], Longo and Sgarro [8], Birgé [9] and Haroutunian [10].

A similar problem for Markov dependence of experiments was investigated by Natarajan [13], Haroutunian [14], Gutman [15] and others. In [16,17] Blahut developed an application of the methods of hypothesis testing to the proper problems of information theory.

It will be very interesting to combine these problems with the approach initiated by the paper of Ahlswede and Csiszár [18] and developed by many authors, particularly, for the exponential error probabilities by Han and Kobayashi [19].

Ahlswede and Burnashev [20] studied a model of an estimation system with compressed information. A similar problem was examined by Zhang and Berger [21]. In the paper of Ahlswede, Yang and Zhang [22] identification in channels via compressed data was considered.

Further considerations will be based on the results from [10] on multiple hypotheses testing. We will now briefly describe corresponding formulations and proofs. In our terms, it is the case of one object $(K = 1)$ and $M$ possible distributions (hypotheses) $G_1, \ldots G_M$. A test $\varphi(\mathbf{x})$ on the base of $N$-sample $\mathbf{x} = (x_1, \ldots x_N)$ suggest a decision about the distribution. Since experiments are independent, the probability of the sample $\mathbf{x}$, if the distribution is $G_m$, will be

$$G_m^{(N)}(\mathbf{x}) = \prod_{n=1}^{N} G_m(x_n), \quad m \in [M].$$

We study error probabilities $\alpha_{m|l}^{(N)}$ for all $m, l \in [M]$. $\alpha_{m|l}^{(N)}$ is the probability that instead of the true distribution $G_m$ the distribution $G_l$ was accepted.

For $m = l$ as in (1) we denote by $\alpha_{m|m}^{(N)}$ the probability of rejecting $G_m$ when it is true, hence

$$\alpha_{m|m}^{(N)} = \sum_{l: l \neq m} \alpha_{m|l}^{(N)}$$

This probability is referred to in [23] as the test's "error probability of type $m$." The matrix $\{\alpha_{m|l}^{(N)}\}$ is sometimes referred to as the "power of the test."

We assume that the list of possible hypotheses is complete. Note that the case of the objects having some unknown distributions different from $G_1, \ldots G_M$ is also of particular interest [24].

Let us define the reliability matrix with components

$$E_{m|l} = \overline{\lim_{N \to \infty}} -\frac{1}{N} \log \alpha_{m|l}^{(N)}, \quad m, l \in [M]$$

According to this definition we can derive that

$$E_{m|m} = \min_{l: m \neq l} E_{m|l} \tag{4}$$

In the case where $K = 2$, the reliability matrix is

$$\begin{pmatrix} E_{1|1} & E_{1|2} \\ E_{2|1} & E_{2|2} \end{pmatrix} \tag{5}$$

and from (4) it follows that there are only two different parameters, namely

$$E_{1|1} = E_{1|2} \text{ and } E_{2|1} = E_{2|2} \tag{6}$$

so in this case the problem is to find the maximal value of one when the value of the other is given.

For given positive and finite $E_{1|1}, \ldots, E_{M-1,M-1}$ let us consider the regions

$$R_l = \{Q : D(Q \| G_l) \leq E_{l|l}\}, \quad l \in [M-1], \tag{7a}$$

$$R_M = \{Q : D(Q \| G_l) > E_{l|l},\ \ l \in [M-1]\}, \tag{7b}$$

$$R_l^{(N)} = R_l \bigcap P^{(N)}, \quad l \in [M]. \tag{7c}$$

Let

$$E_{l|l}^{(*)} = E_{l|l}^{(*)}(E_{l|l}) = E_{l|l},\ \ l \in [M-1], \tag{8a}$$

$$E_{m|l}^{*} = E_{m|l}^{*}(E_{l|l}) = \inf_{Q \in R_l} D(Q \| G_m),$$

$$m \in [M],\ \ m \neq l,\ \ l \in [M-1], \tag{8b}$$

$$E_{m|M}^{*} = E_{m|M}^{*}(E_{1|1}, ..., E_{M-1,M-1}) =$$

$$= \inf_{Q \in R_M} D(Q \| G_m),\ \ m \in [M-1], \tag{8c}$$

$$E_{M|M}^{*} = E_{M|M}^{*}(E_{1|1}, ..., E_{M-1,M-1}) =$$

$$= \min_{l \in [M-1]} E_{M|l}^{*}. \tag{8d}$$

If some distribution $G_m$ is not absolutely continuous relative to $G_l$ then the reliability $E_{m|l}^{*}$ is equal to infinity.

The principal result of [10] is

Theorem 1: If all distributions $G_m$ are different, two statements hold:

a) when the positive numbers $E_{1|1}, ..., E_{M-1,M-1}$ satisfy conditions

$$E_{1|1} < \min_{l=2,...,M} D(G_l \| G_1),$$

$$\dots \tag{9}$$

$$E_{m|m} \leq \min_{l \in [m-1]} E_{m|l}^{*}(E_{l|l}), \text{ and } E_{m|m} < \min_{l=m+1,...,M} D(G_l \| G_m),$$

$$m \in [2, M-1],$$

there exists a LAO sequence of tests, in which each reliability matrix $E^* = \{E^*_{m|l}\}$ is defined as in (8) and all elements of it are positive;

b) even if one of the conditions (9) is violated, then the reliability matrix of any such test has at least one element equal to zero (that is, the corresponding error probability does not tend to zero exponentially).

The essence of the proof of Theorem 1 is the construction of the optimal test sequence.

For the simplest particular case of $M = 2$, elements of the reliability matrix (5) satisfy equalities (6) and for a given $E_{1|1}$ from (7.a) and (8.b) we obtain the value of $E^*_{2|1} = E^*_{2|2}$:

$$E^*_{2|1}(E_{1|1}) = \inf_{Q: D(Q\|G_1) \le E_{1|1}} D(Q \,\|\, G_2). \tag{10}$$

Here, according to (9) we can presume that $0 < E_{1|1} < D(G_2 \,\|\, G_1)$.

## 4. Identification Problem for Models with Independent Objects

This section considers the model, in which $K$ objects are independent and some of them may have the same distributions (Model 2). To illustrate related issues and essential features of this model we consider a few simple cases. It is clear that the case with $M = 1$ is trivial. Let us consider the case $K = 2,\ M = 2$.

The reliability matrix is

$$\begin{pmatrix} E_{1,1|1,1} & E_{1,1|1,2} & E_{1,1|2,1} & E_{1,1|2,2} \\ E_{1,2|1,1} & E_{1,2|1,2} & E_{1,2|2,1} & E_{1,2|2,2} \\ E_{2,1|1,1} & E_{2,1|1,2} & E_{2,1|2,1} & E_{2,1|2,2} \\ E_{2,2|1,1} & E_{2,2|1,2} & E_{2,2|2,1} & E_{2,2|2,2} \end{pmatrix}. \tag{11}$$

Let us denote by $\alpha^{(N,1)}_{m_1|l_1}$, $\alpha^{(N,2)}_{m_2|l_2}$ and $E^{(1)}_{m_1|l_1}$, $E^{(2)}_{m_2|l_2}$ the error probabilities and the reliabilities of the first and second objects, respectively.

Lemma: The following is true:

$$E_{m_1,m_2|l_1,l_2} = E^{(1)}_{m_1|l_1} + E^{(2)}_{m_2|l_2}, \quad \text{if } m_1 \ne l_1 \text{ and } m_2 \ne l_2, \tag{12a}$$

$$E_{m_1,m_2|l_1,l_2} = E^{(i)}_{m_i|l_i}, \quad \text{if } m_i \ne l_i,\ m_{3-i} = l_{3-i},\ i = 1,2, \tag{12b}$$

$$E_{m_1,m_2|m_1,m_2} = \min(E^{(1)}_{m_1|m_1}, E^{(2)}_{m_2|m_2}). \tag{12c}$$

Proof: (12) is a consequence of definition (2) and of the following equality resulting from independence of the objects

$$\alpha^{(N)}_{m_1,m_2|l_1,l_2} = \alpha^{(N,1)}_{m_1|l_1}\alpha^{(N,2)}_{m_2|l_2}, \quad \text{if} \quad m_1 \neq l_1 \ \text{ and } \ m_2 \neq l_2, \tag{13a}$$

$$\alpha^{(N)}_{m_1,m_2|l_1,l_2} = \alpha^{(N,i)}_{m_i|l_i}, \quad m_i \neq l_i,\ m_{3-i} = l_{3-i}, \quad i = 1,2, \tag{13b}$$

Using (12) we can find all elements of the matrix (11).
If we denote

$$E^{(1)}_{1|1} = E^{(1)}_{1|2} = a_1, \quad E^{(1)}_{2|1} = E^{(1)}_{2|2} = b_1,$$

$$E^{(2)}_{1|1} = E^{(2)}_{1|2} = a_2, \quad E^{(2)}_{2|1} = E^{(2)}_{2|2} = b_2,$$

we can see that the matrix (11) is equal to the matrix

$$\begin{pmatrix} \min(a_1,a_2) & a_2 & a_1 & a_2+a_1 \\ b_2 & \min(a_1,b_2) & a_1+b_2 & a_1 \\ b_1 & b_1+a_2 & \min(a_2,b_1) & a_2 \\ b_1+b_2 & b_1 & b_2 & \min(b_1,b_2) \end{pmatrix} \tag{14}$$

(14) can be calculated by using only two numbers $a_1$ and $a_2$, since $b_1$ and $b_2$ are defined in formula (10).

If $E^{(1)}_{1|1} = E^{(2)}_{1|1}$ then $a_1 = a_2 = a$ and only 5 different values $a$, $b$, $2a$, $a+b$, $2b$ define the matrix in (14).

## 5. Identification Problem for Models with Different Objects

The $K$ objects are not independent, they have different distributions, thus the number $M$ of distribution is not less than $K$.

As an example we consider the case of $K = 2, \ M = 2$. The matrix of reliabilities is as follows:

$$\begin{pmatrix} E_{1,2|1,2} & E_{1,2|2,1} \\ E_{2,1|1,2} & E_{2,1|2,1} \end{pmatrix}. \tag{15}$$

Since the objects are strictly dependent (when distribution of one of the objects is known, then the distribution of the second is also known), this matrix coincides with the reliability matrix of the first object (see (5))

$$\begin{pmatrix} E_{1|1}^{(1)} & E_{1|2}^{(1)} \\ E_{2|1}^{(1)} & E_{2|2}^{(1)} \end{pmatrix},$$

because the distribution of the second object is uniquely defined by the distribution of the first.

We can conclude that among 4 elements of the matrix (15) only 2 are distinct, which are defined by given $E_{1|1}^{(1)}$.

## 6. Identification of Distribution of an Object

Let us have one object, $K=1$ with one of $M \geq 2$ possible distributions. The question is whether this distribution is the $m$ th one. There are two types of error probabilities for each $m \in [M]$: the probability $\alpha_{m|(l\neq m)}^{(N)}$ of accepting distribution $l$ different from $m$ when $m$ is the true probability and the probability $\alpha_{(l\neq m)|m}^{(N)}$ that $m$ is accepted when it is not the .

The probability $\alpha_{m|(l\neq m)}^{(N)}$ is already known; it coincides with the probability $\alpha_{m|m}^{(N)}$ which is equal to $\sum_{l:l\neq m} \alpha_{m|l}^{(N)}$.

The corresponding reliability $E_{m|(l\neq m)}$ is equal to $E_{m|m}$ (see (4)).

We have to determine dependence of $E_{(l\neq m)|m}$ of $E_{m|(l\neq m)} = E_{m|m}$. The value of $E_{m|m}$ should satisfy conditions in (9) and therefore $E_{m|m} \leq \min_{l:l\neq m} D(G_l \,||\, G_m)$.

We need to ascertain the probabilities of different hypotheses. Without prior knowledge it is natural to assume that the hypotheses $G_1, \ldots, G_M$ are equiprobable, i.e., $\mathbf{Pr}(m) = 1/M, \;\; m \in [M]$.

Thus the following can be deduced:

$$\alpha_{(l\neq m)|m}^{(N)} = \frac{\mathbf{Pr}(l \neq m, m)}{\mathbf{Pr}(l \neq m)} = \frac{\mathbf{Pr}(l \neq m, m)}{(M-1)/M} =$$

$$= \frac{M}{(M-1)} \sum_{l\neq m} \mathbf{Pr}(l, m) = \frac{1}{M-1} \sum_{l\neq m} \alpha_{l|m}^{(N)}.$$

From here we can see that

$$E_{(l\neq m)|m} = \overline{\lim_{N\to\infty}} -\frac{1}{N} \log \alpha_{(l\neq m)|l}^{(N)} =$$

$$= \overline{\lim_{N\to\infty}} \frac{1}{N}\left( -\log \sum_{l\neq m} \alpha_{l|m}^{(N)} + \log(M-1) \right) = \min_{l\leq m} E_{l|m}. \tag{16}$$

From (16) using a formula similar to (10) we conclude that

$$E_{(l\neq m)|m} = \min_{l\neq m} E_{l|m} = \min_{l\neq m} \inf_{Q\in R_m} D(Q\,\|\,G_l) =$$

$$= \min_{l\neq m} \inf_{Q:D(Q\|G_m)\neq E_{m|m}} D(Q\,\|\,G_l). \tag{17}$$

This result can be summarized in Theorem 2.

Theorem 2: For the model with different objects, for a given sample $\mathbf{x}$ and its type $Q$ we accept hypothesis $m$ when $Q\in R_m$. Under condition that hypotheses are equiprobable the reliabilities of such a test are $E_{m|(l\neq m)}$ and $E_{(l\neq m)|m}$, and they are related as in (17).

## 7. *S*-Identification and Ranking Problems

This section considers the $S$-identification model [4], which has been introduced in [4] as K-identification. Given $N$-sample $\mathbf{x}$ of measurements of the object, the problem is whether a sample is in the part $S$ of $M$ possible distributions or in the complement of $S$.

As in the last case, we can make a decision based on the type $Q$ of the sample. Again we suppose that *a priori* all hypotheses are equiprobable:

$$\mathbf{Pr}(m) = 1/M, \quad m\in [M]. \tag{18}$$

When $Q\in \bigcup_{m\in S} R_m^{(N)}$ the decision is: "$m$ is in $S$."

The ranking model is a particular case of the model of $S$-identification with $S=\{1,2,...,r\}$. Conversely, the $S$-identification problem without loss of generality may be considered as the ranking problem. Let us renumerate the hypotheses, placing the hypotheses of $S$ in the first $r$ places. Because these two models are mathematically equivalent we shall speak below only of the ranking model.

It is sufficient to consider the cases $2\leq r\leq \lceil M/2 \rceil$, because if $r$ is larger we can replace $S$ with its complement while the case $r=1$ was considered in Section 6.

We study two error probabilities of a test: the probability $\alpha^{(N)}_{m\le r|l>r}$ of an incorrect decision when $m$ is not greater than $r$ and the probability $\alpha^{(N)}_{m>r|l\le r}$ of an error when $m$ is greater than $r$. The corresponding reliabilities are

$$E_1(r) = E_{m\le r|l>r} \quad \text{and} \quad E_2(r) = E_{m>r|l\le r}, \ 2 \le r \le \lceil M/2 \rceil \tag{19}$$

With assumption (18) we have

$$\alpha^{(N)}_{m\le r|l>r} = \frac{\mathbf{Pr}^{(N)}(m \le r, l > r)}{\mathbf{Pr}(m \le r)} =$$

$$= \frac{M}{r} \sum_{m\le r} \sum_{l>r} \mathbf{Pr}^{(N)}(m,l) = \frac{1}{r} \sum_{m\le r} \sum_{l>r} \alpha^{(N)}_{m|l}. \tag{20}$$

From the definition (19) of $E_1(r)$ and the equality (20) we have

$$E_1(r) = \overline{\lim_{N\to\infty}} -\frac{1}{N} \log \alpha^{(N)}_{m\le r|l>r} =$$

$$= \overline{\lim_{N\to\infty}} -\frac{1}{N} \log \left[ \sum_{m\le r} \sum_{l>r} \alpha^{(N)}_{m|l} - \log r \right] = \min_{m\le r, l>r} E_{m|l}. \tag{21}$$

Analogically, at the same time,

$$E_2(r) = \overline{\lim_{N\to\infty}} -\frac{1}{N} \log \alpha^{(N)}_{m>r|l\le r} =$$

$$= \overline{\lim_{N\to\infty}} -\frac{1}{N} \log \left[ \sum_{m>r} \sum_{l\le r} \alpha^{(N)}_{m|l} - \log(M-r) \right] = \min_{m>r, l\le r} E_{m|l}. \tag{22}$$

For any test the value of $E_1(r)$ must satisfy the conditions of (21):

$$E_1(r) \le \min_{m\le r} E_{m|m},$$

or it is sufficient that $E_1(r)$ satisfy inequalities from (9) for $m \le r$. Thus for any such test the reliability $E_2(r)$ may be calculated by the equality (22). The best $E_2(r)$ is obtained if we are given the liberty to select the biggest values for reliabilities $E_{m|m}$,

$r < m \le M-1$, of the test satisfying those $m-s$ conditions (9). This reasoning may be summarized in the following theorem.

Theorem 3: When the hypotheses are equiprobable for $E_1(r)$ satisfying for $m \le r$ the inequalities (9), $E_2(r)$ may be calculated by the following expression

$$E_2(r) = \max_{\{E_{m|l},\, m,l\in[M]\}:\ \min\limits_{m\le r,\ l>r} E_{m|l}=E_1(r)} \left[\min_{m>r,\, l\le r} E_{m|l}\right],$$

$$2 \le r \le \lceil M/2 \rceil.$$

## 8. Future Work

The presented problems and results may be extended in different directions, some of which were already noted. It is necessary to examine models when measurements are described by more general classes of random variables and processes [13–15,25]. One possible direction od research is related to the use of compressed data on measurements [19–21]. It is possible to see wide perspectives for application of identification approaches and methods in authentication theory [26].

## Acknowledgements

This lecture surveyed the results of the work produced jointly by the author and Professor Rudolf Ahlswede from Bielefeld University, Germany. The problem statement was also discussed with Michael Malutov to whom the author expresses his gratitude.

## References

[1] Bechhofer R.E., Kiefer J., and Sobel M., Sequential identification and ranking procedures. The University of Chicago Press, Chicago, 1968.
[2] Ahlswede R. and Wegener I., Search problems. Wiley, New York, 1987.
[3] Ahlswede R. and Dueck G., Identification via channels. IEEE Trans. Inform. Theory, vol. 35, no. 1, pp. 15–29, 1989.
[4] Ahlswede R., General theory of information transfer. Preprint 97-118, SFB., Discrete Structures in der Mathematik, Universität Bielefeld.
[5] Hoeffding W., Asymptotically optimal tests for multinomial distributions. Annals. of Math. Statist., vol. 36, pp. 369–401, 1965.
[6] Csiszár I. and Longo G., On the error exponent for source coding and for testing simple statistical hypotheses. Studia Sc. Math. Hungarica, vol. 6, pp. 181–191, 1971.
[7] Tusnady G., On asymptotically optimal tests. Annals of Statist., vol. 5, no. 2, pp. 385–393, 1977.
[8] Longo G. and Sgarro A., The Error exponent for the testing of simple statistical hypotheses, a combinatorial approach. Journal of Combin., Inform. Sys. Sc., vol. 5, No. 1, pp. 58–67, 1980.
[9] Birgé L., Vitesse maximales de décroissance des erreurs et tests optimaux associés, Z. Wahrsch. verw Gebiete, vol. 55, pp. 261–273, 1981.
[10] Haroutunian E.A., Logarithmically asymptotically optimal testing of multiple statistical hypotheses. Problems of Control and Inform. Theory. vol. 19, no. 5–6, pp. 413–421, 1990.
[11] Csiszár and Körner. Information theory: Coding theorems for discrete memoryless systems. Academic Press, New York, 1981.

[12] Csiszár I., Method of types. IEEE Trans. Inform. Theory, vol. 44, no. 6, pp. 2505–2523, 1998.
[13] Natarajan S., Large deviations, hypotheses testing, and source coding for finite Markov chains. IEEE Trans. Inform. Theory, vol. 31, no. 3, pp. 360–365, 1985.
[14] Haroutunian E.A., On asymptotically optimal testing of hypotheses concerning Markov chain (in Russian), Izvestia Acad. Nauk Armenian SSR. Seria Mathem. vol. 22, No. 1, pp. 76–80, 1988.
[15] Gutman M., Asymptotically optimal classification for multiple test with empirically observed statistics. IEEE Trans. Inform. Theory, vol. 35, No 2, pp. 401–408, 1989.
[16] Blahut R.E., Principles and practice of information theory. Addison-Weslay, Massachusets, 1987.
[17] Blahut R., Hypotheses testing and information theory, IEEE Trans. Inform Theory, vol. 20, no. 4, pp. 405–417, 1974.
[18] Ahlswede R. and Csiszár I., Hypotheses testing with communication constraints. IEEE Trans. Inform. Theory vol. 32, no. 4, pp. 533–542, 1986.
[19] Han T.S. and Kobayashi K., Exponential-type error probabilities for multiterminal hypothesis testing. IEEE Trans. Inform. Theory, vol. 35, no. 1, pp. 2–13, 1989.
[20] Ahlswede R. and Burnashev M., On minimax estimation in the presence of side information about remote data. Annals of Statist., vol. 18, no. 1, pp. 141–171, 1990.
[21] Zhang Z. and Berger T., Estimation via compressed information. IEEE Trans. Inform. Theory, vol. 34, no. 2, pp. 198–211, 1988.
[22] Ahlswede R., Yang E., Zhang Z., Identification via compressed data. IEEE Trans. Inform. Theory, vol. 43, no. 1, pp. 48–70, 1997.
[23] Borovkov A.A., Mathematical statistics (in Russian). Nauka, Novosibirsk, 1997.
[24] Rao C.R., Linear statistical inference and its applications. Wiley, New York, 1965.
[25] Han T.S., Hypothesis testing with the general source. IEEE Trans. Inform. Theory, vol. 46, no. 7, pp. 2415–2427, 2000.
[26] Maurer U.M., Authentication theory and hypothesis testing. IEEE Trans. Inform. Theory, vol. 46, no. 4, pp. 1350–1356, 2000.

This page intentionally left blank

# 2. HUMAN COMPUTER INTERACTION

This page intentionally left blank

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Decision Support in Command and Control

## *A Balanced Human-Technological Perspective*

Éloi BOSSÉ, Stéphane PARADIS and Richard BRETON
*Defence Research and Development Canada Valcartier*

**Abstract.** Command and control can be characterized as a dynamic human decision making process. A technological perspective of Command and control has led system designers to propose solutions such as decision support and information fusion to overcome many of the domain problems. This and the lack of knowledge in cognitive engineering have in the past jeopardized the design of helpful computerized aids aimed at complementing and supporting human cognitive tasks. Moreover, this lack of knowledge has most of the time created new trust problems in designed tools, and human in the loop concerns. Solving the command and control problem requires balancing the human factor perspective with that of the system designer and coordinating the efforts in designing a cognitively fitted system to support decision-makers. This paper discusses critical issues in the design of computer aids by which the decision-maker can better understand the situation in his area of operations, select a course of action, issue intent and orders, monitor the execution of operations and evaluate the results. These aids will support decision-makers to cope with uncertainty and disorder in warfare and to exploit people or technology at critical times and places so as to ensure success in operations.

**Keywords.** Command and control, data fusion, situation awareness, decision making, cognitive systems engineering

### 1. Introduction

Command and Control (C2) is defined, by the military community, as the process by which a commanding officer can plan, direct, control and monitor any operation for which he is responsible in order to fulfill his mission. A new definition has been proposed [1] describing C2 as a dynamic human decision making process that establishes common intent and transforms that common intent into a coordinated action.

From a human factors perspective, the complexity of military operations highlights the critical role of human leadership in C2. To resolve adversity, C2 systems require qualities inherent to humans such as decision making abilities, initiative, creativity and the notion of responsibility and accountability. Although these qualities are essential, characteristics inherent to the environment in which C2 occurs, combined with advancements in threat technology, significantly challenge the accomplishment of this process and therefore require the support of technology to complement human capabilities and limitations.

A technological perspective of C2 has led system designers to propose solutions by providing operators with decision support systems (DSS). These DSSs should aid operators to achieve the appropriate Situation Awareness (SA) state for their decision making activities, and to support the execution of resulting actions. The lack of knowledge in cognitive engineering has in the past jeopardized the design of helpful com-

puter based aids aimed at complementing and supporting human cognitive tasks. Moreover, this lack of knowledge has most of the time created new trust problems in the designed tools.

Solving the C2 problem thus requires balancing the human factor perspective with that of the system designer and coordinating the efforts in designing a cognitively fitted system to support decision-makers. The paper starts with a discussion on the C2 and decision-making process followed by the decision support definitions and concepts. Then, the problem of designing a cognitively fitted DSS using the Cognitive Systems Engineering (CSE) approach is presented.

## 2. Command and Control and Decision Making

Modern military operations take place within an enormously complex environment to accomplish missions across the spectrum of conflict from humanitarian assistance to high intensity combat. In the past several decades, the battlespace has expanded enormously in the face of increasingly potent and accurate weapons capable of being launched at progressively further ranges from their targets. Concomitantly, the pace and scope of operations have continued to increase with each successive advance in warfare. In response to these challenges, powerful new sensors have been deployed at sea, ashore and in space, while the capacity of communications systems has multiplied to make huge volumes of data and information available to commanders and their staffs. In short, technological improvements in mobility, range, lethality and information acquisition continue to compress time and space, forcing higher operating tempos and creating greater demands on command decision-making.

Command and control (C2) is the means by which decision-makers synchronize military actions in time, space, and purpose to achieve unity of effort within a military force. C2 is shaped by two factors – uncertainty and time – that dominate the environment in which military decisions are made. The "OODA Loop" (Observe Orient Decide and Act) (fig. 1) is a useful model to represent the decision cycle that lies at the heart of C2. The OODA loop hinges on the fulfillment of two broad functions: first, that all commanders within a force arrive at a shared and consistent understanding of the battlespace arising through **battlespace awareness**; and, second, that **unity of effort** is achieved throughout a joint and combined force through commonly held intent. Within the R&D community we often refer to **Situation Awareness** and **Decision Making**.

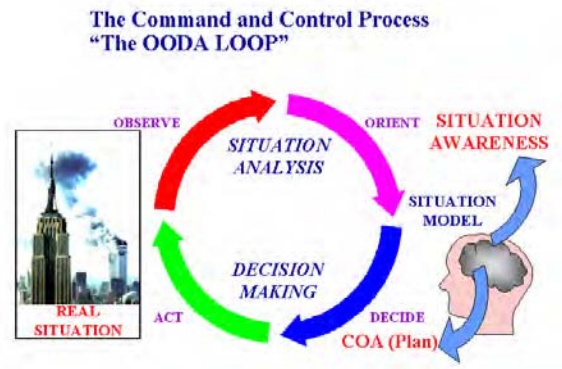



**Figure 1.** The C2 process represented by the OODA loop.

**Situation Awareness** is for decision-makers to understand the current situation and to fully grasp all of the implications associated with each of several possible courses of action (COA). Achieving this is an inherently complex activity. As a process, it begins by rendering, by both automated and manual methods (Data Fusion), huge amounts of raw data into information and displaying it by various means for the commander's use. However, it ends in the mind of the commander himself and involves visualizing the current state of friendly and enemy forces, and the future force relationships that must exist to accomplish the mission.

**Decision making** is for the higher commander to achieve unity of effort in the execution of operations throughout the battlespace. This concept is more comprehensive than the simple verbal or written expression of a commander's decision. It begins with the commander's vision of how the desired end-state is to be achieved – his intent – and how that vision is interpreted and acted upon throughout a force.

### *2.1. Situation Awareness*

The C2 process is seen as an instantiation or an example of a dynamic human decision making process that establishes the common intent and transforms that common intent into a coordinated action. The first half of Boyd's loop (Observe-Orient) gathers a number of processes that mainly perceive, interpret and project the status of the entities included in the C2 environment. Yielding from these processes is the situation awareness required to complete the decision-making process. The latter process corresponds to the second half (Decide-Act) of the OODA loop. Given the tactical situation and the available onboard resources, it decides on the best course of action with respect to own ship mission and supports its implementation.

Figure 2 illustrates a theoretical model derived by Endsley of Situation Awareness (SA) based on its role in dynamic human decision making. SA is defined [2] as the perception of the elements in the environment, within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.

The first level of SA yields, in the *perception* of the status, attributes and dynamics of relevant elements in the environment. Endsley describes the comprehension process as follows: "Comprehension of the situation is based on a synthesis of disjoint level 1 elements." Level 2 of SA goes beyond simply being aware of the elements that are present, to include an understanding of the significance of those elements in light of pertinent operator goals. Based on knowledge of Level 1 elements, particularly when some elements are put together to form patterns with other elements, the decision-maker forms a holistic picture of the environment, comprehending the significance of objects and events. The third and last step in achieving situation awareness is the *projection* of the future actions of the elements in the environment. This is achieved through knowledge of the status and dynamics of the perceived and comprehended situation elements.

The Situation Awareness processes described by Endsley are initiated by the presence of an object in the perceiver's environment. However, processes related to situation awareness can also be triggered by a priori knowledge, feelings or intuitions. In these situations, the picture is understandable, and projections in the future are possible, if any event, which has not been perceived at this time, can be found in the environment. Hence, hypotheses related to the possible presence of an object are formulated. The perceiver then initiates search processes in the environment that confirm or invalidate these hypotheses. Note that this type of SA is possible only if mental models related to the possible objects are available.

If one compares the OODA loop with the SA model of Endsley, one sees a close resemblance. In both models one finds a *decision-making* part and an *action* part. In Endsley's model, SA is one of the main inputs for decision-making. In the OODA loop, the processes Observe and Orient provide inputs for the decision making process. One should recall, however, that situation awareness in Endsley's model is a *state of knowledge* and not a process.

In her theory of SA, Endsley clearly presumes *patterns* and *higher level elements* to be present according to which the situation can be structured and expressed. SA can be interpreted as the operator's mental model of all pertinent aspects of the environment (processes, states, and relationships).

There is a tight link between this mental model used to structure and express situation elements and the cognitive processes involved in achieving the levels of awareness. This link is known as the cognitive fit and requires an understanding of how the human perceives a task, what processes are involved, what are the human needs and what part of the task can be automated or supported. This understanding is crucial and only achieved via a number of specialized human factor investigations known as cognitive engineering analyses.

Cognitive engineering analyses are generally conducted by the human factor engineering community. According to Preece [3], cognitive ergonomics is a discipline that focuses particularly on human information processing and computer systems. By definition, it aims to develop knowledge about the interaction between human information processing capacities and limitations, and technological information processing systems.

The usefulness of a system is closely related to its compatibility with human information processing. Thus, such a system must be developed according to human information processing and human needs. A first step is the identification of the cognitive processes involved in the execution of the task. Many procedures have been developed to identify those processes. Jonassen, Hannum and Tessmer [4] describe task analysis as a process that is performed in many ways, in a variety of situations, and for multiple purposes. This analysis determines what the performers do, how they perform the task, how they think or how they apply a skill.

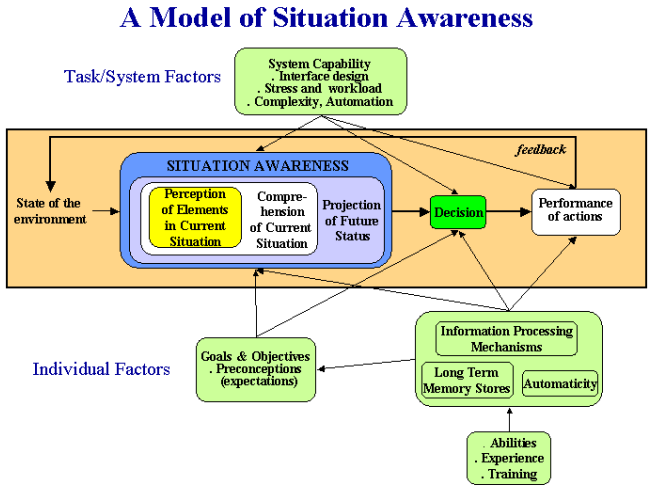



**Figure 2.** Endsley's situation awareness model.

Among the procedures developed to identify cognitive processes, there are the Cognitive Task Analysis (CTA) and the Cognitive Work Analysis (CWA). There are only subtle and ambiguous differences between these two procedures. Moreover, their labels are frequently used in an interchangeable manner in the literature. However, the CWA can be seen as a broader analysis than the CTA. According to Vicente [5], traditional task analysis methods typically result in a single temporal sequence of overt behavior. This description represents the normative way to perform the task. However, traditional methods cannot account for factors like changes in initial conditions, unpredictable disturbances and the use of multiple strategies. The use of the traditional task analysis brings an artifact that will only support one way to perform the task.

### *2.2. Decision Making*

The aim of C2 is to allow the commander to make decisions and take actions faster and better than any potential adversary. Accordingly, it is essential to understanding how commanders make decisions.

One stream of decision-making theory, based on decision theoretic paradigms, views decision making as an analytic process that corresponds closely to the military estimate of the situation. According to this analytic approach, the commander generates several options, then identifies criteria for evaluating these options, rates the options against these criteria and then selects the best option as the basis for future plans and action. It aims at finding the optimal solution, but it is time consuming and information intensive.

A second stream of decision making theory emphasizes a more inductive than analytic approach. Called naturalistic decision making (NDM), it emphasizes the acquisition of knowledge, the development of expertise and the ability of humans to generalize from past experience. It stresses pattern recognition, creativity, experience and initiative.

Intuitive decision making relies on the commander's ability to recognize the key elements of a problem based on experience and judgment. This approach focuses on situation assessment and strives to find the first solution that solves the problem. The intuitive model works on the assumption that, by drawing upon personal experience and judgment, the commander will generate the first workable solution. If time permits, the commander may evaluate his decision in a more analytic way; if he finds it defective, he moves on to the next reasonable solution.

The analytic approach has wide military applicability when time is not a constraint and extensive information may be gathered. It is appropriate for contingency planning and in preparing for operations. It may also have merit in situations falling outside the scope of the commander's previous experience. Having said this, however, the intuitive approach appears to be more applicable for a majority of typical tactical or operational decisions – decisions made in highly fluid and dynamic conditions of combat, when time and uncertainty are the dominant factors affecting C2.

In general, while the two models represent conceptually distinct approaches to decision making, they are not mutually exclusive in practice. The commander will adopt the approach that is best tailored to the situation and may use elements of the two at the same time. Indeed, a combination of the two is probably always together at work within the C2 system.

Central to the decision cycle is establishing intent: the fundamental activity of determining what to do and how to propagate it among subordinates. The concept is

broader and more comprehensive than simple decision-making. It requires a creative act whose purpose is to bound an infinitely large space of possible actions into a finite number of precise, focused objectives. Intent embodies a human commander's vision and will, and is inevitably the product of history, expertise and circumstance. The combination of explicit intent for an operation plus the relevant portions of implicit intent that are shared by subordinates and the commander makes up the "common intent" for that operation.

In striking a balance between the need to coordinate the actions of the force and providing subordinate commanders with the freedom to conduct operations, a commander will need to consider a number of factors. These include, among others: the nature of the operation itself; the level of experience and teamwork achieved by the force; the strategic consequences of combat action; the environment; and the nature and capabilities of the adversary.

### *2.3. Task/Human/Technology Triad Model*

A triad approach has been proposed by Breton, Rousseau and Price [6] to represent the collaboration between systems designers and human factors specialists. As illustrated in Figure 3, three elements compose the triad: the task, the technology and the human. In the C2 context, the OODA loop represents the task to be accomplished. The design process must start with the identification of environmental constraints and possibilities by subject-matter experts within the context of a Cognitive System Engineering approach.

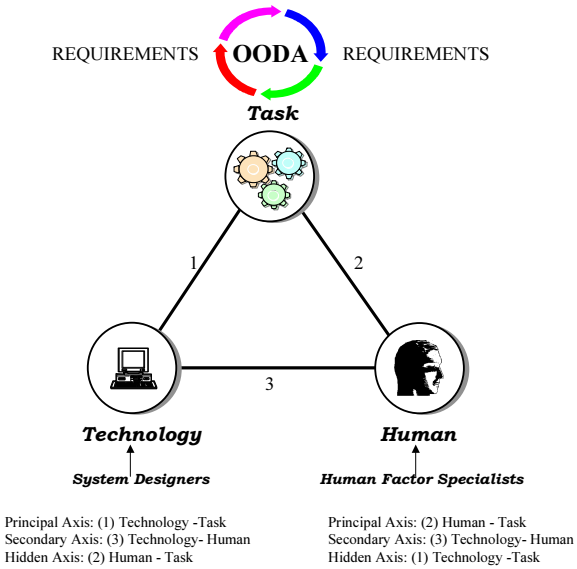



**Figure 3.** Task/human/technology triad model.

Systems designers are introduced via the technology element. Their main axis of interest is the link between the technology and the task. The general question related to this link is: "What systems must be designed to accomplish the task?" Systems designers are also considering the human; their secondary axis of interest is thus the link be-

tween the technology and the human. The main question of this link is: "How must the system be designed to fit with the human?" However, systems designers have a hidden axis. The axis between the human and the task is usually not covered by their expertise. From their analyses, technological possibilities and limitations are identified. However, all environmental constraints may not be covered by technological possibilities. These uncovered constraints, named thereafter deficiencies, are then addressed as statements of requirements to the human factor community (see Fig. 4). These requirements lead to better training programs, the reorganization of work and the need for leadership, team communication, etc.

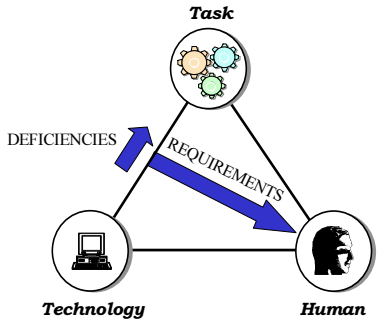



**Figure 4.** Human requirements.

Human factor specialists are introduced via the human element of the triad. Their main axis is the link between the human and the task, which is the hidden axis of systems designers. Using a Cognitive System Engineering approach, they identify how humans perceive the task, what they have to do to accomplish the task, what strategies and resources are involved and what are the shortfalls and human limitations. Their secondary axis of interest is the same as that of the system designers (i.e., human-technology), and their hidden axis is the link between the technology and the task, which is the main axis of the system designers. From their analyses, human possibilities and limitations are identified. However, all environmental constraints may not be covered by human possibilities and resources. The uncovered deficiencies are then addressed as statements of requirements to the technological community (see Fig. 5). These statements become the specification of which part of the task needs support or must be automated, what the system must do, in which conditions, and how the system must interact with the operator.

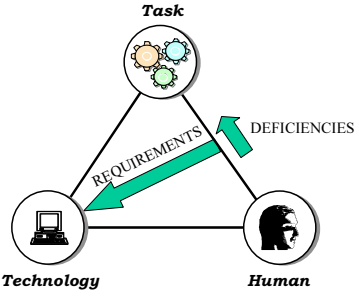



**Figure 5.** Technology requirements.

In this context, everyone involved in the design process has its own field of intervention. The weakness of one is the strength of the other. The sets of statements of requirements produced by the systems designers and human factor specialists are analyzed within a multi-disciplinary team involving both communities. This analysis leads to one set of consolidated requirements that determines the nature of the solution (see Fig. 6). It is very important that both types of specialists work in a close collaboration. Working in isolation would bring unrealizable requirements formulated by one part to the other.

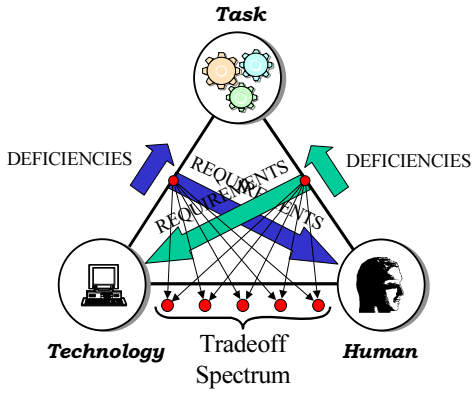



**Figure 6.** Requirements tradeoff spectrum.

Within the context of a war or tactical operations, unpredictable events are expected more frequently and are caused mainly by intelligent sources. The inductive capacity of humans is then required to deal with these events. Some part of the system can be automated, but the system must be mostly designed to support the human in its activities. Hence, the solution cannot be found from a complete technological perspective or a complete human perspective. It must rather be a mixture of both.

Automation has changed the nature of the implication of the human. With automated systems, the human role is mainly related to the supervision of the situation. As mentioned earlier, this new role brings new problems and issues to be considered. In particular, this situation raises the question as to which part has the authority. There is no general answer to this question. A proposed approach is to delegate authority according to the situation. Chalmers [7] proposes five modes of operator-system delegation. The human selects the mode, which applies until mode transition is triggered by a new selection. It is obvious that a good understanding of the situation is crucial for selecting the required mode. Each mode implies a fixed delegation of authority for all the various sub-processes for which automated support is available. Figure 7 presents these modes along with the variations in the level of work distribution and the synergy between the automation and the operator in these various modes.

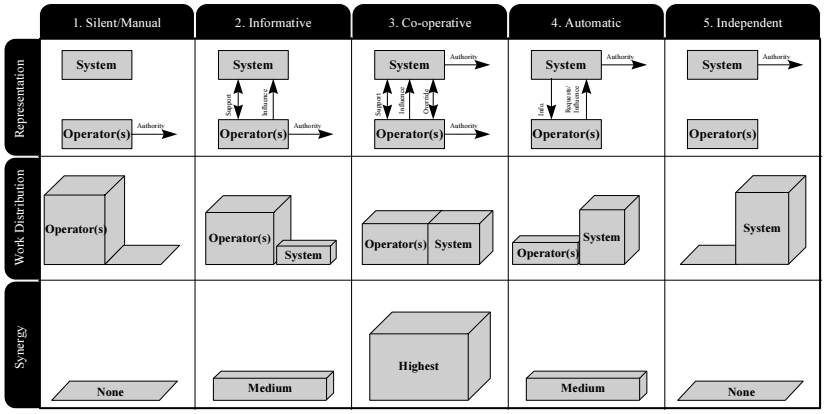



**Figure 7.** Operator-system modes of operation.

## 3. Designing a Decision Support System

The introduction to this section comprises mainly of excerpts from Chalmers [8]. The widely accepted technology-driven response to satisfying operators' information needs for decision making and problem solving in warfare is founded on the premise of a digital model of information that focuses essentially on amassing increasing amounts of data and processing this data into more information, as the basis of reducing battle-space uncertainty for commanders. While certainly providing important capabilities for increasing data coverage and accuracy in the dynamic representation of the battlespace, this effort alone, as characterized by the left portion of the curve in Fig. 8, can easily have the opposite effect to that desired, by contributing instead to additional cognitive demands on operators, and, particularly in critical, high-tempo periods, even greater uncertainty as operators deal with the flood of increased data and making sense of all this data.

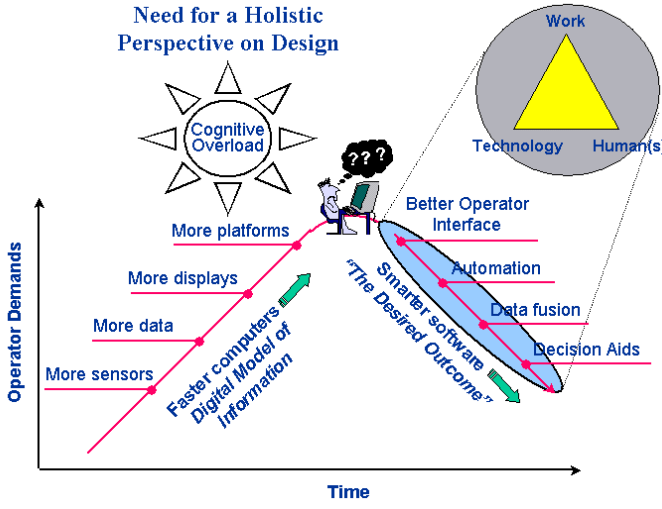



**Figure 8.** Need for a holistic approach on design.

More recent technological efforts, based on developing Data Fusion and decision aiding technologies, promise significant opportunities for coping with the data explosion problem in modern warfare. They are shown on the right portion of the curve of Fig. 8. They offer the potential for improving data processing capabilities by enhancing data integration, and thereby reducing cognitive demands of operators and improving their situation awareness. However, realizing the promised benefits of these newer software-based solutions is extremely challenging. While representing the desired outcome from algorithmic innovations, the benefits implied by the right portion of the curve in Fig. 8 are certainly far from a given.

The literature offers numerous examples that attest to the difficulty of providing effective support in cognitively demanding work environments. In fact, there is increasing evidence that poorly engineered computer-based solutions can lead to substantial performance decrements of the joint human-machine system and even potentially catastrophic results. It has been suggested that when tools dominate, rather than constrain, the joint human-machine system, the designer runs a strong risk of solving the wrong problem, and of creating new problems and undermining existing work strategies.

A critical insight to be derived here is that the characteristics of the workspace and its cognitive demands, the characteristics of human operators, and the opportunities afforded by the technological solution space for supporting operator demands, are in fact so intricately intertwined that successful navigation of the solution space requires considering a number of complex, cross-disciplinary issues in a holistic manner, and at the outset, in dealing with the design of decision support. These are some of the significant features underlying the Cognitive Systems Engineering approaches that will be described next.

### *3.1. Cognitive Engineering System Analyses*

CSE analyses are defined as approaches that aim to develop knowledge about the interaction between human information processing capacities and limitations, and technological information processing systems. The usefulness of a system is closely related to its compatibility with human information processing. Therefore, CSE analyses focus on the cognitive demands imposed by the world to specify how technology should be exploited to reveal the problems intuitively to the decision maker's brain.

#### *3.1.1. Cognitive Work Analysis (CWA)*

Vicente [5] proposes an ecological approach, which can be seen as a CWA, and takes its origin in psychological theories that were first advanced by Brunswick [9] and Gibson [10–11]. These researchers raised the importance of studying the interaction between the human organism and its environment. The *perception* of an object in the environment is a direct process, in which information is simply detected rather than being constructed [11]. The human and the environment are coupled and cannot be studied in isolation. A central concept of this approach is the notion of affordance. The affordance is an aspect of an object that makes it obvious how the object is to be used. Examples are a panel on a door to indicate, "push," and a vertical handle to indicate "pull." When the affordance of an object is obvious, it is easy to know how to interact with it. The environment in which a task is performed has a direct influence on overt behavior. Hence, the ecological approach begins by studying the constraints in the environment that are relevant to the operator. These constraints influence the observed behavior.

**Table 1.** The CWA Phases

| phases of CWA | kinds of information | modelling tools |
|---|---|---|
| work domain analysis | purpose and structure of work domain | abstraction-decomposition space |
| control task analysis | goals to be satisfied, decisions/cognitive processing req'd | decision ladder templates |
| strategies analysis | ways that control tasks can be executed | information flow maps |
| social organisation and cooperation analysis | who carries out work and how it is shared | annotations on all of the above |
| competencies analysis | kinds of mental processing supported | skills, rules and knowledge models |

The ecological approach [5] is comparable to and compatible with Rasmussen's abstraction hierarchy framework [12,13]. This abstraction hierarchy is represented by means-ends relations and is structured in several levels of abstraction that represent functional relationships between the work domain elements and their purposes. Rasmussen has developed a comprehensive methodology that overcomes the limitations of traditional CTA by taking into account the variability of performance in real-life, complex work domains. Conducting CWA requires conducting sequentially five different type of analysis (see Table 1). As indicated in [14], findings from each analysis activity provide a specific type of design information that is captured using a specific modeling tool.

CWA seems to be the best choice to answer questions related to understanding the C2 task. Recently, a feasibility study to investigate the applicability of CWA for C2 was performed to confirm this [14]. The study revealed that the methodology is well suited to deal with decision support design issues but is in practice, if done in a full scale (all sequential CWA phases) for a small problem, time consuming and very expensive to conduct, and the quality of the findings is dependent on the availability of subject-matter experts and on the skills of the people conducting it. In addition, the study did not show the gap between cognitive analyses and design, making the DSS engineering process inefficient.

## 4. A Pragmatic Approach to Cognitive Work Analysis

The Applied Cognitive Work Analysis (ACWA) methodology [15,16] emphasizes a stepwise process to reduce the gap to a sequence of small, logical engineering steps…each readily achievable. At each intermediate point, the resulting decision-centered artifacts create the spans of a design bridge that link the demands of the domain as revealed by the cognitive analysis to the elements of the decision aid.

The ACWA approach is a structured, principled methodology to systematically transform the problem from an analysis of the demands of a domain to identifying visualizations and decision-aiding concepts that will provide effective support.

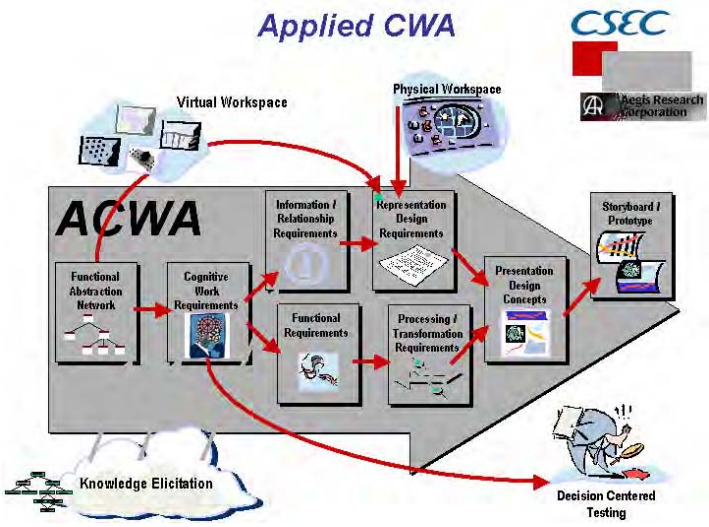



**Figure 9.** The ACWA process from [15,16].

As illustrated in Figure 9, the steps in this process include:

- using a *Functional Abstraction Network* (FAN) model to capture the essential domain concepts and relationships that define the problem-space confronting the domain practitioners;
- overlaying *Cognitive Work Requirements* (CWR) on the functional model as a way of identifying the cognitive demands/tasks/decisions that arise in the domain and require support;
- identifying the *Information/Relationship Requirements* (IRR) for successful execution of these cognitive work requirements;
- specifying the *Representation Design Requirements* (RDR) to define the shaping and processing for how the information/relationships should be represented to practitioner(s);
- developing *Presentation Design Concepts* (PDC) to explore techniques to implement these representation requirements into the syntax and dynamics of presentation forms in order to produce the information transfer to the practitioner(s).

In The ACWA analysis and design approach, design artifacts are created to capture the results of each of these intermediate stages in the design process. These design artifacts form a continuous design thread that provides a principled, traceable link from cognitive analysis to design. However, the design progress occurs in the thought and work in accomplishing each step of the process; by the process of generating these artifacts. The artifacts serve as a post hoc mechanism to record the results of the design progression and as stepping stones for the subsequent step of the process. Each intermediate artifact also provides an opportunity to evaluate the completeness and quality of the analysis/design effort, enabling modifications to be made early in the process.

The linkage between artifacts also ensures an integrative process; changes in one-artifact cascade along the design thread necessitating changes to all. The process is typically repeated in several expanding spirals, each resulting in an improved decision

support system. Ref. [16] provides a more complete and detailed description of the methodology and in [15], the methodology is applied to a practical case of a modern frigate. The following subsections describe briefly the ACWA steps to give the reader a flavor of what this is all about.

### *4.1. Modeling the Work Domain: Functional Abstraction Network (FAN)*

The work domain analysis is performed based on a variety of Knowledge Elicitation (KE) activities. This involved interactions with expert practitioners in the domain and included face-to-face interviews with the experts, watching the experts work in the domain, and observations in simulated exercises with scenarios crafted to address specific aspects of the work domain. In practice, this is an iterative, progressively deepening process. The key is to focus on progressively evolving and enriching the model so as to ultimately discover an understanding of the goal-driven characteristics of the domain that will lead to an understanding of the decisions practitioners are faced with in the domain.

The phrase "bootstrapping process" has been used to describe this process and emphasize the fact that the process builds on itself [17]. Each step taken expands the base of knowledge providing the opportunity to take the next step. Making progress on one line of inquiry (understanding one aspect of the field of practice) creates room to make progress on another. One starts from an initial base of knowledge regarding the domain and how practitioners function within it (often very limited). One then uses a number of KE techniques to expand on and enrich base understanding and to evolve an ACWA model from which ideas for improved support can be generated. For example, one might start by reading available documents that provide background on the field of practice (e.g., training manuals, procedures), the knowledge gained will raise new questions or hypotheses to pursue that can then be addressed in interviews with domain experts, it will also provide the background for interpreting what the experts say. In turn, the results of interviews or exercises may point to complicating factors in the domain that need to be modeled in more detail in the Functional Abstraction Network (FAN). This provides the necessary background to create scenarios to be used to observe practitioner performance under simulated conditions or to look for confirming example cases or interpret observations in naturalistic field studies.

The FAN provides a framework for making explicit the goals to be achieved in the domain and the alternative means available for achieving those goals. High-level goals, such as impacting a critical function, are decomposed into supporting lower-level subgoals. This provides the basis for identifying – through subsequent steps in the analysis and design process – the cognitive activities that arise in the domain and the information needed to support those decisions. The FAN enables the designer to determine where decision-making is likely to be difficult due to the fundamental characteristics of the domain. For example, the FAN helps convey places in the problem space where objectives compete with each other (e.g., where choices have to be made that require some level of sacrificing of one objective in order to achieve another, perhaps more heavily weighted, objective), or otherwise constrain each other (e.g., where the satisfaction of multiple goals need to be considered in determining the best course of action).

### *4.2. Modeling the Cognitive Demands: Cognitive Work Requirements (CWR)*

With the FAN representation of the work domain as the underlying framework, it is possible to derive the cognitive demands for achieving domain goals. In our methodology, we refer to these demands as cognitive requirements. Thus, the term "decision" is used in a broad sense. Based on the underlying premises of the modeling methodology, these decisions center around goal-directed behavior, such as monitoring for goal satisfaction and resource availability, planning and selection among alternative means to achieve goals, and controlling activities (initiating, tuning, and terminating) to achieve goals [18] as well as collaboration activities in team settings [19]. By organizing the specification of cognitive requirements around nodes in the goal-means structure, rather than organizing requirements around predefined task sequences (as in traditional approaches to task analysis), the representation helps insure that the resulting design concepts reflect a decision-centered perspective. The resulting decision support concepts will thus support domain practitioners in understanding the goals to be achieved and what decisions and actions need to be taken to achieve these goals.

The cognitive demands that are derived from a cognitive analysis of the work domain constitute a second key modeling artifact – *Cognitive requirements (CR)s*. CRs are tied directly to nodes in the FAN and provide an intermediate artifact that forms the essential part of the design thread, eventually providing an end-to-end connection from goal nodes in the FAN to supporting decision support concepts.

The FAN forms the basis for the structure of the decision-making activities that will be reflected in the Decision Requirements. For example, every Goal node in the FAN has associated "Goal Monitoring" types of decisions. Likewise, Processes have associated "Process Monitoring" decisions. Similarly, there will always be "Feedback Monitoring" types of decisions related to assessing whether actions are achieving desired results as well as monitoring side effects of actions. Depending on the relationships between nodes in the FAN, there will be decisions related to "Control" (e.g., selection of alternative means to achieve a particular goal) and "Abnormality Detection/Attention Focusing."

Each domain is unique in the decision-making demands imposed on the human operators. As such, each work domain will require slightly different variants of these questions. Successful elucidation of decision requirements will also depend on corroboration from multiple data sources, including case studies, interviews, observations, etc. In addition, guiding insights can come from research on similar work domains as well as basic research on human cognition, decision-making, biases, and errors. For example, previous work on decision making in dynamic, high-risk worlds can guide the analysis and interpretation of analogous worlds in terms of potential points of complexity, typical decision making difficulties and strategies, and critical characteristics of difficult problem-solving scenarios.

### *4.3. Information Requirements to Support Decisions: Information/Relationship Requirements (IRR)*

The next step in the process is to identify and document the information required for each decision to be made. Information Requirements are defined as the set of information elements necessary for successful resolution of the associated decision requirement. This set of information constitutes the third key modeling artifact – *Information*

*Requirements (IR)s*. The focus of this step in the methodology is on identifying the ideal and complete set of information for the associated decision-making.

Information Requirements specify much more than specific data elements; it is data in context that becomes information [20,21]. The data-to-information relationship can be complex and requires a significant amount of computations and/or transformations. For this reason ACWA is a design approach that has a much deeper impact on the entire DSS architecture than merely the look and feel of the final GUI. For example, in the case of a thermodynamic system, an IR might be "flow coefficient with respect to appropriate limits." This requires the estimation of the "flow coefficient" parameter derived from model-based computations and sensor values and the comparison of that parameter against a limit referent. The degree of transformation required can vary from simple algebra to complex, intelligent algorithms. Ref. [22] provides an example of Information Requirements that could only be satisfied by an advanced planning algorithm and significant data transformations.

In addition, it is important to note that identifying Information Requirements focuses on satisfying the decision requirements and is *not* limited by data availability in the current problem-solving environment. In cases where the required data is not directly available ACWA provides a rationale for obtaining that data (e.g., pulling data from a variety of previously stove-piped databases, adding additional sensors, or creating "synthetic" values). This is a critical change from the typical role that Human Factors Engineers have had in the past (designing an interface after the instrumentation has been specified). Consequently, this type of an approach is fundamentally broader in scope than other approaches to interface design that do not consider the impact of Information Requirements on system architecture specifications [23].

The specific context and concatenation of data to form Information Requirements depends on the specific Cognitive/Decision Requirement being satisfied. The same data elements can be cast into Information Requirements in different ways that support very different decisions.

Just as the FAN representation provided the framework for the derivation of decision requirements, the decision requirements provide the essential context for the Information Requirements because they indicate the factors (and thus information) that will need to be considered in making decisions.

### *4.4. Linking Decision Requirements to Aiding Concepts: Representation Design Requirements (RDR)*

The FAN and its associated CWR and IRR "overlays" constitute a solid foundation for the development of aiding concepts to form the decision support system. The design of the decision support system occurs at two levels: at a micro level to ensure that each presentation element effortlessly communicates its information to the user; and at the macro level to ensure that the overall collection of presentation design concepts (the decision support system in a holistic sense) is organized in an intuitive way that does not add its own "manage the decision support system" cognitive burdens to those of the domain.

This step in the ACWA process develops the specification of the display concept and how it supports the cognitive tasks, and is captured in Representation Design Requirements (RDR) for the eventual development of Presentation Design Concepts (PDC). The RDR defines the goals and scope of the information representation in terms of the cognitive tasks it is intended to support (and thus a defined target region of the

FAN). It also provides a specification of the supporting information required to support the cognitive tasks. An RDR is another span of the bridge that helps to link the decisions within the work domain to the visualization and decision support concepts intended to support those decisions. In many cases, multiple design concepts may be generated that attempt to satisfy the RDR's requirements. Typically, other supporting artifacts are generated at this step in the process as required to specify such issues as presentation real-estate allocation, attention management (salience) across the information to be presented, etc.

The RDR also represents a critical system design configuration management tool, critical for ensuring coverage of the functional decision space across all presentations and presentation design concepts. The RDR begins the shift in focus from "what" is to be displayed to "how," including annotations on relative importance that maps to relative salience on the visualization, etc. A complete RDR is actually a set of requirements "documents," each describing the requirements for the intended representation of the IRRs. It contains descriptions of how all presentation mechanisms of the domain practitioner's workspace are to be coordinated, how available audio coding mechanisms are allocated, similarly for visual, haptic, and any other sensory channels to be employed. The RDR is not only a compilation of information developed earlier, it has the added value of a more complete description of the behaviors and features needed to communicate the information effectively as well as an allocation of the Information/Relationship Resources across the entire set of displays within the workspace. When done correctly it is still in the form of a "requirement" and not an implementation. This artifact becomes a key transition between the Cognitive System Engineer, the System Developer, and the System (Effectiveness) Tester.

The RDR also provides one important ancillary benefit, as long as the domain remains unchanged, the RDR serves as an explicit documentation of the *intent* of the presentation *independent* of the technologies available and used to implement the decision support system. As newer technologies become available, and as their interaction with human perception becomes better understood, the technologies used to implement the RDR requirements can evolve.

### *4.5. Instantiating the Aiding Concept as Presentation Design Concepts (PDC)*

From the RDR's specification of how information is to be represented within the decision support system, the next step of the ACWA process is the explicit design of Presentation Design Concepts (PDCs) for the decision support system. (A similar process is used for the design of auditory, visual, or other senses' presentations of the RDR's specification.) This final step requires an understanding of human perception and its interaction with the various presentation techniques and attributes. As such, it requires considerable skill and ability beyond cognitive work analysis. The actual design of a revolutionary aiding concept is probably one of the largest "design gaps" that needs to be bridged within the ACWA process. The ACWA design practitioner must be fluent in the various presentation dimensions: color, layout, line interactions, shape, edge detection, etc. Essentially the designer must really understand what characteristics of presentation implicitly specify about the interaction with the user's perception. The conversion of the requirements in the RDR to a sensory presentation form in a PDC requires considerable skill and background in these areas. With the RDR as a guide, the sketches, proposals, brainstorming concepts can all be resolved back against the display's intent and requirements. The issues of how it is perceived can best be done with

empirical testing of prototypes, and often requires considerable tuning and adjustment to achieve the representational capabilities specified in the RDR.

Of all the steps in ACWA, this final presentation development requires a significant background in presentation technologies, human perceptual characteristics, and how they interact. The other ACWA artifacts, notably the RDR, do provide a test basis to iterate the presentation design concepts. By testing each proposed display prototype against the single indicator question of "does it support the decisions it is supposed to as defined by the RDR?" it is possible to at least identify unsuccessful attempts and continue to design toward a more successful one. This last step across the gap is often difficult, but the ACWA methodology has made it a much smaller step, from a much more solid footing than would be the case if attempting to directly design a presentation without its RDR precursor.

## 5. Conclusions

Although information systems have evolved dramatically through the impact of technology, the fundamental nature of warfare remains unchanged. In essence, C2 is shaped by two factors – uncertainty and time – that dominate the environment in which military decisions are made. This paper addressed some critical issues to be considered in the automated assistance to the decision making team to efficiently access the vast amounts of information available, assist in situation awareness and provide advice on possible actions.

This paper also presented a brief discussion of CSE analyses methods and their benefits. It described CSE analyses as approaches that focus on the cognitive demands imposed by the world to specify how technology should be exploited to reveal the problems intuitively (affordance concept) to the decision maker's brain.

## References

[1] Pigeau, R., "The human in command," Defense Science & Technology, Issues #2, January 1998.
[2] Endsley, M.R., "Toward a Theory of Situation Awareness in Dynamic Systems," Human Factors Journal, 37 (1), 32–64, March 1995.
[3] Preece, J., Human-Computer Interaction. In Y. Rogers, H. Sharp, D. Benyon, S. Holland and T. Carey (Eds.) Addison-Wesley, England, 1994.
[4] Jonassen D.H., Hannum, W.H., & Tessmer, M. Handbook of task analysis procedures. Praeger Publishers: New York, 1989.
[5] K.J. Vicente, A Few Implications of an Ecological Approach to Human Factors, Lawrence Erlbaum Associates, NJ, 1995.
[6] Breton, R., Rousseau, R. & Price, W.L. *The Integration of Human Factors in the Design Process: a TRIAD Approach.* Defence Research Establishment Valcartier, TM 2001-002, November 2001, 33 pages.
[7] Chalmers, B., "On the Design of a Decision Support System for a Data Fusion and Resource Management in a Modern Frigate," NATO Symposium on Sensor Data Fusion and Integration of the Human Element, Ottawa, Canada, September 1998.
[8] Chalmers, B., Easter, J.R., Potter, S., "Decision-Centred Visualisations for Tactical Decision Support on a Modern Frigate," Proceedings of 2000 CCRTS, Naval Postgraduate School, Monterrey, CA, June 26–28, 2000.
[9] Brunswick, E., Perception and the representative design of experiments (2nd ed.) Berkeley: University of California Press, 1956.
[10] Gibson, J. J., The problem of temporal order in stimulation and perception. Journal of Psychology, 62, 141–149, 1966.

[11] Gibson, J.J., The Ecological Approach to Visual Perception, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1979.
[12] Rasmussen, J., Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering, North-Holland, New York, 1986.
[13] Rasmussen J., Pejtersen, A.M., & Goodstein, L.P. (1994). *Cognitive Systems Engineering*. New York: Wiley & Sons.
[14] Chalmers, B.A., Burns, C.M., "A Model-Based Approach to Decision Support Design for a Modern Frigate," TTCP Symposium on Coordinated Maritime Battlespace Management, Space and Naval Warfare Systems Center, San Diego, CA, USA, May, 1999.
[15] Paradis. S., Elm, W.C., Potter. S.S., Breton. R. and Bossé. E., A Pragmatic Cognitive System Engineering Approach to Model Dynamic Human Decision-Making Activities in Intelligent and Automated Systems, NATO RTO HFM Symposium on Human Integration in Intelligent and Automated Systems, Warsaw, October 2002, 8 pp.
[16] Elm, W.C., Potter, S.S., Gualtieri, J.W., Roth, E.M., and Easter, J.R. (in prep). Applied cognitive work analysis: A pragmatic methodology for designing revolutionary cognitive affordances. Hollnagel, E. (Ed.). Handbook of Cognitive Task Design.
[17] Potter, S.S., Roth, E.M., Woods, D.D., and Elm, W.C. (2000). Cognitive task analysis as bootstrapping multiple converging techniques. In Schraagen, Chipman, and Shalin (Eds.). *Cognitive Task Analysis.* Mahwah, NJ: Lawrence Erlbaum Associates.
[18] Roth, E.M. & Mumaw, R.J. (1995). Using Cognitive Task Analysis to Define Human Interface Requirements for First-of-a-Kind Systems*. Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting*, (pp. 520–524). Santa Monica, CA: Human Factors and Ergonomics Society.
[19] Gualtieri, J.W., Roth, E.M., & Eggleston, R.G. (2000). Utilizing the abstraction hierarchy for role allocation and team structure design. In *Proceedings of HICS 2000–- 5th International Conference on Human Interaction with Complex Systems*. (pp. 219–223). Urbana-Champaign, IL: IEEE.
[20] Woods, D.D. (1988). The significance messages concept for intelligent adaptive data display. Unpublished technical report. Columbus, OH: The Ohio State University.
[21] Woods, D.D. (1995). Toward a theoretical base for representation design in the computer medium: Ecological perception and aiding human cognition. In J. Flach, P. Hancock, J. Caird, and K. Vicente (Eds*.) Global Perspectives on the Ecology of Human-Machine Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates, 157–188.
[22] Potter, S.S., Ball, R.W., Jr., and Elm, W.C. (Aug., 1996). Supporting aeromedical evacuation planning through information visualization. In *Proceedings of the 3rd Annual Symposium on Human Interaction with Complex Systems*. Dayton, OH: IEEE. pp. 208–215.
[23] Vicente, K.J., Christoffersen, K., and Hunter, C.N. (1996). Response to Maddox critique. *Human Factors*, 38 (3), 546–549.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Multimodal Input Fusion in Human-Computer Interaction

## *On the Example of the NICE Project*

Andrea CORRADINI[a], Manish MEHTA[a], Niels Ole BERNSEN[a],
Jean-Claude MARTIN[b,c] and Sarkis ABRILIAN[b]
[a]*Natural Interactive Systems Laboratory (NISLab), University of Southern Denmark, DK-Odense M, Denmark*
[b]*Laboratory of Computer Science for Mechanical and Engineering Sciences, LIMSI-CNRS, F-91403 Orsay Cedex, France*
[c]*Montreuil Computer Science Institute (LINC-IUT), University Paris 8, F-93100 Montreuil, France*

**Abstract.** The modality integration issue is addressed with the example of a system that aims at enabling users to combine their speech and 2D gestures when interacting with life-like characters in an educative game context. The use of combined input speech, 2D gesture and environment entities for user system interaction is investigated and presented in a preliminary and limited fashion.

**Keywords.** Human-computer interaction, input fusion, gesture, speech

…"I feel that as a modern civilization we may have become intoxicated by technology, and find ourselves involved in enterprises that push technology and build stuff just because we can do it. At the same time we are confronted with a world that is increasing needful of vision and solutions for global problems relating to the environment, food, crime, terrorism and an aging population. In this information technology milieu, I find myself being an advocate for the humans and working to make computing and information technology tools that extend our capabilities, unlock our intelligence and link our minds to solve these pervasive problems..." (Thomas A. Furness III [1])

## 1. Introduction

Human-Computer Interaction (HCI) is a research area aiming at making the interaction with computer systems more effective, easier, safer and more seamless for the users.

Desktop-based interfaces also referred to as WIMP-based (Windows, Icons, Menus and Pointers) Graphical User Interfaces (GUIs), have been the dominant style of interaction since their introduction in the 80s when they replaced command line interfaces. WIMP interfaces enabled access to computers for more people by providing the user with a look and feel, visual representation and direct control using mouse and keyboard. Nevertheless, they have some intrinsic deficiencies: they passively wait for the user to carry out tasks by means of mouse or keyboard and often restrict input to single

non-overlapping events. As the way we use computers is becoming more pervasive, it is not clear how GUI-WIMP interfaces will accommodate for and scale to a broader range of applications. Therefore, post-WIMP interaction techniques that go beyond the traditional desktop metaphor need to be considered.

In the scientific community, a shared belief is that the next step in the advancement of computing devices and user interfaces is not to simply make applications faster but also to add more interactivity, responsiveness and transparency to them. In the last decade much more efforts have been directed towards building multi-modal, multi-media, multi-sensor user interfaces that emulate human-human communication with the overall long-term goal to transfer to computer interfaces natural means and expressive models of communication [2]. Cross-disciplinary approaches have begun developing user-oriented interfaces that support non-GUI interaction by synergistically combining several simultaneous input and/or output modalities, thus referred to as multimodal user interfaces. In particular, multimodal Perceptual User Interfaces (PUI) [3] have emerged as potential candidates for being the next interaction paradigm. On one hand, these kinds of interfaces can make use of machine perception techniques to sense the environment allowing the user to use input modalities such as speech, gesture, gaze, facial expression and emotion [4]; on the other they can leverage human perception by offering information and context through more meaningful output channels [5]. As benefits, PUIs will provide their users with reduced learning times, performance increase, an increased retention and a more satisfying usage experience.

So far, such interfaces have not yet reached widespread deployment. As a consequence this technology is not mature and most of these interfaces are still functional rather than social, thus far from being intuitive and natural. The rigid syntax and rules over the individual modalities along with the lack of understanding of how to integrate them are the two main open issues.

In this paper, we will address the modality integration issue on the example of the NICE (Natural Interactive Communication for Edutainment) [6] project we are currently working on. We begin by giving an overview of multimodal fusion input in the next section. Section 3 presents related work while Section 4 describes the on-going NICE project. We conclude with discussion on other possible applications and future directions for development.

## 2. Multimodal Input Fusion: An Overview

In multimodal systems, complementary input modalities provide the system with non-redundant information whereas redundant input modalities allow increasing both the accuracy of the fused information by reducing overall uncertainty and the reliability of the system in the case of noisy information coming from a single modality. Information in one modality may be used to disambiguate information in the other ones. The enhancement of precision and reliability is the potential result of integrating modalities and/or measurements sensed by multiple sensors [7].

In order to effectively use multiple input modalities there must be some technique to integrate the information provided by them into the operation of the system. In the literature, two main approaches have been proposed. The first one integrates signals at the feature level whereas the second one fuses information at a semantic level. The feature fusion strategy is generally preferred for closely coupled and synchronized modalities, such as speech and lip movements. However, it tends not to scale up, requires

a large amount of data for training and has high computational costs. Semantic fusion is mostly applied to modalities that differ in the time scale characteristics of their features. In this latter approach, timing plays an important role and hence all fragments of the modalities involved are time-stamped and further integrated in conformity with some temporal neighborhood condition. Semantic fusion offers several advantages over feature fusion. First, the recognizers for each single modality are used separately and therefore can be both trained separately and integrated without retraining. Furthermore, off-the-shelf recognizers can be utilized for standard modalities e.g. speech. An additional advantage is simplicity: modalities integration does not add any extra parameters beyond those used for the recognizers of each single mode allowing for generalization over number and kind of modalities.

Typically, the multimodal fusion problem is either formulated in a maximum likelihood estimation (MLE) framework or deferred to the decision level when most of the joint statistical properties have been lost. To make the fusion issue tractable within the MLE framework, the individual modalities are usually assumed independent of each other. This simplification allows the use of simple parametric models (e.g. Gaussian functions) for the joint distributions that cannot capture the complex modalities' relationships.

Very few alternatives to these classical approaches have proposed to make use of non-parametrical techniques or finite-state devices. [8] put forward a non-parametrical approach based on mutual information and entropy for audio-video fusion of speech and camera-based lip-reading modalities at signal level. Such a method does not make any strong assumptions about the joint measurement statistics of the modes being fused, nor does it make use of any training data. Nevertheless, it has been demonstrated over a small set of data while its robustness has not been addressed yet. In [9] multimodal parsing and understanding was achieved using a weighted finite-state machine. Modality integration is carried out by merging and encoding into a finite-state device both semantic and syntactic content from multiple streams. In this way, the structure and the interpretation of multimodal utterances can be captured declaratively in a context-free multimodal grammar. Whereas the system has been shown to improve speech recognition by dynamically incorporating gestural information, it has not been shown to provide superior performance, either in terms of error rate reductions, or in terms of processing speed, over common integration mechanisms. More importantly, it does not support mutual disambiguation (MD), i.e., using speech recognition information to inform the gestural recognition processing, or the processing of any other modality.

The kind of fusion strategy to choose may not depend upon the input modalities only. There is empirical evidence [10] that distinct individual groups (e.g. children and adults) adopt different multimodal integration behaviors. At the same time, multimodal fusion patterns may depend upon the particular task at hand. A comprehensive analysis of experimental data may therefore help gather insights and knowledge about the integration patterns thus leading to the choice of the best fusion approach for the application, modalities, users and task at hand.

The use of distributed agent architectures, such as the Open Agent Architecture (OAA) [11], in which dedicated agents communicate with each other by means of a central blackboard, is also common practice in multimodal systems.

Besides architectures aiming at emulating the way human beings communicate with each other in their everyday lives, a variety of other multimodal systems have been proposed for recognition and identification of individuals based on their physiological and/or behavioral characteristics. These biometric systems address security is-

sues with the purpose to ensure that only legitimate users access a certain set of services, e.g. secure access to buildings, computer systems and ATMs. Biometric systems typically make use of either fingerprints or iris or face or voice or hand geometry to assess the identity of a person. Because of issues related to non-universality of some single traits, spoof attacks, intra-class variability, and noisy, data architectures that integrate multiple biometric traits have shown substantial improvement in efficiency and recognition performance [12–15]. Being a non issue for such systems, user traits temporal synchronization makes signal integration less complex than in HCI architectures and can be seen as a decision problem within a pattern recognition framework. Techniques employed for combining biometric traits range from the weighted sum rule [16], Fisher discriminant analysis [16], decision trees [15], to a decision fusion scheme [17].

## 3. Related Work

Several multimodal systems have been proposed after Bolt's pioneering system [18]. Speech and lip movements have been merged using histogram techniques [19], multivariate Gaussians [19], artificial neural networks (ANNs) [20,21] or hidden Markov models (HMMs) [19]. In all these systems, the probabilistic outputs of modalities have been combined assuming conditional independence by using either Bayes' rule or a weighted linear combination over the mode probabilities for which the weights were adaptively determined.

While time synchrony is inherently taken care of (at least partially) in the ANN-based systems described in [20,21], this cannot be adequately addressed in the other systems. To address temporal integration of distinct modalities, a generic framework has been put forward in [22]. It is characterized by three steps and makes use of a particular data structure named melting pot. The first step, referred to as microtemporal fusion, combines information that is produced either in parallel or over overlapping time intervals. Further, macrotemporal fusion takes care of either sequential inputs or time intervals that do not overlap but belong to the same temporal time window. Eventually, contextual fusion serves to combine input according to contextual constraints without attention to temporal constraints.

In speech and gesture systems it is common to have separate recognizers for each modality. The outcome of the single recognizers may be used for further monomodal processing at a higher level (e.g. a natural language understanding module to deal with the spoken input representation from the speech recognizer) and/or followed by the late fusion module. QuickSet [23] is a multimodal pen-gestures and spoken input system for map-based applications. A multi-dimensional chart parser semantically combines the statistically ranked set of input representations using a declarative unification-based grammar [24]. Temporal fusion relies on time proximity: time-stamped features from different input channels are merged if they occur within a 3 to 4 second time window.

In [25], two statistical integration techniques have been presented: an estimate and a learning approach. The estimate approach makes use of a multimodal associative map to express, for each multimodal command, the meaningful relations that exist between the set of single constituents. During multimodal recognition, the posterior probabilities are linearly combined with mode-conditional recognition probabilities that can be calculated from the associative map. Mode-conditional recognition probabilities are used as an approximation of the mode-conditional input feature densities. In the learning approach, called Members to Teams to Committee (MTC), multiple teams are built to

reduce fusion uncertainty. Teams are trained to coordinate and weight the output from the different recognizers while their outputs are passed on to a committee that establishes the N-best ranking.

The EMBASSI system [26] combines speech, pointing gesture and the input from a graphical GUI into a pipelined architecture. The Smartkom [27] is multimodal dialogue system that merges gesture, speech and facial expressions for both input and output via an anthropomorphic and affective user interface. In both systems, input signals are assigned a confidence score that is used by the fusion module to generate a list of interpretations ranked according to the combined score.

## 4. The NICE Project

### *4.1. The NICE Project and Its Multimodal Scenario*

The NICE PC-based system aims at enabling users to combine their speech and 2D gestures when interacting with characters in an educative game context. It addresses the following scenario. 3D animated life-like fairy tale author Hans Christian Andersen (HCA) is in his 19th Century study surrounded by artifacts. At the back of the study is a door which is slightly ajar and leads out into the fairy tale games world. This world is populated by some of his fairy tale characters and their entourage, including, among others, the Naked Emperor and the Snow Queen. When someone talks to HCA, this user becomes an avatar that walks into HCA's study. In the study, the user can have spoken conversation with HCA, including the use of gesture input to support interaction by for example indicating artifacts during conversation. At some point, the user may wish to visit the fairy tale world and is invited by HCA to go through the door at the back of the study. Once in the fairy tale world, the user may interact with the characters populating the fairy world using speech and 2D gesture. The intended users are primarily youngsters and, secondarily, everyone else. The primary scenario of use is in technology and other museums in which, expectedly, the duration of individual conversations will be 5–30 minutes. Secondarily, we investigate the feasibility of prototyping the world's first spoken computer game for home use averaging 30 hours of user interaction time.

The primary research challenge addressed in NICE is to move from the existing paradigm of task-oriented spoken dialogue with computer systems to the next step which we call domain-oriented spoken dialogue. In domain-oriented spoken dialogue, there is no longer any user task to constrain the dialogue and help enormously in its design and implementation, but only the semi-open domain(s) of discourse which, in the case of HCA, are: his life, his fairy tales, his 3D physical presence, his modeling of the user, and his role as a kind of gate-keeper for the virtual fairy tales world. In a limited fashion, however, we also investigate the use of combined input speech and 2D gesture for indicating objects and other entities of interest.

### *4.2. Requirements for Multimodal Input from Experimental Data*

Early multimodal prototypes have been developed without much knowledge about how the potential final users would combine the distinct modes to interact with the system. This design approach has changed over the years and it is now considered important to collect behavioral data prior to and/or while the design phase via a simulation of the

future system using a Wizard of Oz (WoZ) approach. In this kind of study, an unseen assistant plays the role of the computer, processing the user's input and responding, as the system is expected to.

In order to collect data on the multimodal behavior that our future system might expect from the users, we have built a simple 2D game application. In this application the user can interact with several 2D characters located in different rooms to which he/she has to bring some objects back. The user can issue spoken input and/or pen-gesture to accomplish the desired task. In the following, we focus on how we are currently taking these observations into account for the specification and development of a first demonstrator of the NICE multimodal module.

The observed commands were classified into six sets: *getIn* where the user wants to get in a room from the corridor, *askWis* when the user asks the character for an object, *getOut* when the user wants to leave the room he/she is currently in, *takeObject* when the user wants to take an object in the current room and later hand it over to another character, *giveObject* when the user wants to give an object to the character in the current room and this is placed in a deposit area graphically visible from the interface, and finally *social dialogues* when the user utterance is not directly related to the task at hand.

By analyzing the way the user carried out these commands, we were able to detect a few common multimodal patterns useful for the design of the multimodal module. For example, we were able to find out that a few single commands are always issued unimodally (e.g. when the user utters "What do you want?" without any accompanying gesture) while others are issued indifferently either unimodally, with no dominant modality (e.g. in the case of the user either uttering "get into the red room" to express the wish to enter a red painted room or just circling the door of the red room), or multimodally (providing both spoken and gestural input to the system). In case of multimodal commands, we have seen that gesture always precedes speech and this is consistent with previous empirical evidence [28]. Other commands were noticed to use multiple gestures in sequence (e.g. to get into a room the user clicks on a door and then circles it). Also, gesture-only commands have at present a high semantic variability which can be resolved only if information about location of the gesture or the object is known (e.g. drawing a circle about an object in the room means *takeObject* whereas the same gesture referring to an object in the deposit area means *giveObject*). Eventually, few unexpected speech and gesture combinations were observed such as when the user utters "thank you" while for instance performing a *takeObject* gesture. The observed gestures were classified into the following shape categories: pointing (makes up for 66% of the data), circling (18,1%), line (5.4%), arrow (2.1%) and explorative gestures (8.5) i.e. those that occur when the user gestures without touching the screen. Accurate details on the experiment and its results can be found in [29].

### *4.3. Gesture Recognition Module*

While both pointing and exploring categories observed in the corpus do not need any specific recognition algorithm, to recognize circling, line and arrow, a 2D gesture recognition module was developed using Ochre Neural Networks technology [30] trained with templates extracted from the experimental data corpus. The approach is easily extendable to more gestures and other patterns may be added later if it will turn out necessary.

An N-best hypotheses list results from the gesture classification task. The list is wrapped into an XML-like format that has been agreed upon to allow messages to be exchanged by the different modules.

### *4.4. Speech Processing Module*

In order to test the input fusion we developed a very simple speech processing module to provide input to the Input fusion module. So far, a fairly simple speech grammar has been manually specified out of the set of utterances in the corpus. 94 sentences were defined: 18 formulations of the *askWish* command, 15 for *giveObject*, 37 for *takeObject*, 16 for quitting and 8 for greetings. We used the off-the-shelf IBM ViaVoice [31] technology as speech recognizer. Currently, no natural language processing module is employed. In addition, the grammar being very limited no conversation dialogue is possible with the system. In the near future, we will be adding a natural language processing module to add partial dialogue conversation capabilities. Similarly to the gesture modules, the speech processing results in an XML-like message to be passed on to the input fusion component.

### *4.5. Input Fusion*

The input processing architecture of the NICE system has been specified as shown in Figure 1. The speech recognizer sends a word lattice including prioritized text string hypotheses about what the user said to the natural language understanding module (NLU), which parses the hypotheses and passes a set of semantic hypotheses to the input fusion module. In parallel, the gesture recognizer sends hypotheses about the recognized gesture shape to the gesture interpreter. The gesture interpreter (GI) consults with the simulation module (SM) to retrieve information on relevant objects visible to the user, interprets the gesture type, and forwards its semantic interpretations to the input fusion module. The input fusion module combines the information received and passes on its multimodal input interpretation to the dialogue manager (DM).

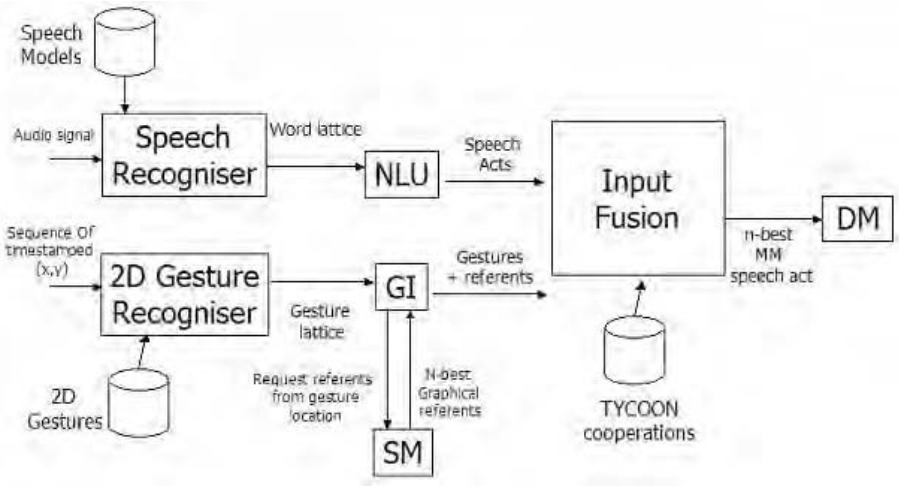



**Figure 1.** Sketch of the NICE input processing architecture.

In previous work [32,33] we proposed a typology made of several types of cooperation between modalities for analyzing and annotating the user's multimodal behavior and also for specifying interactive multimodal behaviors. Basic types of cooperation among modalities are: *equivalence* to specify modalities that occur interchangeably in the same unimodal command, *specialization* for commands that are always specified with the same modality, *redundancy* for modalities that either combined or taken separately produce the same command, and *complementarity* for modalities that need to be merged to result in a meaningful command. We have also included the notion of referenceable objects to specify entities the user can refer to using uni- or multimodally utterances.

We utilize a text file to contain the description of the expected modalities combination where the variables are defined and reused later by multimodal operators such as specialization, complementarity, etc. For example, a *giveObject* command can be specified using the following text script

```
…
#- giveObject command
specialisation   CC3      IS3
specialisation   CC4      IG1

semantics CC4 position

complementarity   temporalProximity 5000 CC5   CC3   CC4
endHypothesis  CC5  giveObject
…
```

Here, IS3 stands for one of the possible utterances associated with a *giveObject* spoken command, IG1 stands for the detection of a gesture associated to the gestural part of the same command, and the CC# tags are contextual units which are activated by different multimodal patterns. For example CC5 gets activated if CC3 and CC4 are activated within a 5000 ms time window. The multimodal module [33] parses this text file and makes use of the TYCOON symbolic-connectionist technique to classify multimodal behaviors. TYCOON was inspired by the Guided Propagation Networks [34] that are composed of processing units exchanging symbolic structures representing multimodal fusion hypotheses.

### *4.6. Input Fusion and Message Passing: An Example*

The following example illustrates the result of the fusion given the incoming messages from the distinct modes. The messages were generated when the user, after asking permission for picking up an object (a coffee machine), uttered "thanks" while pointing to the object.

OUTPUT FROM SPEECH PROCESSING MODULE

```
<semanticRepresentation>
   <score>0.8</score>
   <function>thank</function>
<semanticRepresentation>
```

OUTPUT FROM GESTURE RECOGNITION MODULE

```
<recognisedGesture>
  <hyp n="1">
    <score>0.75</score>
    <shape>point</shape>
    <begin> …</begin>
    <end>…</end>
    <2DboundingBox>…</2DboundingBox>
  </hyp>
  <hyp n="2">
    <score>0.2</score>
    <shape>line</shape>
    <begin> …</begin>
    <end>…</end>
    <2DboundingBox>…</2DboundingBox>
    <direction>…</direction>
  </hyp>
</recognisedGesture>
```

OUTPUT FROM GESTURE INTERPRETER MODULE

```
<semanticRepresentation>
  <score>0.75</score>
  <function>takeObject</function>
  <object>coffeeMachine#1</object>
</semanticRepresentation>
```

OUTPUT FROM INPUT FUSION

```
<semanticRepresentation>
  <score>0.9</score>
  <function>takeObject</function>
  <object>coffeeMachine#1</object>
</semanticRepresentation>
```

In this example, the function and the object were not provided by speech but by gesture. Yet, the compatible fusion enables the increase of the score of the command after merging hypothesis from speech processing and gesture recognizer.

## 5. Conclusions and Future Directions

There is evidence that people are polite to the computer they are using, treat them as member of the same team but also expect them to be able to understand their needs and be capable of natural interaction. In [35], for instance, is reported that when a computer asked a human being to evaluate how well the computer had been doing, the individual provides more positive responses than in the case of a different computer asking the same question. Likewise, it was shown that people tend to give computers higher performance ratings if the computer has recently praised the user. In light of these inclinations, systems making use of human-like modalities seem to be more likely to provide

users with the most natural interface for many applications. Humans will benefit from this new interface paradigm as automatic systems will capitalize on the inherent capabilities of their operators, while minimizing or even eliminating the adverse consequences of human error or other human limitations.

The rigid syntax and rules over the individual modalities along with the lack of understanding of how to integrate them are the two main open issues in the development of multimodal systems. This paper provided an overview of techniques to deal with the latter issue and described the fusion in the on-going NICE project. The current version of the input fusion module will have to be improved in the following directions: recognize more complex and multi-stroke gestures, integrate with the other modules such as the NLU and the 3D environment, and add environment information to resolve input ambiguities.

To illustrate this latter issue, suppose, for instance, that the user says, "What is written here?" whilst roughly encircling an area on the display. Let's assume the speech recognizer passes on hypotheses, such as "what is it gray here," "what does it say here," along with the correct one, while the gesture recognizer passes on hypotheses, such as that the user wrote the letter Q and that the user drew a circle. The simulation module would inform the gesture interpreter that the user could have referred to the following adjacent objects: a bottle up front on the display and a distant house. We would refer to these objects as environment content. Eventually, the input fusion module will have to combine the time-stamped information received from the natural language understanding and gesture interpretation modules, select the most probable multimodal interpretation, and pass it on to the dialogue manager. The selection of the most probable interpretation should allow ruling out inconsistent information by both binding the semantic attributes of different modalities and using environment content to disambiguate information from the single modalities [36].

Multimodal fusion can be used to deal with either multimodal sensors or multimodal inputs or a combination of the two. Several relevant families of applications could benefit from an accurate and reliable fusion integration strategy. Possible applications range from gesture-cum-speech systems for battlefield management [23,37], biometric systems [15], remote sensing [38], crisis management [39], to aircraft and airspace applications [40].

## Acknowledgments

The support from the European Commission, IST Human Language Technologies Programme, IST-2001-35293 is gratefully acknowledged.

## References

[1] Furness, T.A. III, Towards Tightly Coupled Human Interfaces, In: Frontiers of Human-Centred Computing, Online Communities and Virtual Environments, Earnshaw, R., Guedj, R., van Dam, A., and Vince, J., eds., pp. 80–98, 2001.
[2] Bernsen, N.O., Multimodality in language and speech systems. From theory to design support tool, In: Granström, B., House, D., and Karlsson, I. (Eds.): Multimodality in Language and Speech Systems, Dordrecht: Kluwer Academic Publ., pp. 93–148, 2002.
[3] http://www.cs.ucsb.edu/conferences/PUI/index.html.
[4] Picard, R.W., Affective Computing, MIT Press, 1997.

[5] Turk, M., Perceptual User Interfaces, In: Frontiers of Human-Centred Computing, Online Communities and Virtual Environments, Earnshaw, R., Guedj, R., van Dam, A., and Vince, J., eds., pp. 39–51, 2001.
[6] www.niceproject.com.
[7] Kittler, J., On Combining Classifiers, IEEE Transactions on PAMI, vol. 20, no. 3, pp. 226–239, 1998.
[8] Fisher, J.W., and Darrell, T., Signal Level Fusion for Multimodal Perceptual Interface, In: Proceedings of the Conference on Perceptual User Interfaces, Orlando, Florida, 2001.
[9] Johnston, M., and Bangalore, S., Finite-state Multimodal Parsing and Understanding, Proceedings of the International Conference on Computational Linguistics, Saarbruecken, Germany, 2000.
[10] Xiao, B., Girand, C., and Oviatt, S.L., Multimodal Integration Patterns in Children, Proceedings of 7th International Conference on Spoken Language Processing, pp. 629–632, Denver, Colorado, 2002.
[11] Cheyer, A., and Martin, D., The Open Agent Architecture, In: Journal of Autonomous Agents and Multi-Agent Systems, vol. 4, no. 1, pp. 143–148, March 2001.
[12] Dieckmann, U., Plankensteiner, P., and Wagner, T., Sesam: A biometric person identification system using sensor fusion, In: Pattern Recognition Letters, vol 18, no. 9, pp. 827–833, 1997.
[13] Kittler, J., Li, Y., Matas, J., and Sanchez, M.U., Combining evidence in multimodal personal identity recognition systems, Proceedings 1st International Conference on Audio-Video Personal Authentification, Crans-Montana, Switzerland, pp. 327–334, 1997.
[14] Maes, S., and Beigi, H., Open sesame! Speech, password or key to secure your door?, Proceedings 3rd Asian Conference on Computer Vision, Hong-Kong, China, pp. 531–541, 1998.
[15] Ross, A., Jain, A., Information Fusion in biometrics, Pattern Recognition Letters, vol. 24, no. 13, pp. 2115–2121, 2003.
[16] Wang, Y., Tan, T., Jain, A.K., Combining Face and Iris Biometrics for Identity Verification, Proceedings of 4th International Conference on Audio- and Video-Based Biometric Person Authentication, Guildford, UK, 2003.
[17] Jain, A., Hong, L., and Kulkarni, Y., A Multimodal Biometric System Using Fingerprint, Face and Speech, Proceedings of 2nd Int'l Conference on Audio- and Video-based Biometric Person Authentication, Washington D.C., pp. 182–187, 1999.
[18] Bolt, R.A., Put that there: Voice and gesture at the graphic interface, Computer Graphics, vol. 14, no. 3, pp. 262–270, 1980.
[19] Nock, H.J., Iyengar, G., and Neti, C., Assessing Face and Speech Consistency for Monologue Detection in Video, Proceedings of ACM Multimedia, Juan-les-Pins, France, 2002.
[20] Meier, U., Stiefelhagen, R., Yang, J., and Weibel, A., Towards Unrestricted Lip Reading, International Journal of Pattern Recognition and Artificial Intelligence, vol. 14, no. 5, pp. 571–585, 2000.
[21] Wolff, G.J., Prasad, K.V., Stork D.G., and Hennecke, M., Lipreading by neural networks: visual processing, learning and sensory integration, Proc. of Neural Information Proc. Sys. NIPS-6, Cowan, J., Tesauro, G., and Alspector, J., eds., pp. 1027–1034, 1994.
[22] Nigay L., and Coutaz, J., A Generic Platform for Addressing the Multimodal Challenge, Proceedings of CHI'95, Human Factors in Computing Systems, ACM Press, NY, pp. 98–105, 1995.
[23] Cohen, P.R., Johnston, M., McGee, D.R., Oviatt S.L., Pittman, J., Smith, I., and Clow, J., Quickset: Multimodal Interaction for Distributed Applications, In: Proceedings of the 5th International Multimedia Conference, ACM Press, pp. 31–40, 1997.
[24] Johnston, M., Unification-based multimodal parsing, Proceedings of the 17th International Conference on Computational Linguistics, ACL Press, pp. 624–630, 1998.
[25] Wu, L., Oviatt, S.L., Cohen, P.R., Multimodal Integration – A Statistical View, IEEE Transactions on Multimedia, vol. 1, no. 4, pp. 334–341, December 1999.
[26] Elting, C., Strube, M., Moehler, G., Rapp, S., and Williams, J., The Use of Multimodality within the EMBASSI System, M&C2002 – Usability Engineering Multimodaler Interaktionsformen Workshop, Hamburg, Germany, 2002.
[27] Wahlster, W., Reithinger, N., and Blocher, A., SmartKom: Multimodal Communication with a Life-Like Character, Proceedings of Eurospeech, Aalborg, Denmark, 2001.
[28] Oviatt, S.L, DeAngeli, A., and Kuhn, K., Integration and synchronization of input modes during multimodal human-computer interaction, Proceedings of the Conference on Human Factors in Computing Systems (CHI '97), ACM Press, New York.
[29] Buisine, S., and Martin, J.-C., Experimental Evaluation of Bi-directional Multimodal Interaction with Conversational Agents, Proceedings of INTERACT, Zurich, Switzerland, pp. 168–175, 2003.
[30] http://www.hhs.net/tiscione/applets/ochre.html.
[31] http://www.ibm.com/software/speech/.
[32] Martin, J.C., Julia, L., and Cheyer, A., A Theoretical Framework for Multimodal User Studies. Proceedings of 2nd International Conference on Cooperative Multimodal Communication, Theory and Applications, Tilburg, The Netherlands, 1998.

[33] Martin, J.C., Veldman, R., and Beroule, D., Developing multimodal interfaces: a theoretical framework and guided propagation networks, In: Multimodal Human-Computer Communication. Bunt, H., Beun, R.J. & Borghuis, T. (Eds.), 1998.
[34] Béroule, D., Management of time distortions through rough coincidence detection, Proceedings of EuroSpeech, pp. 454–457, 1989.
[35] Reeves, B., and Nass, C., The media equation: How People Treat Computers, Television, and New Media Like Real People and Places, Cambridge University Press, Cambridge, 1996.
[36] Kaiser, E., Olwal, McGee, D.R., A., Benko, H., Corradini, A., Li, X., Cohen, P.R., and Feiner, S., Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality, In: Proceeding of the International Conference on Multimodal Interfaces, Vancouver (BC), Canada, pp. 12–19, 2003.
[37] Corradini, A., Collaborative Integration of Speech and 3D Gesture for Map-based Applications, In: Proceedings of the International Conference on Computational Science, Workshop on Interactive Visualization and Interaction Technologies, Lecture Notes in Computer Science 3038, Springer Verlag, pp. 913–918, 2004.
[38] Aleotti, J., Bottazzi, S., Caselli, S., and Reggiani, M., A multimodal user interface for remote object exploration in teleoperation systems, IARP International Workshop on Human Robot Interfaces Technologies and Applications, Frascati, Italy, 2002.
[39] Kraahnstoever, N., Schapira, E., Kettebekov, S., and Sharma, R., Multimodal Human-Computer Interaction for Crisis Management Systems, IEEE Workshop on Applications of Computer Vision, Orlando, Forida, 2002.
[40] Blatt, M., Grossman, T., and Domany, E., Maximal a-posteriori multi-sensor multi-target neural data fusion, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), can be downloaded from: http://www.weizmann.ac.il/~fedomany/papers.html.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Spatio-Temporal Data Visualization and Analysis for Multi-Target Tracking

Kiril ALEXIEV
*Central Laboratory for Parallel Processing – Bulgarian Academy of Sciences*

**Abstract.** Recently, the development of complex sensor networks has received considerable attention from the scientific society and industry, and many such systems have been implemented in different applications. Sensor networks consist of homogeneous or heterogeneous sensors spread in a global surveillance volume, acting joinly to optimally solve required tasks. Multiple target tracking is one of the most essential requirements for these systems; it is used to interpret an environment that includes both true targets and false alarms simultaneously. In spite of the great interest centered on fully automatic sensor data processing, the man-machine interface is nevertheless vital in such complex scenarios. This paper concerns spatio-temporal sensor data visualization and analysis for such complex systems, with a focus on components and data flow in sensor systems. Simplified examples illustrate the use of a developed graphic user interface and programming package for radar data processing.

**Keywords.** Multiple target tracking, spatio-temporal visualization and analysis, modeling, simulation, information fusion, man-machine interface

## 1. Introduction

In the last few years, the development of complex sensor systems has received considerable attention from the scientific society and industry, and many such systems have been used in different applications. Sensors systems detect, correlate and analyze events, activities and movements of objects of interest as they occur in time and space, determine the location, identity and status, assess the situation, and detect patterns in activity that reveal intent or capability. The quality of sensors has grown quickly and now a large variety of excellent low cost and lightweight sensors are available off the shelf. Advances in computer technology stimulate the progress of sensor intelligence. The high bandwidth and sensitivity of modern sensors can lead to a huge data load. For the design of real time target tracking systems, computation load becomes an important factor. However, the complexity of the developed target tracking system makes the analysis difficult. Different signal and data processing algorithms have to be considered such as signal detection, target tracking, data association, track fusion, etc.

Often, sensors work in hostile environments with the presence of false alarms and powerful noise and this increases additionally the quantity of generated data. One of the most essential requirements for surveillance systems employing one or more sensors is to interpret an environment that includes both true targets and false alarms simultaneously. This is the so-called Multiple Target Tracking (MTT) problem [1]. The most widely utilized form of multiple target tracking system is the radar track-while-scan system, which will be considered in this paper. In the track-while-scan system,

data is received at regular intervals as the radar regularly scans a predetermined search volume. An important issue in the multiple target tracking system is that it usually involves complicated data association logic in order to process and sort out the sensing information into useful information. The computational complexity of this difficult task is known to be NP-hard. This is substantially more costly than linearly increasing the effort for state estimation of a single target with the number of processed measurements.

In spite of the great interest centered on fully automatic data processing, the interface between man and machine remains very important in such complex scenarios. With advances in computing and visual display technology, the interface between man and intelligent sensors has become increasingly complex. The effectiveness of a modern interactive system depends on the design of interactive software package and its graphical user interface — GUI. Over the past forty years, advances in display and computing technology have revolutionized the interface between man and computer. Today, people can interact with rich, realistic, 3D graphics with relatively low cost equipment. Now the time has come to focus on designing our systems so that we maximize their capabilities in ways most effective for the user.

The main purpose of this paper is to describe a particular GUI solution for radar surveillance systems. The example system used is a radar data simulator. Simulators are training systems that display computer-generated scenes imitating real-world situations. Digital computer simulation is a valuable tool, also used for the design, analysis, testing and tuning of complex systems like the radar surveillance system. These activities are known to be a hard job in practice. The main reason for this is that in a surveillance system there are many design parameters and plenty of competing requirements to be met. The current radar data simulator is designed to create scenarios of multiple target environments and to study the computational effect and performance evaluation of the overall system.

The simulation includes the next three main steps: input data generation, modeling of examined systems and performance evaluation with proper visualization. GUIs found in simulator systems have laid the groundwork for completely computer-generated or virtual environments. A virtual environment is a computer-generated scenario that may seem real but it is not required to match any of the rules of the real world.

The paper does not concern problems of visualization as a choice of types of stereo displays like time-multiplexed glasses-based systems, free-viewing stereo systems like the Cambridge Autostereo Display, separate display panels for each eye, systems using holographic technology, oscillating mirrors or slice-stacking devices, etc. The paper does not discuss proper tuning of display parameters affecting spatio-temporal sampling like field-of-view, spatial resolution, refresh rate, and frame rate. The problem under consideration is how to organize information processing, information flow and visualization for timely and correct decision making.

The paper is organized as follows. The next section describes the main components of a system for multiple target tracking. The third section concerns target modeling and graphic user interfaces for complex scenario generation. The fourth section is devoted to the environment and jamming simulation and corresponding GUI. The fifth section depicts sensor and measurement errors modeling. The sixth section describes the principal algorithms for sensor data processing. The last section gives the conclusion.

## 2. Functional Description and Main Components of Simulation System

The simulation of radar data processing can be defined as a set of algorithms and data which allows:

- complex scenario generation, including a great number of different kind of simultaneously maneuvering targets (air targets, sea targets, ground targets), environment phenomena modeling like snowfalls, rain, fog, clouds, generation of noise and pulse jamming, passive clutter, radar signal propagation, shade zones, maps of reflected local objects, etc.;
- radar data input based on recorded real radar signals or calculated (simulated, modeled) radar signals, using generated space-time scenarios;
- low level radar signal processing – target detection;
- recognition of a pattern of successive detections as pertaining to the same target (track initiation);
- target state estimation (estimation of parameters like position, velocity and acceleration), thus establishing a so-called "target track";
- extrapolation of the track parameters;
- distinguishing different targets and thus establishing a different track for each target;
- information fusion, using several sensors;
- visualization on the radar display;
- real time interaction with operator's console and simulation of results and consequences of operator's interference;
- spatio-temporal synchronization of events;
- The main components of such a simulation system are described below.

### *2.1. Temporal Database*

A spatio-temporal data model is used to represent and analyze space-time information. Space-time information is a combination between any available and relevant a priori information, in order to enrich and refine the situation picture, and sensor information about dynamic targets and other objects in the surveillance volume. Focusing henceforth on ground and seaside based scenarios, the a priori information will typically consist of geographical data. This data model defines the manner in which spatio-temporal information has to be stored and retrieved. In some cases, time can be stored as an attribute in a database. In these cases some issues such as time duration, data retrieval, updating, storing, preserving historical data, etc., induce us to use special data structures for temporal databases. Object oriented and relational data structures are two major structures that are increasingly used in common databases in recent years. The spatial presentation often leads to Geographic Information System databases (GIS), which facilitates data presentation in the real world. If a temporal GIS does not have a good data model, the analysis of temporal information cannot be efficient. A number of researches have been undertaken in this field representing the feasibility of conversion from these kinds of databases to new temporal extensions.

### *2.2. Temporal Visualization and Representation*

Visualization and representation are among the most important stages in an information processing system. Because most end users of such systems may not have a detailed knowledge about all processing algorithms and analyses, systems with simple, fast, intelligent and deductive representation tools are increasingly requested. This point must also be considered in the design of spatio-temporal information systems. Some approaches that are offered for representation of temporal information have to allow for untrained operators, have to be intolerant to any incorrect actions, has to fully resemble a real sensor system with its control panel display and have timely response of operator's actions.

Visualization of temporal information can be considered from the following aspects:

- representation of changes in a measurement database;
- representation of changes in sensor operation mode;
- representation of changes in visualization mode;
- representation of changes in simulation mode.

A graphical user interface has to provide a convenient environment with all the necessary tools. The GUI visualizes instantly generated scenario elements and gives comprehensive information about it. The GUI has to be equipped with modern means for creation and edition of a complex scenario with thousands of elements.

### *2.3. Signal and Data Processing and Analysis*

This component includes a large variety of algorithms used for signal and data processing. Often this set of algorithms is cited as a suite or library of processing algorithms. The suite of processing algorithms in a simulator consists of the algorithms implemented in the simulated sensor system. Sometimes the simulation system is used with new algorithms or for tuning of already embedded algorithms for different conditions of sensor use. In these cases, it is necessary to add powerful algorithms for different statistical measures of performance, which give developers a clear outcome about tested algorithms.

### *2.4. Hardware Specialized Processing Units*

This component is not always mandatory. Modern high-power high-frequency microcomputers manage to execute a billion FLOPs for solving real time tasks. However, a radar sensor system generates intensive signal and data flow. A proper signal is generated for every elementary radar volume. Let us consider that a radar sensor has 5000 cells for every azimuth (distance cells) and 5000 cells for every distance (azimuth cells). If there are four elevation beams, the total number of cells is 100 million. The signal intensity varies over approximately 80 dB. The generated information is approximately 200 MB per scan (scan rate between 1–5 seconds). This example shows that a huge volume of information is generated in real time. If the necessary input information rate surpasses the capacity of a standard computer processor unit, it is necessary to develop a specialized hardware processing unit. This unit accelerates the generation of signals and data and makes real time implementation possible.

## 3. Target Modeling

There are two different approaches for the simulation of input data. The first approach uses scenarios with real participants, acting as targets and data, recorded from real radars observing this scenario. In this case no errors caused by data modeling are inserted. However, there is a severe drawback – it is very difficult, dangerous, expensive and sometimes impossible to explore estimated algorithms in a complex scenario. Even if a scenario has a low probability, it can exist in real life critical conditions. Another drawback is that the true target path and the true target maneuver parameters are unknown and the researcher does not have exact reference data for an accurate evaluation of the explored algorithm [2]. Nevertheless, a simulation tool usually has an entry for real life data, using a common record format for data exchange.

The usage of simulated data has considerable flexibility in the selection of complex target and clutter scenarios and an a priori known reference input is provided. The simulation program generates hundreds of targets moving rectilinearly or maneuvering with given transversal and longitudinal accelerations.

The radar parameters (scan rate $T$ and detection probability $P_D$) can vary in wide intervals.

The simulation program has the ability to synchronize the position of generated targets in the space and thus to create complex and critical scenarios. However, only an approximate representation of the operating conditions can be obtained.

The trajectory generator is generally better suited for algorithm estimation and tuning. Recorded real sensor data can be used as a more realistic test in an advanced stage of the design or as a last approval of system characteristics.

The main problem in trajectory generator design is how to organize data input to create a complex scenario easily without any mistake and deep knowledge about modeling algorithms. This part of the GUI was realized as follows.

At the very beginning, the operator has to choose the class of the target and concretize its type in this class using the main menu of the radar simulator (figure 1). The operator's choice of a target type defines strictly realistic ranges of the target's parameters. To specify the values of these parameters, the operator has to choose one of available List Of Value (LOV) for each parameter. Default values are set to determine moving targets with usual velocity, no longitudinal acceleration and minimal maneuvering acceleration. The choice of transversal acceleration of an air target is displayed on figure 2.

The trajectory is assumed to be in 3D space for air targets and planar for sea targets. A trajectory is modeled by straight-line sections and maneuvering (circular) sections.

It is considered that the easiest way to define a trajectory is to point with the mouse, and click the key points of the target track on the geographic (or other type of) map. Key points on the map define these sections implicitly. Every point characterizes target state by target position, target speed, target transversal acceleration and time [3]. Generally, the direction of the target track is changed at every key point (except the first and last ones). The maneuver is considered with transversal acceleration, corresponding to the chosen value in the target parameter list for this point. All spatio-temporal target parameters for maneuvering sections can be calculated, solving a simple system of a few equations. When the system is solved (the system may be not solved if unreal conditions are inserted), the new key points are determined and straight line and maneuvering sections are explicitly defined.

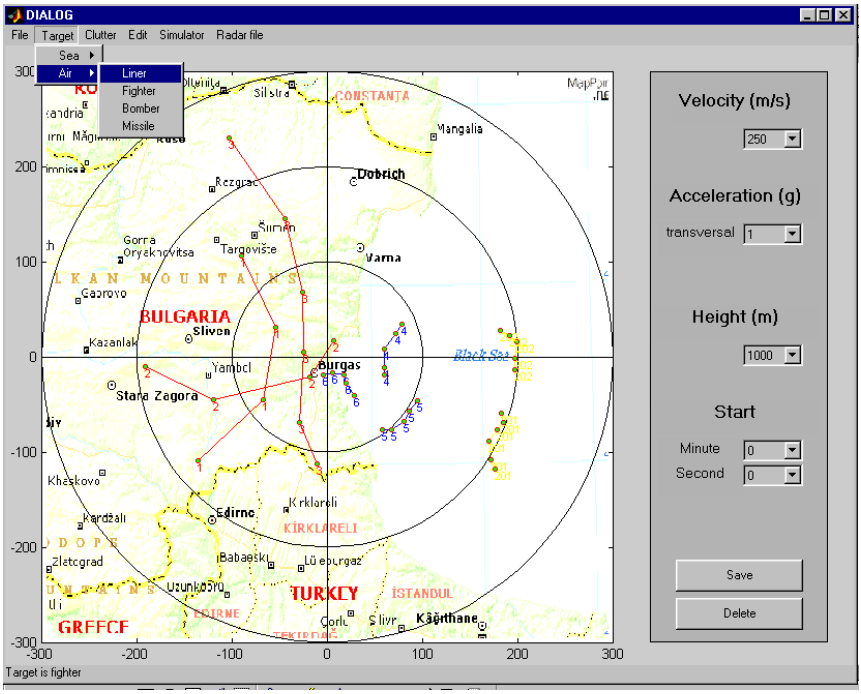



**Figure 1.** The graphic user interface of target simulation program.

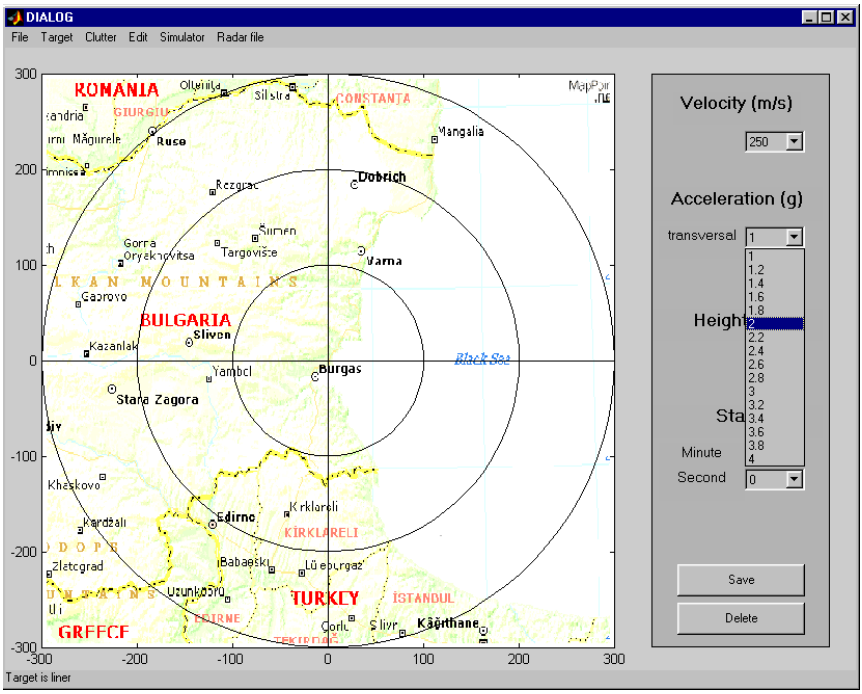



**Figure 2.** Parameter setting for an air target.

Target motion is modeled by computing the position, velocity and acceleration at the moment of radar wave reflection. The time interval between two consecutive detections of a target may differ slightly from the radar scan rate. A special algorithm for the exact calculation of target position in the moment of radar wave reflection is implemented. The antenna scan rate modulation, which is commonly encountered in practice (e.g. due to the wind), is not considered.

If the target is air based, it will be necessary to point additionally the target's altitude at these key points.

The operator's last action is to save the new scenario.

The operator may change the target's parameters. To do this, he deletes invalid key points and sets new ones with proper new values of parameters.

An example for an air target with smooth space trajectory and smooth velocity change is shown on figure 3. The target speeds up to "North-East," turns with small transversal acceleration (big turn radius) and then it speeds down to "South-East" and near the state border turns to the "East" with middle transversal acceleration (smaller turn radius).

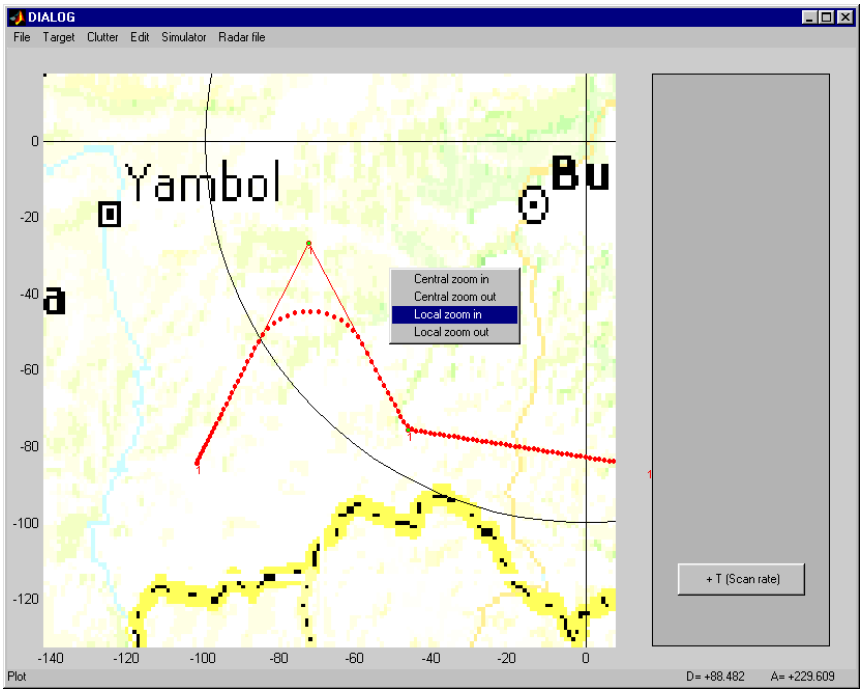



**Figure 3.** Target trajectory consists of rectilinear sections and maneuvering section.

Often, when the operator points to sites on the map, more accuracy is needed. Two types of magnification – local (around the current marker position) and central (sensor remains in the middle of the map) are available in these cases. The central magnification is used to preserve a panoramic field of view around the sensor. The central magnification loses detail when placed remotely from the sensor. To zoom into remote zones without moving bars and tediously search a place of interest, local magnification is added. Local magnification visualizes details of interest and moves them on the cen-

ter of the display. In this way, we can very easily create a complex multiple target scenario.

Nevertheless, there is a big disadvantage – all targets are created in the space. The temporal relations between targets remain unknown. The targets have to be synchronized in time and in space. The critical moment in a scenario is to place many targets in a small area at a given moment of time. The solution of this problem is found by dividing it into two independent steps. The first step includes determining trajectories in the surveillance volume. It is considered that the operator can place targets at the right place in the space. Every target is placed in the time space with its start time (the time, when the target appears on display – in the surveillance volume). The calculation of maneuvering parts from trajectories and the correction of trajectories due to smoothing is done on the base of the trajectory start time and trajectory key points. This allows targets to move with reasonable speed, but it affects the exact positioning of targets at the suitable moment of time in the right place. The second step synchronizes targets in time space on the base of their already known trajectories. This is realized by simulation mode and editor mode (figure 4).

In simulation mode, the operator observes a target's movement in time and he can easily estimate if a target needs to be started earlier or later and the exact time of target start. This is a compromise for the solution of the problem, when temporal changes in 3D spatial surveillance volume have to be projected onto a 2D display. To allow the possibility of correcting a target's parameters an additional mode "Edit" is included in the simulation package (figure 5). In this mode, the operator can change the temporal sequence of observed events. Repeating these two steps the operator tunes the target's interaction in the scenario. Every intermediate scenario can be saved and used as a start point of another scenario. The synchronization of clutter can be realized in the same way. The only difference is that the clutter has additional an "End time" parameter. In this way, a complex spatio-temporal scenario can be designed with the participation of hundreds of targets.

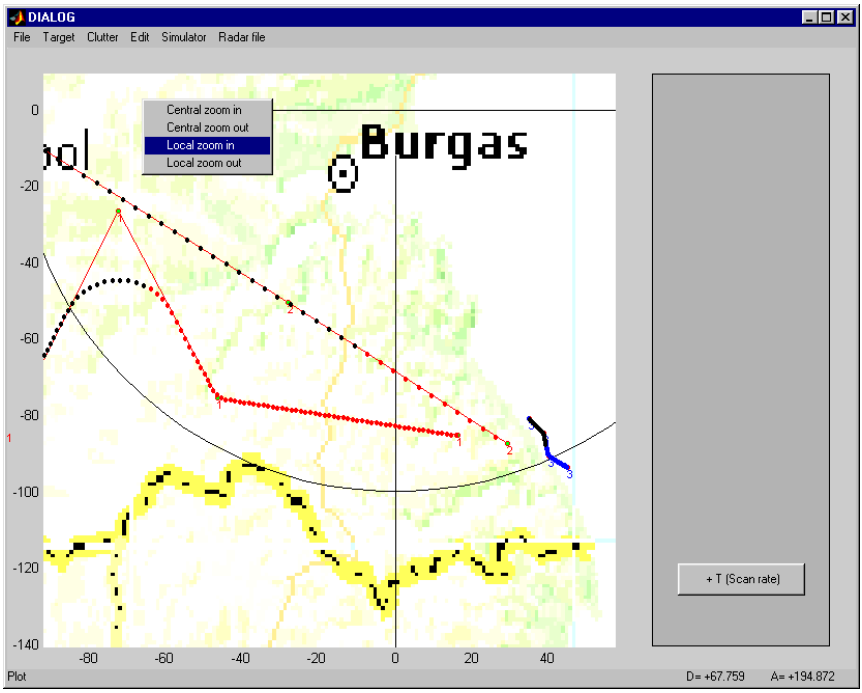



**Figure 4.** Simulation mode. Operator can observe spatio-temporal movement of targets scan by scan and synchronize targets. The grey points depict target tracks, the black points depict already received measurements.

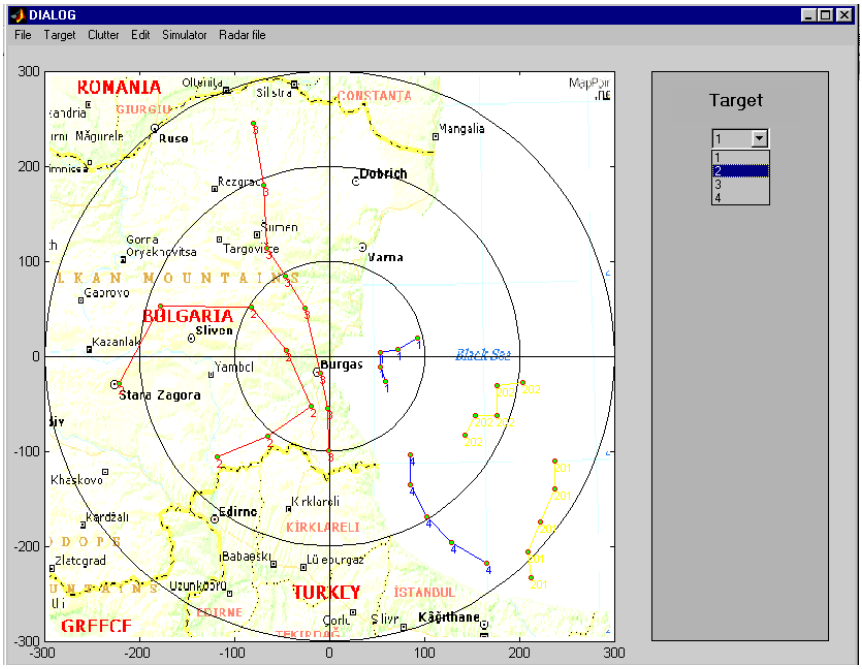



**Figure 5.** Editor mode. Operator chooses to edit target, identified as “1”.

## 4. Clutter Modeling

Clutter can have a natural or artificial origin. It is easy to explain clutter in the case of imaging sensors. The received image consists of targets of interest and background. The targets appear, move and disappear above a constant or variable background. Usually, this background has a natural origin. The environment clutter also includes weather conditions – rain, snowfall, the influence of the sun, moon or stars, water vapor, lighting conditions, etc.

In military radar applications, the adversary targets generate active countermeasure return signals (figure 6) or spread passive reflectors to deceive the sensors.

Radar, operating in a hostile environment, may be subjected to noise jamming. Noise jamming is generated by adversary broadband white noise in the band covered by the radar. The effect of noise jamming is expressed in decreasing sensor characteristics. Noise jamming decreases the radar's range capability because the total received noise is the sum of the noise jamming and thermal noise. The increased noise power results in a decreased SNR, and thus a decreased maximum detection range. The noise jamming generator can be carried on a penetrating aircraft or a non-penetrating support aircraft as a standoff jammer. A major difference between noise jamming and thermal noise is that noise jamming emanates from a single spatial angle, whereas thermal noise has an essentially uniform density over all spatial angles.

The presence of a large false alarm rate overwhelms the radar data processor and deteriorates sensor characteristics. Usually, the false alarm rate due to noise cannot be too large. The effective false alarm rate can be increased by an enemy jamming that transmits replicas of the radar's pulses (so called pulse jamming). The received pulses will appear to be targets. Often the replicas are received through the radar's antenna side lobes, and thus the angular location of false targets appears to be very different from that of the jamming source. Side lobe false target jamming can create many false targets, with each at apparently different angular positions. The large number of false targets leads to computers overloading and prevents the successful tracking of true targets.

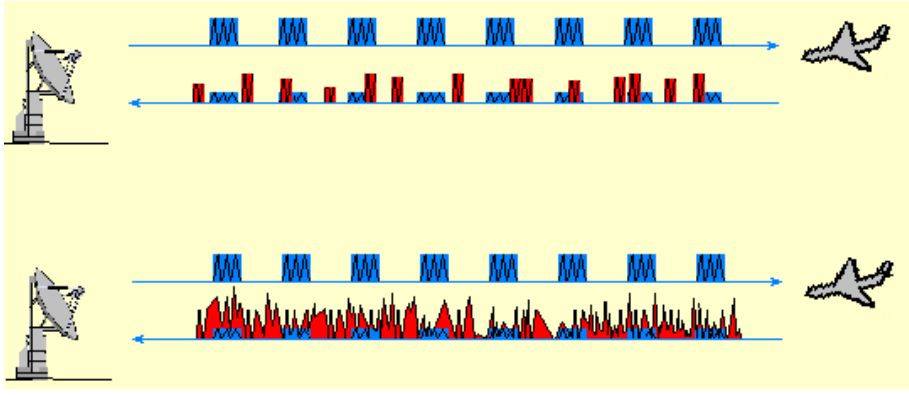


**Figure 6.** Pulse jamming and noise jamming.

Noise and clutter from natural and artificial origins are modeled by corresponding programs. The input data generator for a radar surveillance system generates several types of natural clutter like clouds, rain and snowfall, several types of passive jamming

like dipoles, clouds of reflecting bands and several types of pulse and jamming noise, generated by adversary transmitters.

## 5. Radar Sensor Model

The main function of every sensor is to give a true picture of measured (observed, examined, watched) areas of interest. A sensor can be modeled in two different ways. The first requires a detailed simulation of every sensor unit. This type of modeling is more difficult and troublesome. The units are modeled by their functionality. The result is good enough only if all simulated sensors units work correct. A small mistake in the modeling of a unit leads to wrong results when it is amplified by other subsequent sensors units. Another type of modeling is black box modeling, where the sensor is presented by its transition function. In this type of simulation, inner sensor structure is absent and the interaction between units could not be revealed. The simplicity of this approach makes it very attractive and it was used in the simulator.

A radar operates by transmitting energy into the environment and obtaining information concerning the location of objects by detecting reflection energy. The classical radar equation describes the relationship between radar parameters and returned signal power $P_R$, reflected by the target. A possible form of this fundamental equation is:

$$P_R = \frac{P_T G_T G_R \lambda^2 \sigma}{4\pi \left(4\pi R^2\right)^2} \tag{1}$$

where $P_T$ is transmitted power, $G_T$ and $G_R$ are power gains of transmitter and receiver antennas, $\lambda$ is radar wave length, $\sigma$ is target backscattering coefficient (radar cross section) and $R$ is the distance between antenna and target. The radar sensor model is based on this equation with the following particular features.

### *5.1. Measurement Error Simulation*

In surveillance track-while-scan systems which are simulated, the statistic of the quadrature components of the radar clutter was supposed to be jointly Gaussian because of the low radar resolution capabilities. In this case, the clutter is viewed as a sum of responses from a very large number of elementary scatterers and, in accordance with the Central Limit Theorem, the statistic of the additive noise can be regarded as Gaussian [1]. This model is used in the radar modeling. However, with the advancement of radar resolution, the statistic of the additive noise is no longer observed to be Gaussian. A new model of noise distribution has to be used. It has both higher tails and a larger standard deviation to mean than predicted by the Rayleigh distribution.

### *5.2. Calculation of Measurement Time*

For a surveillance radar with a constant scan rate for a given radar mode, the exact moment of receiving the return signal from a target can be calculated from two differential equations. The first equation describes target movement and the second concerns

radar beam rotation. A simple recursive program is used in the simulator for the calculation of this moment with given accuracy. A short description of this algorithm follows for scan $k$ and scan rate $T$ (figure 7):

1. let us express the time of beam at the position "North" for scan *k*: $t_R = (k-1)T$;
2. now we can find the target coordinates $C_T(t)$ for $t = t_R$;
3. we assign $t_{R_{old}} = t_R$. Calculate the new radar beam time $t_R$, when the beam overlaps target coordinates $C_T(t)$;
4. if the difference $\delta t = \left| t_{R_{old}} - t_R \right| < \varepsilon$ is small enough – end of procedure, else repeat step 2 and 3.

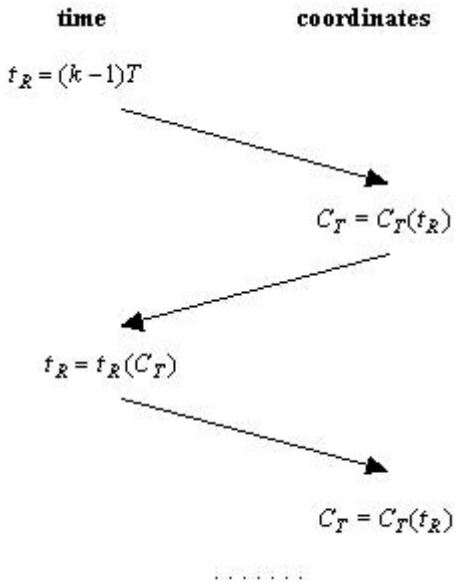



**Figure 7.** Calculation of measurement time.

*5.3. Return Signal Calculation*

The return signal is modeled using a classical radar equation in three forms (figure 8). The first is identical to the classical radar equation. This equation is used for modeling of returned signal power from targets in the zone, where the interference between reflected signals from target and ground can be disregarded. The second form is used for modeling the returned signal power from targets which are in the interference zone. The third model is used for targets in the diffraction zone near the radio-horizon. Every target is checked as to which zone it belongs, and then the return signal power is calculated by algorithms, described in [4] for the corresponding zones.

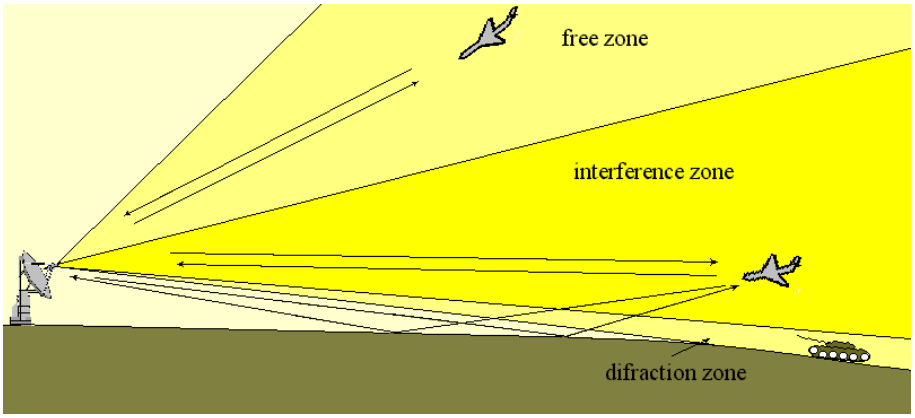



**Figure 8.** Zones of wave propagation.

### *5.4. Antenna Pattern Modeling*

The performance of an antenna is characterized by its directive properties. The measures of the directional characteristics are the antenna receiving pattern and its radiation pattern [5]. For one and the same antenna both patterns are identical and unique for a particular aperture. Usually patterns have a broadened powerful mainlobe and lower adjacent sidelobes (figure 9). For a given aperture, the radiation pattern is measured and included in the antenna model.

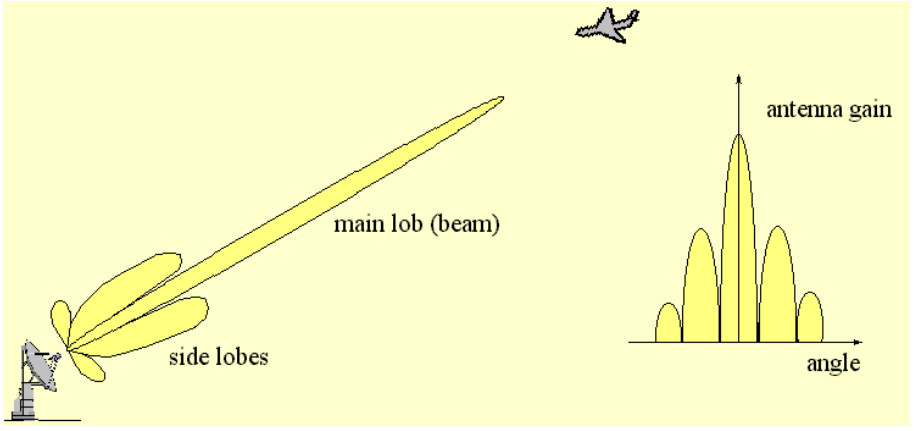



**Figure 9.** Zones of wave propagation.

## 6. Processing Algorithms

The suite of simulated sensor information processing algorithms reproduces real time sensor data processing algorithms, which allows the formation of estimated target states based on incoming measurements provided by sensors. When the simulator is used to verify a newly developed version of a processing algorithm, it is better to have the possibility to compare it with basic tracking algorithms. That is the main reason for creating a library with different processing algorithms.

The processing algorithms can be grouped as signal processing algorithms, data processing algorithms and collaborative information processing algorithms.

### *6.1. Signal Processing Algorithms*

Signal processing algorithms filter noisy raw data and locate, detect, or recognize targets of interest. They have to reject all unnecessary information. The choices of detection threshold and correlation detection algorithm determine the characteristics of the whole sensor system. For example, for a radar sensor system, the low detection threshold reduces the possibility of undetected targets, but the larger volume of data requires more effective track initiation and estimation algorithms. High detection threshold increases the probability of losing useful information.

### *6.2. Data Processing Algorithms*

Advanced multi-target/multi-sensor data processing algorithms have to possess the following characteristics:

- real time data processing;
- variable revisit rate;
- robust processing of weaker or noisy targets at lower SNR (low value of $P_D$ );
- continuous processing of maneuvering (unpredictably moving) targets;
- multiple sensor information fusion (possibility to create a common tactical picture of the surveillance volume of several sensors).

#### *6.2.1. Track Initiation Algorithms*

A track is an estimate of a target's kinematics, including such factors as its position, velocity, and rate of acceleration. Track initiation represents the initial step, necessary for localizing a target; this can later be enhanced with the identification of other characteristics associated with the target. This step is not always needed. There are tracking methods like Multiple Hypotheses Tracking, which have embedded techniques for track initiation. However, most methods require such an initial step. The fundamental challenge is the task of deciding which observations should be correlated and combined into track estimates. Different correlation techniques are used to associate sensor measurements with potential track trajectories (figure 9). The task is to find several measurements ordered in space and time. The most common approach is the classical one, which uses $N$ sequential measurements (usually $N$ equals to 4–5) and implements weighted least squares to find an initial approximation of target state. The classical approach can be a very computationally intensive process, because the number of hypotheses grows exponentially with the number of measurements under consideration. This hypothesis growth can be overcome by careful hypothesis pruning. A gating technique is introduced in order to reduce the combinatorial problem, but this algorithm does not solve the problem completely. In dense target and clutter environments, however, the number of hypotheses remains big enough and the classical approach leads to poor results and can't initialize the trajectories. In this case, another type of track initiation procedure has to be used. The vast surveillance volume is fragmented in a set of cells and the combinatorial problem is decomposed on many such problems of lesser size, solved in small fragments. Two types of such track initiation procedures are

implemented. The first of them uses uniform surveillance volume fragmentation. The measurement selection method typically uses a mosaic grid to group the measurements in subsets. The track initiator uses these subsets for potential track determination. The problems of optimization in this case are determination of cell size and how to process measurements on (or near to) the cell borders. The second algorithm uses template matching techniques like the Hough transform [6], Fourier transform, etc. The cells in the surveillance volume in this case correspond to the initialized target trajectories. Both methods require additional computer resources to resolve the combinatorial problem in the case of dense target and clutter environments.

The main parameters, which have to be estimated, are the probability of detection of a trajectory, the probability of false track detection as a function of a number of considered measurements $N$, radar probabilistic characteristics like $P_D$ and $P_{fa}$, the gate's size, cell or template, etc.

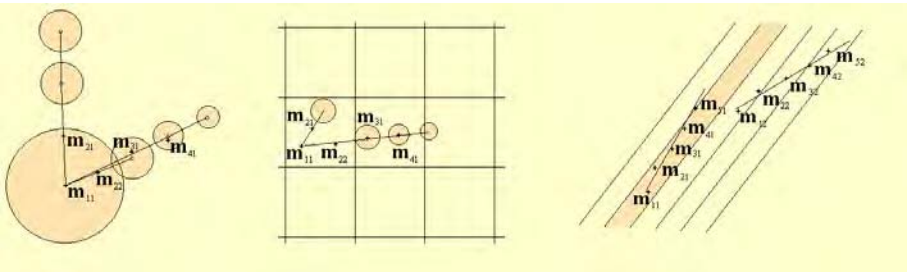


**Figure 10.** Track initiation algorithms.

### *6.2.2. Track Estimation Algorithms*

Modern surveillance systems using radars as sensors require rapid and highly accurate data to be subsequently processed. Location, velocity, maneuver and possible identification of each target of interest can be provided by radar data processing with an accuracy and reliability greater than that available from single-look radar reports. Furthermore, radar data processing can enhance the signal-processing function by removing false detections caused, for example, by residual clutter.

Several estimation procedures are implemented in the described simulation tool. They use $\alpha-\beta$, $\alpha-\beta-\gamma$, Kalman and extended Kalman filters in different realizations. All procedures have to improve estimation accuracy. Today, only in Europe more than 30 different procedures for tracking are currently in use [7]. Some of them use the same filters but work with different coordinate systems (polar and orthogonal) or state vectors have different lengths. Many of these filters are available for use in the subroutine library.

Classically, plots are associated by a "nearest-neighbor" algorithm to the potential tracks. Wrong nearest-neighbor assignments, however, cause tracking filter divergence. Such a problem arises when there are false alarms in the target's gate or in the case of closely spaced targets.

The problem of the presence of false alarms can be resolved by a probabilistic data association (PDA) algorithm [8]. This is a basic algorithm for plot-to-track association, which uses all measurements in the target's gate. PDA allows more than one measurement to be associated to a track, each with a different probability and a corresponding

weight, according to its distance to the target prediction. The PDA filter is a very simple and robust algorithm against false alarms.

The Joint Probabilistic Data Association (JPDA) algorithm is another advanced technique, implemented as a tracking algorithm in the simulation tool. This algorithm resolves the case of closely spaced targets with common measurements in the overlapping gates. In this case, measurement-to-track association for one track cannot be done independently of other tracks in the cluster (a cluster is a set of closely spaced tracks). Joint means that all possible measurement-to-track combinations have to be evaluated. Furthermore, the track state vector update in principle must be done jointly. Through appropriate approximations in the JPDA algorithm, the latter is fortunately not necessary. Still the complexity of JPDA grows exponentially with the number of tracks and measurements involved in the resolution situation. The main advantage of JPDA is that the track quality can be maintained at a high level even in complex situations. Several modifications of this algorithm were realized and estimated [9,10].

The Interacting Multiple Model (IMM) filter is a robust filter, used for maneuvering target tracking. It assumes that a target is in a state corresponding to one of a number of modes of movement, each of which may be modeled by its own equations of motion. This approach uses several filters. Each filter corresponds to a mode of movement of the target. All filters process each measurement. The particular filter innovation and the probability of holding target in (or moving target to) this mode define the weight of a particular filter estimate onto the common estimate. In the next interaction step, the information from all particular filters is combined and fed back into the filters. The choice of filters and its suitable parameters remains a difficult problem to solve. It is obvious that robustness of IMM filters is received at the expense of estimate accuracy. For example, if a filter matches exactly with target motion mode, its estimate is deteriorated by influence of the other filters, which gives poor estimates. Another disadvantage of the IMM filters is increased computational complexity. The IMM filter may also be used in conjunction with PDA filters [8] and JPDA filters [11,12]. The researchers have on hand several versions of the algorithms described above in the described tool library.

### *6.3. Collaborative Information Processing Algorithms*

Contemporary multi-sensor systems are used to eliminate limitations in the exploitation of sensors in stand-alone mode. In this mode, traditionally, radar information processing algorithms are reviewed in terms of a target's state estimation and identification with necessary accuracy. Today, when the radar is often a part of an integrated sensor system which shall cooperate with other sensor systems, the stress is on collaborative information processing algorithms. The main purpose of these algorithms is to take advantage of the synergy in the combined information acquired from multiple sources. Sensor collaboration technology must address ground, airborne and spaceborne systems and processes in a fully distributed environment. Collaborative information processing algorithms treat multi-sensor information in order to provide highly robust, reliable and precise information about regions of interest. Sensors networks consist of homogeneous or heterogeneous sensors spread in a global surveillance volume, which act jointly for solving required tasks optimally. The sensors may include video cameras, acoustic microphone arrays, thermal imaging systems, seismic or magnetic sensing devices, microwave or millimeter wave radar systems, laser radar systems, etc. Contemporary computers and communications make possible automation in data proc-

essing and fast and reliable information exchange. Sensor networks combine these two technologies. The implementation of sensor networks can already be found in a variety of civilian and military applications, including factory and home automation, environmental monitoring, health-care delivery, security and surveillance, and battlefield awareness. Several advantages make sensors networks very attractive in many applications:

- the overall system is more robust against the failure of a number of its sensors or communication channels (reliability, survivability);
- the information from many homogenous sensors increases the accuracy of estimation;
- the information from many heterogeneous sensors increase the coloring of the received picture and this plays an important role in target identification and correct situation reaction;
- there is the ability to cover a wide surveillance area, which is impossible for one-sensor systems.

There are two main aspects of collaborative information processing algorithms. The first considers efficient managing of the detections or tracks, provided by the different radar sets of a netted system looking at the same portion of the controlled space, in order to provide a better picture of the latter. The second discusses information exchange between sensors.

Several years ago, almost all multi-sensor systems of practical interest worked with a common processing center which concentrated all information from available sensors. Currently, there is a revolution of "distributed" information processing. Distributed ways of communicating, processing, sensing, and computing are dislodging more traditional centralized architectures. Contemporary communication allows a low cost and robust exchange of enormous information over long distances. Powerful processors treat information from more than one sensor. A new generation of distributed algorithms for target initiation, estimation and tracking appears in scientific journals. The modeling of collaborative information processing is aimed at the optimization of a set of sensors and communication channels for different purposes. Some systems are used in hostile environments where the minimal communication time is the optimization criterion. Another criterion can be maximal accuracy of target parameter estimation or minimal target identification time.

## 7. Conclusions

This paper concerns information flow, analysis and visualization in digital computer simulations of surveillance radar sensors acting in dynamically changing environments. The multi-target case is considered. The principal tasks, which arise in simulation of such complex systems, are described. Special attention is devoted to spatio-temporal sensor data visualization. The appropriate design of graphic user interfaces increases effectiveness and convenience of simulators in designing and building simulations. Such an organization of GUI and data flow offers a significant productivity increase through parametric design entry, advancements in data management and design automation, and state-of-the-art algorithms. This is also due to a faster development of scenarios and models, extensive reuse of scenarios, models, and program modules. The extensive suite of algorithms is presented for solving different tasks. Described algo-

rithms are not a "panacea" for all relevant issues, but the algorithms are a good base for further development and improvement. Several themes are only touched upon and further research is needed. They are not included in the presented version of the simulator, but it is considered that in its revised version most of them will be included in the algorithm suite. One of them is the distributed sensor system. It is considered that this interesting topic will be in the center of interest in the next several years. The presented simulating system has two directions for development. The first concerns training tools. The second is a testbed tool for analysis and design purposes.

## Acknowledgements

The research reported in this paper is partially supported by the Bulgarian Ministry of Education and Science under grants I-1205/2002 and I-1202/2002 and by Center of Excellence BIS21 grant ICA1-2000-70016.

## References

[1] S. Blackman, Multiple Target Tracking with Radar Applications, Artech House, 1986.
[2] A. Farina, F.A. Studer, Radar Data Processing, John Wiley & Sons inc., 1985.
[3] K. Alexiev, E. Djerassi, L. Bojilov, "Flight object modeling in radar surveillance volume," Sixth international conference: "Systems for automation of engineering and research, SAER'92, St. Konstantin – Varna, Bulgaria, pp. 316–320, 01–03, 1992, in Bulgarian.
[4] K.W. Golev, Estimating Radar Range, "Sovietskoe radio," Moscow, 1962 (in Russian).
[5] R. Nitzberg, Adaptive Signal Processing for Radar, Artech House, London, 1992.
[6] K.M. Alexiev, "Implementation of Hough Transform as Track Detector," Proc. of the International Conf. On Multisource Multisensor Information Fusion, FUSION, 2000, pp. ThC4-11–ThC4-16.
[7] ARTAS, European Organization for the Safety of Air Navigation (EUROCONTROL), 2001.
[8] Y. Bar-Shalom, Multitarget multisensor tracking: Advanced applications, Artech House, 1990.
[9] L.V. Bojilov, K.M. Alexiev, P.D. Konstantinova – "A particular program realization of JPDAF algorithm," Comptes rendus de l'Academie bulgare des Sciences, Tome 55, No. 9, pp. 37–44, 2002.
[10] P. Konstantinova, K. Alexiev, "A Comparison of Two Hypothesis Generation Algorithms in JPDAF Multiple Target Tracking," Proc of the International Conference on Computer System and Technologies, pp. II.18-1–II.18-5, 21–22 June 2001.
[11] L. Bojilov, K. Alexiev, P. Konstantinova – "An Accelerated IMM JPDA Algorithm for Tracking Multiple Maneuvering Targets in Clutter," Information & Security An International Journal "Sensor Data Fusion," Vol. 9, 2002, pp. 141–153.
[12] K. Alexiev, "A MATLAB Tool for Development and Testing of Track Initiation and Multiple Target Tracking Algorithms," Information & Security: An International Journal "Sensor Data Fusion," Vol. 9, 2002, pp. 166–174.

# 3. SYSTEMS AND ARCHITECTURES

This page intentionally left blank

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Principles of Systems Engineering for Data Fusion Systems

Alan STEINBERG
*Space Dynamics Laboratory,Utah State University, 1675 North Research Parkway, North Logan, Utah 84341-1947, asteinberg@sdl.usu.edu*

**Abstract.** Methods are presented for the cost-effective development and integration of multi-sensor data fusion technology. The key new insight is in formulating the system engineering process as a resource management problem. Effectively, system engineering is formulated as a problem of planning under uncertainty, where the probability of outcome of various actions, the utility of such outcomes and the cost of actions are posed as time-varying functions. The fusion-specific problem is that of decomposing system operational and support requirements into requirements for a network of processing nodes, involving functions for data alignment, data association and state estimation.

**Keywords.** Data fusion, system engineering, resource management, actions, utility, data alignment, data association and, state estimation

## 1. Data Fusion Engineering

The most widely-used model for data fusion is that developed by U.S. Joint Directors of Laboratories (JDL) [1,2]. Per the current version of that model, we define data fusion as a process of combining data or information to estimate or predict entity states [2]. So-defined, data fusion pervades virtually every automated approach to the use of information, as well as all biological cognitive activity.

### *1.1. Project Correlation Data Fusion Engineering Guidelines*

The present work builds on a set of Data Fusion Engineering Guidelines that were developed in 1995–96 as part of the U.S. Air Force Space Command's Project Correlation [3,4]. The Guidelines were developed to provide

- a standard model for representing the requirements, design and performance of data fusion systems; and
- a methodology for developing multi-source data fusion systems, selecting among system architecture and technique alternatives for cost-effective satisfaction of system requirements.

Recognizing the common elements across the diversity of data fusion problems can provide enormous opportunities for synergistic development. Such synergy – enabling the development of information systems that are cost-effective, reliable and trustworthy – requires commonly understood methods for performance evaluation, system engineering methodology, architecture paradigms, and models of the character-

istics and behaviors of the applicable sensor systems and of sensed entities (i.e. targets).

Integral to the Guidelines is the use of a functional model for characterizing diverse system architectures and processing and control functions within a data fusion process. This architecture paradigm has been found to successfully capture the salient operating characteristics of the diversity of automatic and manual approaches that have been employed across a great diversity of data fusion applications [5,6].

The Guidelines recommend an architecture concept that represents data fusion systems as comprising one or more networked processing nodes. Generally speaking, each fusion node is amenable to representation as in Figure 1, performing some or all of the following functions:

Correcting received data for estimated spatio/temporal and other measurement biases and normalizing measures of confidence in the received data.

Data Association:

- hypothesis generation: selecting batches of sensor data that feasibly refer to the same entity; generally performed either by clustering the input data or by state prediction from existing association hypotheses (generally termed tracks);
- hypothesis evaluation: evaluating the likelihood that batches of data are so associated;
- hypothesis selection: determining which of the candidate hypotheses are to be used in subsequent processing; e.g. to infer entity states.

State Estimation:

- inferring, on the basis of selected association hypotheses the corresponding entities' existence and state; e.g. their locations, kinematics, identity, classification, attributes, behavior, intent, situational context, etc.

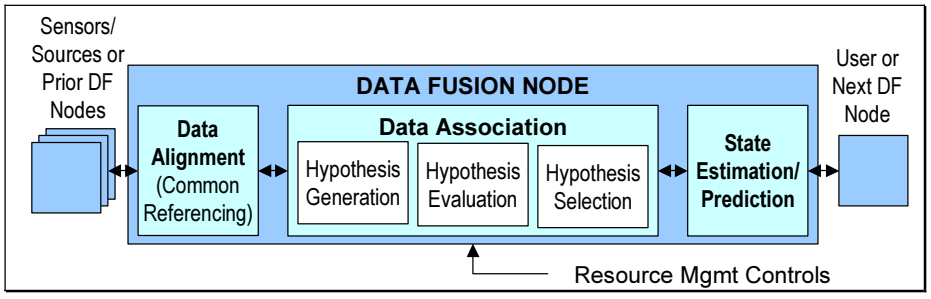



**Figure 1.** Data fusion node paradigm.

When the data fusion process is partitioned into multiple networked nodes, the process is represented via a data fusion tree, illustrated in Figure 2. Indeed, the figure illustrates the common tight coupling of data fusion processes (in nodes labeled 'F') with resource management processes (in nodes labeled 'M'). Fusion and management trees are so-called because they often involve hierarchical branching as shown. Other network topologies are, of course, possible. Such tight coupling of an in-fanning data fusion (DF) tree with an out-fanning resource management (RM) tree is characteristic of many successful system designs [6,7].

As seen in Figure 2, fusion nodes accept information from one or more sources: from sensor sources or from other fusion nodes. Such information is aligned and associated and combined in the form of estimates of the states of entities perceived from the input data.

The process of generating, evaluating and selecting fusion trees is an integral phase of the data fusion system engineering process, discussed in Section 3 below. The fusion tree should be designed so as to combine source information streams that are expected to reduce the incoming data most effectively.

A fusion tree architecture represents design decisions concerning:

- opportunities to exploit commensurate measurement data locally, converting measurement data to abstractions that are commensurate across different data streams (e.g., in combining data from imagery sources before combining imagery with signals intelligence);
- expectations of greater marginal gain in combining specific data streams; e.g., multi-senor tracking will generally converge most quickly if data are combined from the sensors with the most accurate location and tracking performance before adding in contributions from the less accurate sensors. Similarly, multi-sensor target detection and identification processes will tend to be most stable if high confidence sources are exploited in the process (i.e. at the initial nodes of the fusion tree);
- opportunities to reduce the burden on downstream nodes that involve a higher degree of complexity (e.g., in performing attributive-based report-to-target data association and fusion before performing relation-based situation assessment).

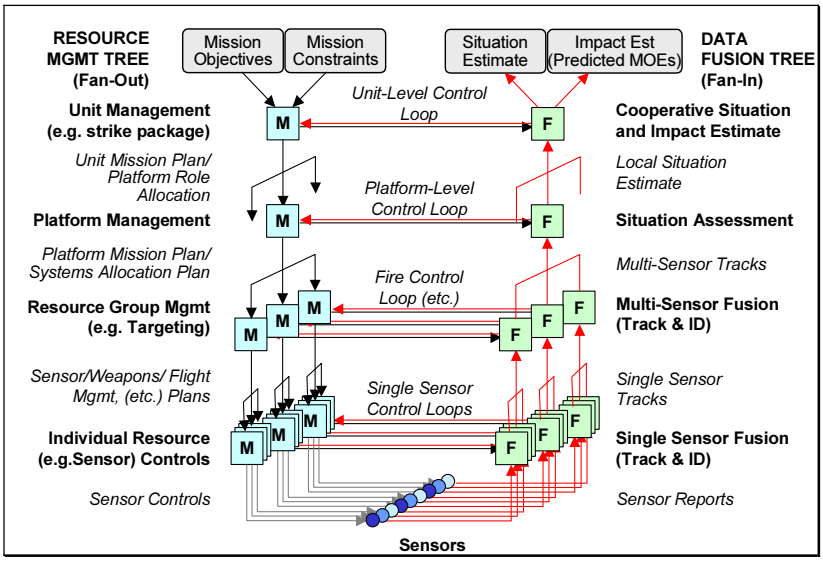



**Figure 2.** Integrated data fusion and resource management (DF/RM) tree architecture for a generic weapon platform.

Also seen in Figure 2 are interconnections among the nodes of the fusion tree and the resource management tree. The design may include control loops governing the

processes within individual fusion nodes (shown as horizontal arrows between pairs of fusion and management nodes). These can include such controls as:

- selecting batches of data for processing;
- selecting among algorithms within the fusion node (e.g., increasing the dimensionality of a tracker's covariance matrix when a target entity enters the atmosphere from space);
- controlling validation gates in hypothesis generation;
- determining the depth or breadth of search in hypothesis selection;
- controlling pruning of hypotheses (etc.).

The management tree also integrates the control of the fusion process into the overall system management and mission management processes. This permits coordinated use of all system resources: sensors, weapons, processing, guidance and control, as appropriate.

The close relationship between fusion and management – both in interactive operations and in their underlying design principles – will play an important part in effective system designs. It will also be important in defining effective engineering methods for achieving these designs. These topics are taken up in Sections 2 and 3, respectively.

## 2. Resource Management

Resource management may be defined as a process that combines multiple available actions (e.g., allocation of multiple available resources) over time to maximize some objective function. Such a process must contend with uncertainty in the current situational state and in the predictive consequences of any candidate action. A resource management process will:

- develop candidate response plans to respond to estimated world states;
- estimate the effects of candidate actions on mission objectives;
- identify conflicts for resource usage or side-effects of candidate actions; and
- resolve conflicts to assemble composite resource assignment plans, based on the estimated net impact on mission attainment.

The formal duality between data fusion and resource management – evident in the above definition and first propounded by Bowman[7] – allows the designer to adapt techniques and design principles from one domain to the other. In particular, Figure 3 shows how resource management processes are amenable to a canonical nodal representation that corresponds precisely to the representation for data fusion nodes given in Figure 1.

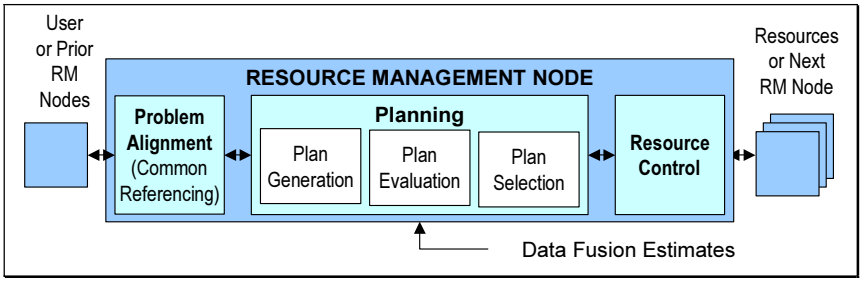



**Figure 3.** Resource management node paradigm.

According to this representational model, a resource management node performs some or all of the following functions:

*Problem Alignment (Common Referencing)*: normalizing performance metrics of the given (sub)problem and normalizing performance models of available resources, as well as any format conversion or spatio/temporal alignment of controls.

*Planning*:

- plan generation: Candidate partitioning of the (sub)problem into (sub)subproblems and candidate assignment of resources;
- plan evaluation: Evaluating the conditional net cost (e.g., probability of outcome × value of outcome − expected cost of plan execution);
- plan selection: Determining a decision strategy.

*Plan Execution/Resource Control*: generating the control commands to implement the selected resource allocation plan.

Planning is a process analogous to data association in fusion, involving functions corresponding to association, hypothesis generation, evaluation and selection. As with hypothesis generation in data fusion, plan generation involves potentially massive searches, which must be constrained in practical systems. The objective is to reduce the number of feasible plans for which a detailed evaluation is required.

As multi-nodal data fusion trees are useful in partitioning the data association and state estimation problems, so are resource management trees useful in partitioning planning and control problems. A data fusion tree performs an association/estimation process; a resource management tree performs a planning/execution process. Both these trees − one synthetic (i.e. compositional), the other analytic (i.e. decompositional) − are characteristically recursive and hierarchical.

Deeper planning can be accomplished by recursively partitioning a goal into candidate sets of sub-goals (plan generation) and combining them into a composite higher-level plan (plan selection). At each level, candidate plans are evaluated as to their effectiveness in achieving assigned (sub)goals, the global value of each respective goal and the cost of implementing each candidate plan (plan evaluation). By evaluating these cost/payoff factors to global mission utility, the need for deeper planning or for selection of alternate candidate plans is determined. In many applications, plan selection becomes an allocation search function in n-dimensions (i.e.; over n–1 future time intervals).

In many ways, the least understood part of problem decomposition is that of plan generation. This is ultimately the problem of systematically finding feasible approaches to given problems. The difficulty derives from the high-dimensionality of the feasible solution space in challenging real-world planning situations. Plan generation ultimately involves novel ways of partitioning problems and applying logical and physical principles in novel ways. Current automated planning systems generate hypotheses via a template method to constrain the plan evaluation/selection space.

In comparison with the difficulties of plan generation, plan evaluation and plan selection offer only the challenges similar to those encountered in designing the corresponding data fusion functions. Challenges to plan evaluation are in the derivation of efficient scoring schemes that reflect the expected utility of alternative response plans.

## 3. Recasting the Data Fusion System Engineering Problem

The above model for resource management – with its formal and operational relationships to data fusion – permits a powerful general method for system engineering; to include the engineering of data fusion systems. It is only necessary to realize that system engineering is itself a resource management process.

### *3.1. System Engineering as a Resource Management Process*

Resource management involves determining and executing a mapping from a problem space to a solution space. System engineering is such a process, in which the problem is to build a system to meet a set of requirements under constraints of budget, schedule, risk and availability of resources and technology. Fundamental to the system engineering process – as in all resource management processes – is a method for representing the structure of a problem in a way that is amenable to a patterned solution.

System engineering is generally not a discovery process, whereby an idealized essence of the problem is revealed. Rather, system engineering is a usually a process of imposing a candidate structure on a given problem, then evaluating that structure for net utility.

As a resource management process, system engineering can be implemented as a hierarchical, recursive planning and execution process.

### *3.2. Data Fusion System Engineering*

Issues specific to data fusion system engineering include problems of:

- selecting types of data for exploitation (e.g. feature sets);
- discovering exploitable context;
- modeling problem variability;
- discovering patterns that allow solution generalization;
- combining uncertain or poorly-modeled information;
- predicting technique and system performance;
- predicting development cost and schedule.

The Data Fusion Engineering Guidelines recommend a process for developing data fusion functionality within an information processing system. Design and development decisions flow from overall system requirements and constraints to a specification of the role for data fusion within the system. Further partitioning results in a specification of a data fusion tree structure and corresponding nodes. Pattern analysis of the requirements for each node allows selection of appropriate techniques, based on analysis and experience of applicability in the specified conditions.

The phases of the process may be summarized as follows:

- *operational architecture design*: System-level problem decomposition; assigning the role for data fusion, as well as for other system functions (sensors, communications, response resources, human operators, etc.);
- *system architecture design*: design of the data fusion network (usually a tree) by partitioning the process among C3 nodes and into processing nodes; specifying interaction with sensors/sources, resource management nodes, and information users;

– *component function design*: design of data fusion nodes, to include specifying data inputs/outputs of component functions (alignment, association and estimation), allocation to human/automatic processes, and technique selection;
– *detailed design and development*: pattern application, algorithm tailoring, software adaptation and development.

In each phase, analysis of requirements leads to a further functional partitioning. Performance analysis of the resulting point design can lead to further analysis, repartitioning and redesign, or to initiation of the next design phase. Thus, this process is amenable to implementation via waterfall, spiral or other development methods. Indeed, systems have been developed in which data fusion trees – the topology of the network for selecting batches of data and fusion techniques – are assembled adaptively to mission conditions.

In our Resource management framework, Data Fusion Engineering Guidelines can be thought of as the specification for an interlaced resource management/data fusion (RM/DF) process; the "phases" being levels in an interlaced pair of hierarchical trees, as depicted in Figure 4. A resource management tree assembles and implements designs, which are evaluated in each phase by a corresponding "fusion tree."

The functional architecture of the system engineering process represented in Figure 4 operates similarly to the interlaced tree architecture of an operational information exploitation system – the tactical weapon system represented in Figure 2. In both cases, goals and constraints flow downward from the system level to allocations over successively finer problem partitioning. At each level, a design phase constitutes a grouping of situations (e.g., principles for selecting batches of data to be fused) for which responses (design approaches) are coordinated.

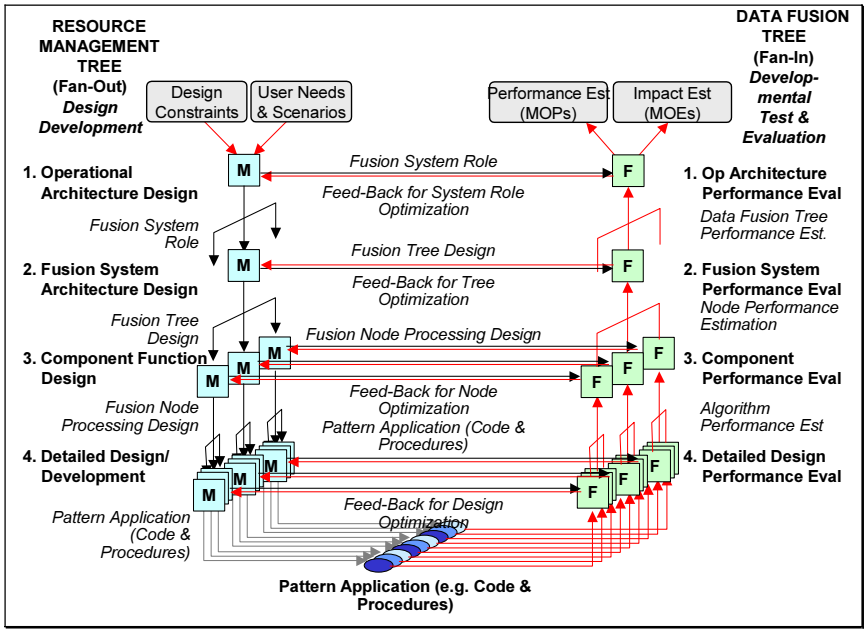



**Figure 4.** DF/RM tree for data fusion system engineering.

This procedure is hierarchical and recursive. Subordinate resource management nodes are activated to develop more detailed candidate design segments when a higher-level node determines that more detailed design is both feasible (in terms of estimated available schedule and resource cost) and beneficial (in terms of improved likelihood of attaining the assigned goal).

A tree-structured DF/RM process is used to build, validate and refine a system concept (which, in our application, may itself be a tree-structured RM or DF process). Each RM node in a system engineering process involves the functions characteristic of all RM nodes:

- problem alignment (Common Referencing): normalizing performance metrics of the given (sub)problem and normalizing performance models of available resources (e.g., DF tree types or DF techniques – using problem-space/solution-space matrices);
- planning: generating, evaluating and selecting design alternatives (partitioned according to the four design phases shown in Figure 4);
- plan execution/resource control: building or evaluating a system or system component (e.g. a DF tree, node or component technique).

The data fusion system engineering process builds, evaluates and selects candidate designs for the system and its components via a hierarchical, recursive process, which permits simultaneous reasoning at multiple levels of depth. The recursive planning process enables both an optimization of design against a given set of requirements and an ability to redesign as requirements change.

The engineering process distributes the problem solution into multiple design nodes (i.e.; RM nodes). The fusion nodes communicate to accumulate incremental evidence for or against each plausible solution. At any given stage, therefore, the engineering process will provide the best plan of action for achieving current goals, consistent with the current design assumptions, the available situational and procedural knowledge and the available planning and reaction times.

As in the integrated system shown in Figure 2, planning and control flow downward, while estimation and prediction (in this case, design evaluation) flow up. In response to the requirements/constraints flow-down, there is an upward flow, consisting of evaluated candidate design segments; i.e. proposed technique assignments and controls plus estimates of the performance of proposed design segments relative to their assigned goals.

A higher-level node evaluates cost versus benefit (in terms of its own higher-level goal) of the received design proposals and may re-partition its higher goal into a new set of sub-goals if the initial allocations are not achievable. For example, if a particular system requirement cannot be met, a new criterion for batching data must be developed.

The upward flow of evaluated plan segments is performed by a data fusion tree interleaved with the resource management tree. In the system engineering tree depicted, there can be an evaluation process corresponding to each design node.

### *3.3. Coordinated Multi-Level Data Fusion and Resource Management*

The above recasting of the system engineering process as a class of resource management process – coupled with the formal duality between resource management and data fusion – permits the systematic integration of a wide range of engineering and analysis

efforts. In effect, this insight allows data fusion techniques themselves to be employed in the systematic design and validation of data fusion systems. It also opens the way for a rigorous approach to the general discipline of system engineering.

Finally, the search for general methods that span system engineering and in-mission resource management can lead to a coordinated multi-level approach to real-world problems. The concept is depicted in Figure 5, as an elaboration of Boyd's well-known "OODA"-loop, in which a military force or other system responds to its environment in cycles of Observe-Orient-Decide-Act. [8].

As shown, perceived changes in the world state can elicit a diversity of responsive actions, ranging from immediate reaction with tactical resources to preparations for response over the long term, via system and technology improvements. As an example, the first appearance of German Me-262 jet fighters in World War II stimulated reactions in the individual allied aircraft and air defenses within seconds and minutes of encountering the novel threat. The appearance of the Me-262 also prompted changes over the course of days, weeks and months in allied counter-air tactics. It also stimulated a diversity of system and technology developments in jet aircraft and air defense systems that are still evolving sixty years later.

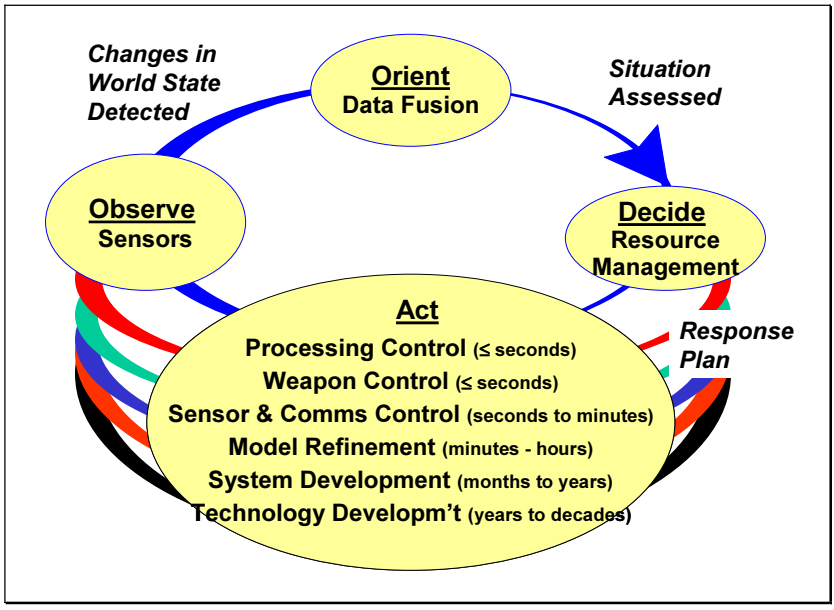



**Figure 5.** Resource management at multiple levels of granularity.

As with all resource management problems, tactical and strategic responses will be evaluated in terms of

- the assessed impact (i.e. the cost if no change in the current action plan is taken);
- timeliness and other physical constraints;
- resource capability available over time; and
- risk, due to uncertainty in the above factors.

An integrated resource management process will employ such factors in considering, evaluating and selecting one or more of the action types shown in the figure 5.

It should be noted that, in much the same way as resource management can build coordinated action plans operating at diverse levels of action granularity, so is it possible to perform data fusion that is coordinated at diverse levels of estimation granularity. For example, if a mission objective is to characterize and track a tank column, it may not be necessary to characterize and track each individual vehicle. By associating observations and estimating entities consistently at the appropriate level of granularity, it is possible to reduce combinatorial complexity and, possibly, to recognize features that only emerge at higher levels of aggregation. Analogous to the concept of Global Resource Management, Global Data Fusion will coordinate all the system's responses at all time scales.

## 4. Summary – Future Directions

We believe that this paper has opened several new avenues for potential research, including:

- refinement of the Data Fusion Engineering Guidelines;
- development of corresponding guidelines for resource management;
- developing methods for innovative plan generation;
- developing the formal theory of system engineering;
- adaptive modeling;
- adaptive system and technology acquisition; and
- coordinated data fusion and resource management at multiple levels of granularity.

## References

[1] White, Franklin E., Jr., "A model for data fusion," Proc. 1st National Symposium on Sensor Fusion, vol. 2, 1988.
[2] Steinberg, Alan N., Bowman, Christopher L. and White, Franklin E., Jr., "Revisions to the JDL Data Fusion Model," Proc. Third NATO/IRIS Conference, Quebec City, Canada, 1998.
[3] Engineering Guidelines, SWC Talon-Command Operations Support Technical Report 96-11/4, 1997.
[4] Steinberg, Alan N. and Bowman, Christopher L., "Development and application of data fusion engineering guidelines," Proc. Tenth National Symposium on Sensor Fusion, 1997.
[5] Bowman, Christopher L, and Steinberg, Alan N., "A systems engineering approach for implementing data fusion systems," Chapter 16 of Handbook of Multisensor Data Fusion, ed. David L. Hall and James Llinas, CRC Press, London, (2001).
[6] Steinberg, Alan N., "Data fusion system engineering," Proc. Third International Symposium for Information Fusion, Fusion2000, Paris, (2000).
[7] Bowman, Christopher L., "The data fusion tree paradigm and its dual" Proc. 7th National Symposium on Sensor Fusion, 1994.
[8] Boyd, J.R., "A discourse on winning and losing" (unpublished), August 1987; excerpted in http://www.d-n-i.net/fcs/ppt/boyds_ooda_loop.ppt.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# A Taxonomy of Sensor Processing Architectures

## *A Foundation for Data Fusion*

Shivakumar SASTRY [a] and S.S. IYENGAR [b]
[a] *Department of Electrical and Computer Engineering, The University of Akron, Akron OH 44325-3904, USA*
[b] *Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803 USA*

**Abstract.** Distributed Sensor Networks have evolved from the early networks of sensors coupled with processing elements to wireless networks of resource-constrained embedded devices. Such networks usher in new paradigms for computation, control and communication. Data Fusion is an important application for Distributed Sensor Networks as it facilitates the synthesis of new information by integrating data from multiple sensors in a deterministic, time-critical and reliable manner. In general, sensors are used either in complementary, competitive, or collaborative modes. The mode of the sensors forces a consideration of architectural issues. In this paper, we explore the landscape of architectures for Distributed Sensor Networks and identify the critical elements that are essential for Data Fusion.

**Keywords.** Distributed sensor networks, architecture, data fusion

### 1. Introduction

A Distributed Sensor Network (DSN) consists of a set of sensors that are interconnected by a communication network. The sensors are deeply embedded devices that are integrated with a physical environment and are capable of acquiring signals, processing the signals, communicating and performing simple computational tasks. The sesors are deployed in various environments and the *data* gathered by the sensors must be integrated to synthesize new *information*. Often, this synthesis must be performed reliably, within fixed time limits to support business objectives. In certan applications such as automation systems, these tasks must be performed periodically while satisfying demanding performance constraints.

The efficient synthesis of information from noisy and possibly faulty signals emanating from sensors requires the solution of problems relating to (a) the architecture and fault tolerance of the DSN, (b) the proper synchronization of sensor signals and (c) the integration of information to keep the communication and computation demands low. From a system perspective, once deployed, a DSN must organize itself, adapt to changes in the environment and nodes and continue to function reliably. Current technology trends and devices facilitate several architectures for DSNs. In this paper, we propose a taxonomy for DSN architectures. Such a taxonomy is useful for understanding the evolution of DSN and for planning future research.

## 2. Benefits and Limitations of DSN

A DSN is an evolution of a traditional approach that is used to acquire inputs from a collection of sensors to support user decision-making. The traditional approach to solving this problem is depicted in Figure 1 (a). Data from a collection of sensors are gathered by connecting the sensors to interface cards in a computing system. This data is presented to applications in suitable formats, and applications present information that is synthesized from such data to users. Figure 1 (b) shows the organization of a DSN. Sensors are networked via wired or wireless media. There is no central computer that performs the coordination tasks. The network itself is a computer and users interact with this network directly, possibly in interactive or proactive paradigms [1]. The data gathered by various sensors is integrated to synthesize new information using Data Fusion techniques [2,3].

DSNs may be distributed spatially and temporally to suit application demands. Often such networks lead to low-cost and reliable implementations. Quick response times are feasible for demanding sensing-loops. DSNs operate over large time-scales and may be developed incrementally. Sensors can detect multiple input modalities and combining such values provides new information that cannot be sensed directly. The overall throughput for sensing is increased due to concurrent operations. The reliability of sensing is improved by using redundant sensors. Redundant sensors also make it feasible to develop fault tolerant systems that degrade gracefully under exceptional circumstances. Groups of sensors work in complementary, competitive, or collaborative modes – and it is necessary to use different Data Fusion techniques to synthesize information in each of these cases.

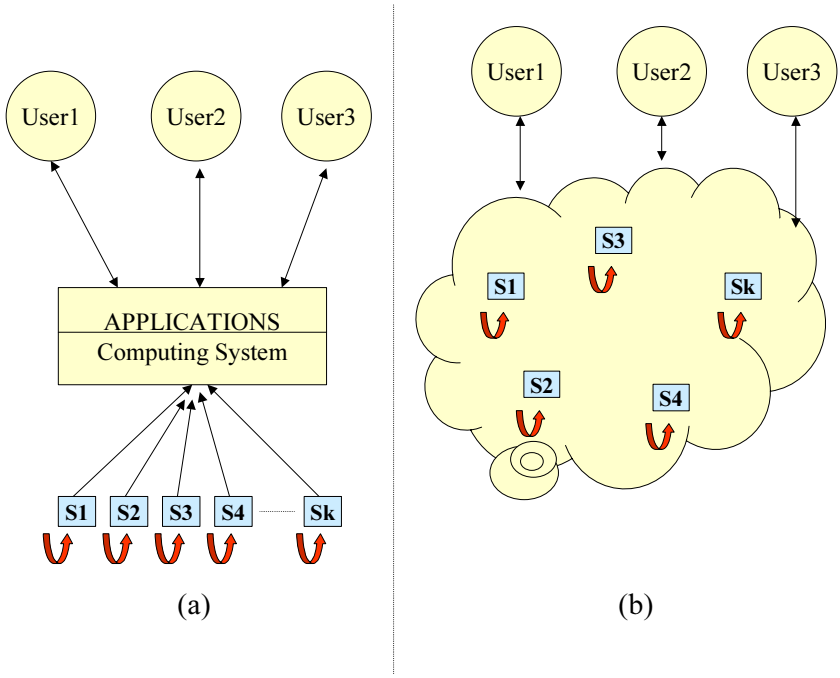



**Figure 1.** Fundamental architectural change in DSN.

If the individual nodes of a DSN require configuration or programming, such tasks are difficult because of the scale involved. Communication mechanisms have to match application demands to achieve effective coordination between sensors and hence these mechanisms tend to be application specific. Sensors, typically, do not have individual identifiers. Distributed security mechanisms haven't yet matured and time synchronization across nodes is a significant challenge. New operating systems, communication protocols and security mechanisms are required to work with DSNs.

## 3. General Technology Trends Affecting DSNs

Following [4] we examine technology trends that impact the architecture of a DSN. A DSN is built using a large number of resource constrained devices that are capable of acquiring inputs from the physical environment, processing and digitizing the data, communicating, and maintaining system cohesion. The primary impetus for distribution comes from an overruling of Grosch's Law in 1970.

### *3.1. Grosch's Law Overruled*

Grosch's law states *that the cost per machine instruction executed is inversely proportional to the square of the size of the machine*. This trend is now reversed [5–9]. Microprocessors that are embedded in sensors are relevant for DSNs. Network operating systems and tools for designing, configuring, and maintaining DSNs are coming into existence and promise to unleash the power of the aggregate.

The increased capabilities of individual sensor devices is consistent with Moore's law.

### *3.2. Moore's Law*

*Computational hardware performance doubles, for about the same cost, every two years.* This trend of increased computational power is largely due to the mass production of microprocessors. While at one time it may have been cost effective to optimize hardware resources, the current trend is to favor sacrificing hardware utilization if the software development effort can be streamlined. Moore's law also motivates hardware realizations of critical functionality.

### *3.3. Greater "Siliconization"*

Communications devices are manufactured with an increasingly larger proportion of Integrated Circuit (IC) technology (not necessarily limited to silicon, e.g., Gallium Arsenide). Thus, Moore's law is applicable to communications as well, although other factors (e.g., Shannon's law) may limit the rate of the curve compared to microprocessor development. Greater siliconization is caused by two effects: first, cost efficiencies result from mass production in chip foundries and second, achieving greater transistor densities means that sensing, communicating and computing capabilities can be integrated in a single chip, thereby further decreasing the costs of interconnection, especially by using wireless media.

Much of the functionality of a sensor may be more cost-effectively produced using Complementary Metal-Oxide Semiconductor (CMOS) and related technology. Power devices are also seeing potential realization in silicon by means of Micro-Electro-Mechanical Systems (MEMS) technology. Devices at the nano-scale are appearing on the horizon. *Silicon sensors*, i.e., analog and digital sensor devices that can be manufactured using chip-processing techniques, are becoming increasingly successful. We are witnessing a trend to host communication stacks and security mechanisms in silicon.

### *3.4. Increasing "Digitization"*

Closely related to the above trend (of greater siliconization) is the trend to favor digital over analog realizations by collocating analog to/from digital conversion functions with sensors. For example, in smart sensors, a microprocessor or a digital signal processor can perform signal-processing functions that were previously performed either with additional analog circuits or in interface cards. Such collocation improves the reliability of data transmission and decreases costs (e.g., wiring, installation, maintenance).

### *3.5. Increasing Use of "Generic" Components*

Start-up costs are taking an increasingly greater proportion of the overall cost (including production). The chip manufacturing case is a good example of this general trend, since the majority of the cost is in manufacturing the first chip. Economy of scale then results from the fact that if more chips are produced, then the cost per chip is reduced.

This trend encourages the use of "generic" products (which may be over-qualified in some respects, but are still more cost-effective) due to the economies of scale arising from mass production. Other benefits, such as cost of deployment, maintenance and training, also contribute towards reducing long-term operational costs.

### *3.6. Open Systems*

In the current computer marketplace, there is a strong trend towards open systems. There are both economic and philosophical advantages to producing open systems but there is also greater competition in any particular arena. In the context of a DSN, open systems appear to be unavoidable as generic intelligent sensors begin to flood the market. For easier integration with the physical world, DSNs must be able to operate with device level open standards such as Serial Realtime Communication System (SERCOS) and Ethernet; simultaneously, to integrate with supervisory systems, DSNs must also support emerging industry-specific open interfaces.

## 4. Taxonomy of DSN Architectures

To realize systems that are depicted by Figure 1 (b), we need to address five major aspects that are shown in Figure 2. For each of the major aspects, there are variations that fundamentally alter the structure and performance of the DSN. These variations are captured in the taxonomy proposed in this chapter.

Following our prior work [4], we distinguish between *function* and *implementation*. Function refers to the basic operations or capability that the DSN aspect must address. Implementation refers to the methods that are used to accomplish the functions

and the location of these methods. The location of the implementation is important because it is closely related to how the sensors are packaged. Packaging is a critical consideration because of cost. For each sensor, there is a certain minimum cost incurred in installing the sensor and maintaining it throughout the system lifecycle. By collocating groups of sensors, some of these costs can be amortized across sensors. While distributing sensors is desirable from an architecture perspective, collocating sensors is desirable from a cost perspective and hence finding an appropriate balance between these considerations is the principal focus for packaging design.

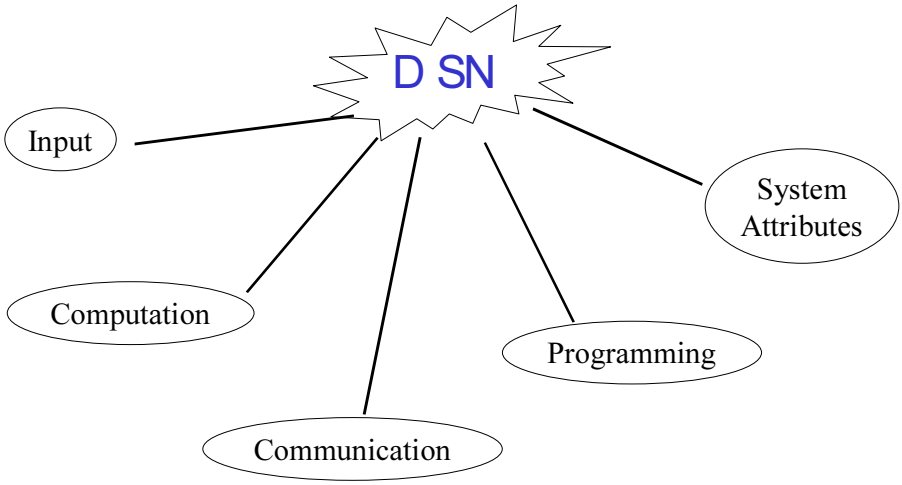



**Figure 2.** Major aspects of a DSN.

The primary purpose of a DSN is to gather data from a physical environment within a predictable, bounded response time. A *hard realtime* DSN is one in which the inability to respond to input events within their associated deadlines results in a system failure. In a *soft realtime* DSN the inability to meet the deadlines does not result in a system failure but a cost penalty is associated with tardiness. For example, if the data gathered is used to regulate the operation of a critical actuator (such as coolant in a power plant), we need a hard-realtime DSN; in contrast, if the data gathered is used to locate a nearby restaurant in a future automobile SmartSpace, we need a soft-realtime DSN. A non-realtime system will provide the outputs as fast as possible, with no guarantees and no innate design that supports predictability. Features of an implementation that supports this predictability property will be emphasized in the taxonomy.

Depending on the way in which the DSN operates, it is said to be deterministic, quasi-deterministic, or non-deterministic. In a *deterministic* DSN, it is possible to accurately predict the performance of the DSN. In a *quasi-deterministic* DSN, although performance cannot be predicted as accurately as in a deterministic DSN, it is possible to determine worst case upper bounds within which the DSN can be guaranteed to perform. In a *non-deterministic* DSN, it is not always possible to guarantee performance and it may be difficult to predict response time without the detailed modeling of various parameters. Under "normal" operating conditions, the performance of non-deterministic systems can be significantly better than that of other systems; however, behavior under "abnormal" operating conditions is difficult to characterize. The predictability of the various subsystems will be discussed; for example, the communica-

tion subsystem or the operating system in sensors can either be deterministic, non-deterministic, or quasi-deterministic.

To construct a DSN, one must select at least one of the options in Input, Computation, Communication and Programming aspects. Zero or more choices may be made among the system attributes depending on a cost/performance balance. In the following taxonomy diagrams, solid lines under a heading indicate choices that are mandatory for a successful system realization; dotted lines indicate optional choices (e.g. see Function in Figure 3). A set of dotted lines connected by a solid arc (e.g. see Transduction in Figure 3) represents the situation when at least one choice must be made.

### *4.1. Input*

We now consider the Input aspect of a DSN as depicted in Figure 2. The function of the Input subsystem in a DSN is to capture the input signals from the physical environment and convert the signals to values suitable for processing. As shown in Figure 3, there are four primary functions in the Input subsystem: *Transduction* and *Signal conditioning* are mandatory functions while *Diagnostics* and *Processing* are optional functions.

*Transduction* is either of type analog or discrete. Discrete Input is typically one bit of information (i.e., on/off) while analog values may require a substantially larger number of bits and represent continuous values within bounds of representational errors. Data Fusion strategies are significantly affected by the type of transduction. For example, several results exist when the sensor values are continuous [10,11]. The theory for discrete type needs further work. Applications tend to be predominantly one or the other, although mixed systems are becoming more prevalent.

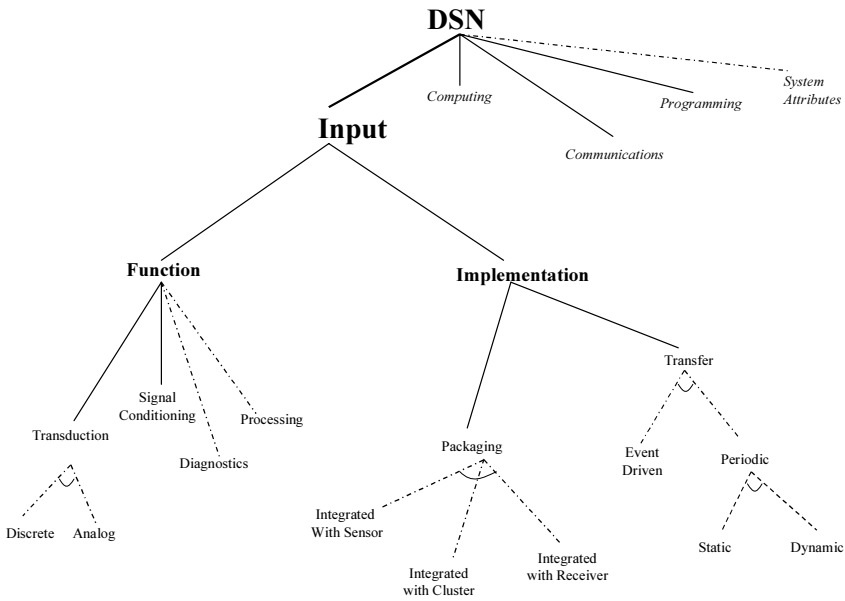



**Figure 3.** DSN input aspect taxonomy.

The *Signal Conditioning* function includes activities such as amplifying, digitizing, filtering, forcing, or other signal processing computations. During input, the result of signal conditioning (irrespective of transduction type) is a digital representation of sensed values. Digital representations are desirable because of their robustness with respect to noise, ability to support error detection and correction schemes, ease of storage and manipulation, and ability to superimpose security mechanisms.

*Diagnostics* for a sensor refers to methods for determining whether the sensor is functioning properly or has failed. Often, additional circuitry is required for performing self-tests at the level of an individual sensor. In certain applications, it is feasible to use the values of other sensors, or small history tables, to deterministically ascertain whether or not a sensor has failed.

*Processing* refers to a list of tasks that may be included at the level of a sensor. Examples of such tasks include historical trending, data logging, alarming functions and support for configuration and security management.

The *Implementation* of these Input functions significantly affects the overall architecture and performance of the DSN. The transduction function can only be collocated with the sensor since transduction is the primary purpose of a sensor. The remaining Input functions (i.e., signal conditioning, diagnostics and processing) may be:

- located by integrating these functions with the sensor; or
- located in special modules and provide these services to a cluster of sensors; or
- located in devices that use data from the sensors.

These implementation options are shown in Figure 3 as *Packaging* options. A particular choice affects the response time (under normal and fault conditions), wiring costs and the processing requirements of the DSN. By performing the functions locally, we can optimize implementation with respect to specific devices and avoid contention for communication and DSN resources thus resulting in faster sampling rates. By locating these functions in a data concentrator, we can reduce the cost of design and maintenance, use more resource constrained sensor devices and apply the functions at the level of a cluster. Such a choice tends to increase security risks because of the cost of securing the links between individual links and clusters. The third alternative is to transmit the raw data gathered by sensors directly to the receivers in the DSN that require the data. Typically, this choice tends to increase communication demands and precludes options for early recognition of critical status information.

As an example, consider the diagnostics function. This function can be implemented at the level of a sensor, a cluster, or at the receiver that uses data from the sensor. By locating the diagnostics function at the sensor, we can make local decisions within required time limits. The propagation of erroneous data is prevented. However, we need additional (redundant) circuitry and system resources (memory, timers, etc.) at the sensor level. By performing the diagnostic function at the cluster-level, we reduce design, implementation and maintenance costs. It is feasible to use values of other sensors in the cluster to diagnose a sensor. The resources of sensors can be constrained while the resources at some of the devices (data concentrators) are less constrained and better utilized. On the other hand, if we choose to locate diagnostics at the receivers that use sensed data, we may require redundant implementations of the function, increase the resource requirements for receivers and increase the risk of propagating erroneous data in the DSN. The specific choice depends on the application and must be made to balance system-wide cost/performance issues.

The Input *Transfer* function refers to how sensed data is delivered to the DSN and is primarily the responsibility of the communications subsystem. However, from the Input aspect's perspective, the implementation of a transfer method involves the specification of the synchronization method (either periodic or event driven). This choice affects the manner in which the operating system and communication protocols at the level of sensors are designed.

*Periodic Input* synchronization can either be static or dynamic. Depending on the packaging, such synchronization can be initiated either by the DSN by using a master clock, by sensors using local timers, or by data concentrators. Periodic transfer is said to be *Static* if the data is gathered deterministically within a fixed time period, called the scan time. The time period for each sensor in the DSN may be different. Static-periodic systems have significant, unintentional variations in the scan time. For example, if the strategy is to scan *as-fast-as-possible*, scan time is affected if the time to process certain pieces of data is different from others. The scan time also varies when the DSN initiates special processing and transfers in response to certain abnormal events such as security breaches or multiple sensor faults. Periodic transfer is said to be *dynamic* if successive scan times are not equal. When using dynamic transfer mechanisms, it is important to track both the value and the time at which the data is obtained before synthesizing information.

*Event Driven* Input synchronization is fundamentally different from periodic synchronization and is based on either detecting:

- a Change-Of-State (COS) of predefined variables; or
- predefined events (as sequences or expressions of COS of predefined variables).

The advantages of an event-driven system over a periodic system are: 1) that it is on average more responsive, in the same sense that an Ethernet has a smaller average delay compared to an equivalent TDMA scheme and 2) the amount of communication in a COS-based system can be reduced by not sending repetitive information. However, the disadvantages are: 1) additional measures are necessary to guarantee the delivery of data, 2) methods to detect failures in a sensor-receiver path are required (since it is difficult to distinguish between a failure and a long period of no COS) and 3) mechanisms are necessary to prevent an avalanche of data from overwhelming the communications system under exceptional situations. Unlike periodic Input synchronization, event driven Input synchronization is non-deterministic unless techniques for bounding the performance are used (e.g., priority scheduling).

### *4.2. Computing*

The availability of effective communications, coupled with the computational capability of sensors, makes it feasible to host various tasks in sensors. As shown in Figure 4, the four primary functions are *Algorithm Processing*, *Process Diagnostics*, *Data Management* and *System Interfaces*.

*Algorithm Processing* concerns tasks that are performed in a sensor node. Specialized algorithms are required to condition signals, encrypt data and process data in the node. Depending on the overall design of the DSN, the nodes may implement components of a distributed algorithm. The operating environment of a node is responsible for ensuring that these algorithms are executed fairly and effectively.

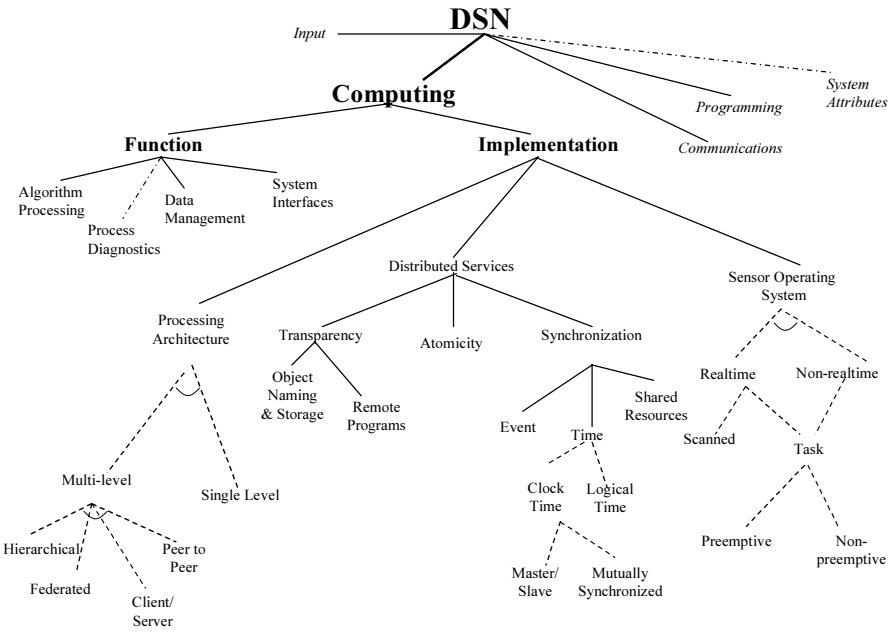



**Figure 4.** DSN computing aspect taxonomy.

*Process Diagnostics* are additional computations that are performed at the sensor or cluster levels to augment the Processing function of the Input subsystem. Various techniques for automatically embedding code, for diagnostics, monitoring, or distributed services, in the algorithms are being investigated. For example, such embedded code could provide status information and alarm data to operator monitoring stations. Some diagnostic strategies require temporal information in addition to the Input data.

*Data management* is another function that is becoming increasingly important for DSNs. Because of the size of contemporary systems, the data that is gathered by the collection of sensors is immense. Typically, it is not feasible to associate mass storage devices at the level of a sensor and the amount of memory available in a resource-constrained sensor is limited. Thus it is necessary to manage the data in a DSN and effectively synthesize information that is useful for decision-making. Data management considerations for periodic systems are more critical because of issues of data freshness.

The Computing subsystem must support multiple *System Interfaces* to effectively integrate with other systems. For the interface with the physical environment, it is necessary to interface to proprietary and open sensor interface standards. For example, there are several sensors that interface with Ethernet or SERCOS. To allow users to work with emerging pervasive devices or to incorporate the DSN as an infrastructure for a SmartSpace for Automation [12], the DSN must support open interfaces that are based on XML or such other technologies.

The implementation of these functions is discussed under the categories of *Processing Architecture*, *Distributed Services* and *Sensor Operating System*.

A DSN is a collection of sensors that are interconnected via some communications media. There are two choices for the *Processing Architecture* of a DSN. In a single-level architecture, all the sensors in the DSN are considered uniformly. Typically, in such an organization, we need to capture and reason about contextual information to properly manage system evolution. Because of the immense scale of DSNs, multi-level architectures are more likely to be successful. There are four common variations of multi-level architectures: 1) *Hierarchical* in which there are tiers of authority in which sensors in higher tiers are masters of sensors (slaves) in lower tiers of the system; 2) *Federated* in which certain responsibilities are granted to sensors in a higher tier, but many functions are performed autonomously by sensors in lower tiers; 3) *Client-Server* in which sensors are delineated into roles so that clients request services or data from the servers; and 4) *Peer-to-peer* in which sensors can be either clients or servers or both. These architectures are not always clearly separable. We expect most systems in the future to be basically federated, with many subsections organized as peer-to-peer or client/server.

*Distributed Services* facilitate the coding and operation of a DSN and are provided by a distributed operating system that is represented by the collection of operating systems on each sensor. *Transparency* refers to the ability to regard the distributed system as a single computer. Tannenbaum [13] defines several forms of transparency for distributed systems: 1) data or program location, 2) data or process replication, 3) process migration, 4) concurrency and 5) parallelism. For our purposes, in a DSN, transparency concerns the *Object Naming and Storage* service, which provides the ability to access system objects without regard to their physical location, and *Remote Program Services*, which provide the ability to create, place, execute or delete a program without regard to the sensor. Typically, servers are necessary to perform the registration and lookup functions to provide these services.

The *Atomicity* service is used to increase the reliability of the system by insuring that certain operations (called transactions) occur in their entirety, or not at all. Various forms of recovery mechanisms can be implemented to checkpoint and restore the component state should the atomic operation fail. Typically atomicity is more important at the level of information-based transactions and less important at the level of periodic data gathering.

The order in which data from various sensors is gathered and the nature of interactions among the multiple sensors depends on the *Synchronization* method. The *Event* service allows a sensor to register an interest in particular events and to be notified when they occur. The *Time* service is used to provide a system-wide notion of time. An important application of system time is in the diagnostic function where it is used to establish event causality.

Two forms of time are possible: *Clock Time* and *Logical Time*. Providing a system-wide *Clock Time* that is globally known within a specified accuracy to all the controllers in a distributed system can be difficult. Clock time can represent a standard Universal Coordinated Time (UTC) or it can be a common time local to the system. Two common techniques are: 1) provide a hierarchical *master/slave* system in which a master sensor in one device is then transmitted to the other slave sensors or 2) use a peer-to-peer distributed mechanism to exchange local times among various sensors. For certain applications it is possible to use GPS (Global Positioning System) devices as master clocks to synchronize multiple controllers with UTC.

*Logical time* only provides the relative order of events in the system, not their absolute clock time. For many applications, exact time may not be as important as ensur-

ing that actions occur in the correct sequence, or within certain relative time intervals between events. Many algorithms can be rewritten to use logical time instead of clock time to perform their function. Providing logical clocks in a distributed system may be more cost effective if the applications can be restructured.

The management of *shared resources* across the network is supported through mechanisms that implement mutual exclusion schemes for concurrent access to resources.

All tasks in a sensor execute in an environment provided by the *Sensor Operating System*. This operating system provides services to manage resources, handle interrupts and schedule tasks for execution. The operating system is said to provide *Realtime* services if the length of time required to perform tasks is bounded and predictable. The operating system is said to be *Non-Realtime* if such services are not supported. Realtime services are supported either by providing a periodic execution model or by providing a realtime scheduler (e.g., rate monotonic scheduling). These schedulers are priority based and can be preemptive (interruptible) or not. Preemptive scheduling can provide the fastest response times, but there is an additional context swap overhead.

Depending on the way in which the scheduler operates, the methods used to program computing tasks and the interaction with the communication interfaces, the execution in a sensor can be deterministic, quasi-deterministic, or non-deterministic. One of the main challenges in DSN research is to design efficient deterministic and quasi-deterministic sensor nodes.

### *4.3. Communications*

The communication subsystem is the primary infrastructure on which the DSN is constructed and hence design choices made in this subsystem strongly affect the other capabilities of the DSN. Figure 5 presents a taxonomy of this subsystem. The primary functions in this aspect are: *Data Transport* and *Bridging*.

We distinguish between three types of data and each of these types has different characteristics. *Input Data* gathered by sensors is typically limited to a few bytes and needs guaranteed, deterministic message delivery to maintain integrity. Sensors communicate primarily to synchronize and to recover from failures. Thus, *Inter-Sensor* traffic is likely to be sporadic, contain more information (aggregated data) and be more suitable to quasi-deterministic or non-deterministic delivery mechanisms. *System data* refers to all the other data delivery needs that may or may not have hard realtime requirements. For example, data required for system monitoring and status alarms may be critical and realtime, while that used by Internet based supervisory systems may not. Non-realtime system data, such as downloads, can be typically handled in a background mode using a “best effort” protocol.

The *Bridging* function, which transports data between multiple networks, is an important component of contemporary distributed systems such as DSNs that are likely to be integrated into existing engineering systems. Bridging refers to tasks performed on interface devices that connect two (or more) networks. The protocol used on the networks may or may not be the same. These intelligent devices provide services such as data filtering, Data Fusion, alternate routing and broadcasting and serve to partition the system into logical subsets.

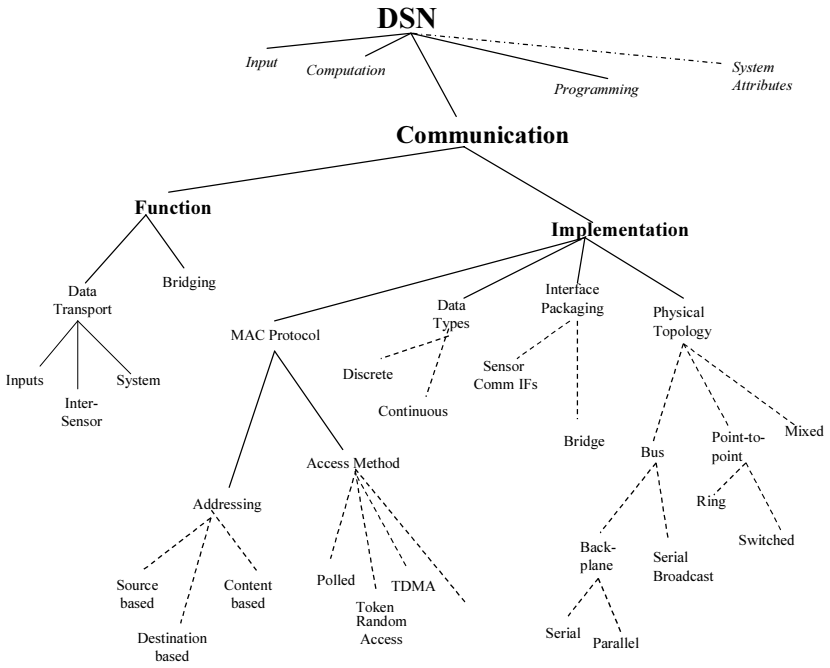



**Figure 5.** DSN communications aspect taxonomy.

A communication protocol definition such as in the Open Systems Interface (OSI) is designed as layers of services from low-level physical implementation, to media access, through internetworking, up to the application layer. Such layered communication protocols are unlikely to be implemented in resource constrained sensor nodes. For this taxonomy, we focus only on the Media Access Communication (MAC) layer since it appears to be the layer where most variations occur. Under the *MAC Protocol* implementation attributes, we consider two attributes: the addressing scheme and the access mechanism. The method of addressing messages, called the *Addressing Scheme*, can be *source-based*, in which only the producing device's address is used in messages, versus using the destination address to route the message. Source-based schemes can be extended to use *content-based* addressing in which codes are used to identify the type of data within the message. Source or content-based schemes are typically used on a broadcast bus, a ring, a data server, or when routing schemes can be a priori specified. *Destination-based* schemes are used when there is usually one destination or when the routing is dynamically constructed.

The capability to provide deterministic service is strongly affected by the *Access Method* that establishes the rules for sharing the common communication medium. Polled, Token-based and Time Division Multiple Access (TDMA) schemes that use a fixed time slot allocation are deterministic. Token-based schemes that allow nodes to skip their time slot when they have nothing to transmit, have quasi-deterministic behavior. Random access schemes such as Ethernet result in non-deterministic performance while a priority-bus scheme (e.g., CAN) can be made quasi-deterministic.

The *Data Types* supported on the network is an important design consideration and is related to the type of transduction in the Input aspect of a DSN. If the communication system is optimized for binary or discrete data then other types of data (e.g., analog)

must be transmitted less efficiently. On the other hand, using general protocols precludes the possibility of optimizing special data sets. The choice will be guided by the demands of the application. There may be segregated networks in which the data on each network can be strictly Input data, strictly inter-sensor messages, or strictly system data (e.g., Ethernet); each through a *sensor communications interface* implementing possibly different protocols. Alternatively, the traffic may be mixed on a single network through separate interface devices sharing the media or through a common integrated interface. A bridging function can be packaged as a separate device or integrated with special sensors.

Another influence on the overall architecture is the *Physical Topology* of the communication system. The communication media (wired or wireless) largely determines the topology in which sensors are connected. Wired systems are either *bus based* (single or multiple), a *point-to-point* system or a combination of the two (e.g., switched busses). The bus-based systems can refer to a local *backplane bus* or to a *serial bus* as in a local area network (LAN). Typically, bus-based systems are physically able to support message broadcast schemes. Local backplane busses are usually very high speed, relatively short length and able to perform memory operations at processor speeds. They can be serial (1 data line) or parallel (many data lines). Serial busses that are used in LANs are typically slower but capable of extending from a few meters to a few kilometers and permitting larger numbers of nodes to be connected. The actual range depends on the protocol, e.g., a token-bus is unlimited in length (except that performance degrades) while a CAN bus has a basic limit due to end-to-end propagation time. The point-to-point systems are usually organized as: 1) a ring, in which data is circulated to all devices on the net; or, 2) switched, to allow messages to be delivered to their specified destinations. Switched topologies can vary from tree-structures to grids to irregular patterns. Several interconnection topologies such as hierarchical, committee, binary trees and deBruijn networks have been considered in the past. With the recent trends in wireless networks, these interconnection topologies are important more for maintaining system cohesion in the presence of changing conditions and less for interconnection.

### *4.4. Programming*

This aspect has been largely ignored in the DSN literature and must cover a range of activities including *designing, developing, debugging and maintaining programs* that perform computing, input and communication tasks at the sensor levels. Programs must also be developed to support distributed services that are essential for proper functioning of the DSN. In addition, activities such as *abnormal state recovery*, *alarming* and *diagnostics* must be supported.

Figure 11 shows the primary functions of the programming category: support for coding of the *algorithm*, *system testing*, *diagnostics*, *exception handling*, *data management*, *documentation* and *synchronization*. A key component of each function is the differences that are imposed by having to run in a distributed environment and what services are provided by the programming languages and operating system. For example, the algorithm at a given sensor may require data from another sensor. An issue is whether the data is easily available (transparent services) or the programmer must provide code for accessing the remote data explicitly. System testing, diagnostics and exception handling are complicated by the fact that data is distributed and determination of the true system state is difficult. Documentation includes the program source code

and details of system operation. Questions of where do programs and documents reside in the distributed system arise, as well as issues in version control and concurrent access. Lastly, the degree of transparency in synchronization that is provided by the languages and environment is a key to simplifying distributed programming.

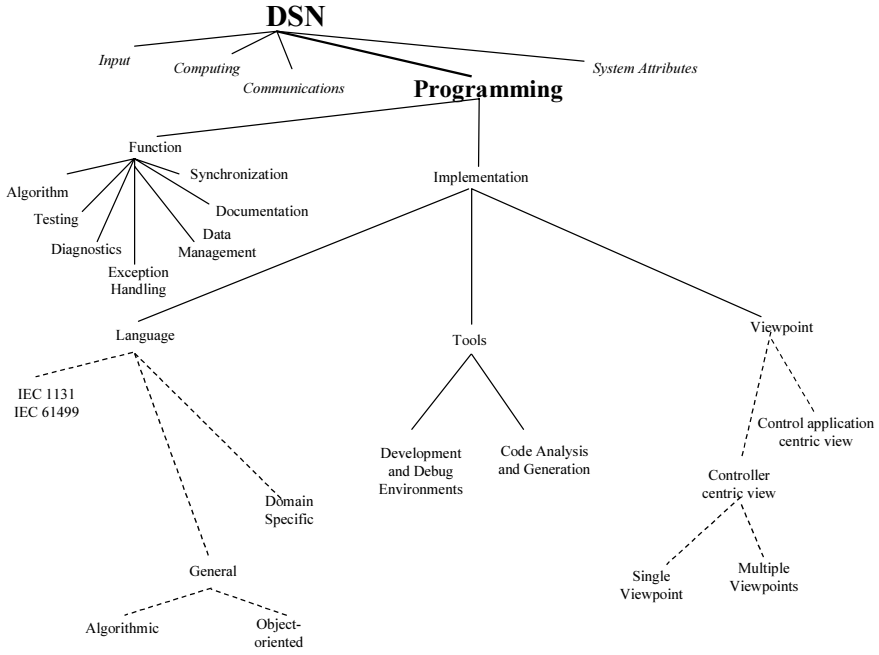



**Figure 6.** DSN programming aspect taxonomy.

The *Language* chosen in a DSN to implement the algorithm affects the services and tools that must provide support (e.g., operating system, compilers, partitioning, performance estimation). The IEC 1131 Programming Standards for digital controllers and the more recent IEC 61499 extension that defines an event-driven execution model are interesting considerations for programming DSNs. Ladder Logic is relatively simple to learn, easy to use and provides a low level ability to react to process changes. Sequential function charts, Petri Nets and Finite-state machines (FSMs) are examples of *state-based* languages. FSM models are intuitively simple but the size of these models grows rapidly as the size of the control system increases. Hierarchical representation methods, such as Hierarchical FSMs, have been used to cope with the large size of state based models. While such hierarchical methods were well suited for hardware design, their use in software design is still an ongoing research issue. Function blocks are designed as a replacement for Ladder Logic programming in an industrial environment. They provide a graphical, software-IC style language that is simple to use and understand. Function blocks are modular software units that can contain internal state and represent the inputs and outputs of the function. Libraries of common blocks are provided for building control programs.

*General Purpose* languages such as FORTRAN or C are employed to specify the processing in sensors. More recently, *object oriented* languages are being used to program controllers. *Domain specific* languages, with extensions to specify Data Fusion functions that are tailored to the needs of particular applications are likely to be useful.

*Development and debugging environments* for a DSN should support modular, independent programming of different sensors. Key distributed programming constructs can be provided to the programmer by distributed system services or they can be embedded in the language and implemented by its compiler/linker. For example, a name server can provide the location transparency or it can be a remote procedure call generated by the compiler. In addition, the programmer must be able to debug and maintain the system by viewing and manipulating the code in many sensors simultaneously. Formal models and theory help in simplifying this complex task.

Because of the immense scale of DSNs, techniques that support the *automatic generation and analysis* of software are important. In an automated code generation system, the responsibility for managing and maintaining all the interactions between sensors (either by message passing, shared memory, or sharing IO status) is handled automatically. Formal models and theory such as Petri Nets or compiler transformation theory make the task of software synthesis (and integration) simpler by exploiting the underlying mathematical structure. The user is only responsible for providing a high level specification of the application needs. In addition, the formal models and theory are also useful for introducing new functionality such as abnormal state recovery, alarming and diagnostics.

The *Viewpoint* is another important issue in the programming aspect. Most of the current programming environments support a *sensor centric view*. In this view, the needs of the control application must be expressed in terms of the capabilities of the sensor that is used in the DSN. When dealing with large applications, managing such programs is a cumbersome activity. In an *application centric view*, users express Data Fusion and integration needs by describing relationships among objects in the domain of the control application. Application centric views can be supported with any level of abstraction (i.e., low, medium, or high). However, control application centric views with a low level abstraction tend to be more akin to a CAD drawing for a printed circuit board than a traditional software program. A high level of abstraction is preferable.

A sensor centric view makes the language more general (i.e., the language and programming environment can be used for different control applications). On the other hand, programming a general-purpose control application centric view can be a complex and difficult task.

### *4.5. System Attributes*

Several distributed systems support the aspects that are discussed in the preceding sections. DSNs are distinguished by the attributes that are discussed in this section. Teams of researchers are actively investigating all these areas and we expect that the systems landscape will be significantly changed over the next two years.

#### *4.5.1. System Integration*

DSNs are unlikely to operate as stand-alone applications. Such systems are likely to be installed as a complementary system in existing engineering infrastructures. Therefore, it is critical for the DSN to integrate with such systems. In particular, the Bridging function of the communications aspect must provide support for open information and data exchange standards and protocols. Such support is necessary both for active monitoring of the DSN and for passive information gathering from the DSN.

#### *4.5.2. Operating Systems*

Architecture issues discussed in [6] are relevant for DSNs. TinyOS is an interesting node-level operating system [6,14]. Some of the physical world integration issues discussed in [1,15]. are relevant to DSNs. The reflexive relationship between the computing devices and the application is emphasized and exploited in DSNs by embedding the goal-seeking paradigm (see Emerging trends section) in the infrastructure. Mechanisms for self-organization in sensor networks [16]. are important for a DSN. However, because of the tight integration with the physical world, performance demands and the heterogeneous nature of DSN nodes, we require a new, localized approach to self-organization that preserves determinism, safety and predictability. The issue of synchronizing time in a sensor network [17]. is critical for DSNs and deserves considerable investigation.

#### *4.5.3. Communication Protocols*

The large number of nodes in a DSN motivates research into new methods for naming nodes [18]. and discovering services [19]. A low-overhead communications mechanism is necessary and Active Messages [20] is unlikely to provide the low jitter required in certain DSN applications. Ideas of gradients and interests in Directed Diffusion [21] are likely to be useful for disseminating low-priority information in a DSN during normal operations. SPEED, the new soft-realtime protocol for sensor networks [22] is likely to be adequate for slow, non-demanding, applications.

#### *4.5.4. Security*

Distributed security mechanisms are a topic of active research. DSNs add to the security issues that are inherited from general distributed systems [23,24], wireless networks [25,26], sensor networks [27–30] and Ethernet based factory systems [31] The realtime nature of the DSNs and the rugged operational characteristics that are desired offer new challenges in service discovery, device management and Data Fusion [2,3,10].

## 5. Conclusions

The landscape of architectures of Distributed Sensor Networks is vast. The major aspects of a DSN are Input, Computing, Communication, Programming, and System Attributes. The taxonomy proposed here provides a systematic mechanism to traverse this vast landscape. The taxonomy is a useful tool for research planning and system development.

## Acknowledgements

This work is supported in part by the University of Akron, College of Engineering Research Startup Grant, 2002–2004 to Dr. Sastry and an NSF Grant # IIS-0239914 to Professor Iyengar.

## References


[1] D. Tennenhouse, "Proactive Computing," Communications of the ACM, Vol. 43, No. 5, pp. 43–50, May 2000.
[2] R.R. Brooks and S.S. Iyengar, Multi-Sensor Fusion: Fundamentals and Applications with Software, Prentice Hall, NJ, 1997.
[3] S.S. Iyengar, S. Sastry and N. Balakrishnan, "Foundations of Data Fusion for Automation," (Invited Paper) IEEE Instrumentation and Measurement Magazine, Vol. 6, No. 4, pp. 35–41, December 2003.
[4] J. Agre, L. Clare and S. Sastry, "A Taxonomy for Distributed Real-time Control Systems," Advances In Computers, vol. 49, pp. 303–352, 1999.
[5] Computer Science Telecommunication Board, Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers, National Research Council, 2001.
[6] J. Hill, R. Szewczky, A. Woo, S. Hollar, D. Culler and K. Pister, "System Architecture Directions for Networked Sensors," in ACM Sigplan Notices, Vol. 35, pp. 93–104, 2000.
[7] S.S. Iyengar, K. Chakrabarty and H. Qi, "Distributed Sensor Networks for Real-time Systems with Adaptive Configuration," Journal of the Franklin Institute, vol. 338, pp. 571–582, 2001.
[8] M. Sathyanarayanan, "Pervasive Computing: Vision and Challenges," Pervasive Computing, vol. August, pp. 10–17, 2001.
[9] M. Weiser, "The Computer for the 21st Century," Scientific American, Vol. 265, No. 3, pp 94–104, 1991.
[10] S.S. Iyengar, D.N. Jayasimha and D. Nadig, "A Versatile Architecture for the Distributed Sensor Integration Problem," IEEE Transactions on Computers, Vol. 43, No. 2, February 1994.
[11] K. Marzullo, "Tolerating Failures of Continuous-valued Sensors," ACM Transactions of Computing Systems, Vol. 4, pp. 284–304, Nov. 1990.
[12] S. Sastry, "A SmartSpace for Automation," Assembly Automation, Vol. 24, No. 2, pp. 201–209, 2004.
[13] A. Tannenbaum, Modern Operating Systems, Prentice-Hall, NJ, 1992.
[14] P. Levis and D. Culler, "Mate: A Tiny Virtual Machine for Sensor Networks," in Architectural Support for Programming Languages and Operating Systems, 2002.
[15] D. Estrin, D. Culler, K. Pister and G. Sukhatme, "Connecting the Physical World with Pervasive Networks," IEEE Pervasive Computing, Vol. 1, No. 1, pp. 59–69, 2002.
[16] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," IEEE Personal Communications, Vol. 7, No. 5, pp. 16–27, 2000.
[17] J. Elson and D. Estrin, "Time Synchronization in Wireless Sensor Networks," in Proceedings of the 15th International Parallel and Distributed Processing Symposium, IEEE Computer Society, 2001.
[18] J. Heidman, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin and D. Ganesan, "Building Efficient Wireless Sensor Networks with Low Level Naming," in ACM Symposium on Operating Systems Principles, pp. 146–159, 2001.
[19] A.V. Lim, "Distributed Services for Information Dissemination in Self-Organizing Sensor Networks," Journal of the Franklin Institute, Vol. 338, pp. 707–727, 2001.
[20] J. Hill, P. Bounadonna and D. Culler, "Active Message Communication for Tiny Network Sensors," in INFOCOM, 2001.
[21] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, pp. 56–67, 2000.
[22] T. He, J.A. Stankovic, C. Lu and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-time Communication in Sensor Networks," Technical Report, University of Virginia, 2002.
[23] W.J. Caelli, "Security in Open Distributed Systems," Information Management and Computer Society, Vol. 2, No. 1, pp. 18–24, 1994.
[24] M. Horrell and Z. Limited, "The Security of Distributed Systems – An Overview," Information Security Technical Report, Vol. 1, No. 2, pp. 10–16, 1996.
[25] R. Kannan, S. Sarangi, S.S. Iyengar and L. Ray, "Sensor-Centric Quality of Routing in Sensor Networks," in Proceedings of IEEE Infocom, April 2003.
[26] National Institute of Science and Technology, DRAFT: Wireless Network Security, NIST, Special Publication 800-48, Computer Security Division, 2002.
[27] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," Computer Networks, vol. 38, pp. 393–422, 2002.
[28] A. Perrig, E. Szewczyk, V. Wen, D. Culler and J.D. Tygar, "SPINS: Security Protocols for Sensor Networks," in MOBICOM, pp. 189–199, 2001.
[29] S. Sastry and S.S. Iyengar, "Sensor Technologies for Future Automation Systems," Sensor Letters, Vol. 2, No. 1, pp. 9–17, 2004.

[30] A.D. Wood and J.A. Stankovic, "Denial of Service in Sensor Networks," IEEE Computer, pp. 54–62, October 2002.
[31] Siemens, Information Security for Industrial Communications, White Paper, 1999.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Knowledge Fusion in the Scalable Infosphere

## *State-of-the-Art and Advanced Technologies*

Alexander SMIRNOV
*Head of Computer Aided Integrated Systems Laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, 39, 14th Line, St. Petersburg, 199178, Russia*

**Abstract.** Current trends require the use of a global information environment, including end-users and loosely coupled knowledge sources (experts, knowledge bases, repositories, etc.) for decision making. This leads to an expansion in e-applications dealing with knowledge storing in the Internet and based on the intensive use of WWW-technologies and standards such as XML, RDF, DAML, etc. A vast diversity of knowledge management tools has made the problem of knowledge fusion (KF) from distributed sources crucial. The above necessitates the development of a KF approach to complex operations management (global understanding of ongoing processes, global knowledge exchange, etc.). The presentation discusses a Knowledge Source Network (KSNet) configuration approach to KF and its potential e-applications for a scalable information environment (infosphere). This approach is based on utilizing such technologies as ontology management, intelligent agents, constraint satisfaction, etc.

**Keywords.** Knowledge fusion, ontology management, agent, virtual reality

### 1. Introduction

Current trends in designing decision making systems in a wide range of applications require operating in a global information environment. This requirement has led to an expansion of tools dealing with knowledge storing in the Internet, based on the intensive use of WWW-technologies and such standards as XML, RDF, DAML+OIL, etc. [1,2]. Thus it is possible to speak about an evolution of the information environment, incorporating end-users and loosely coupled knowledge sources – KSs – (experts, knowledge bases (KBs, repositories, etc.), from "regular" (with fixed interactions between its elements) to "intelligent" (with flexible configuration of knowledge networks in which humans are involved)). The growing importance of knowledge, which emerged due to this evolution, results in a need for acquisition, integration, and transfer of the right knowledge from distributed sources in the right context to the right person in the right time for the right business purpose. These activities, called Knowledge Logistics (KL), are required for global awareness, dynamic planning and global information exchange in the information environment.

Here the described approach to KL through knowledge fusion (KF), called "Knowledge Source Network" (KSNet-approach), implies a synergistic use of knowl-

edge from different sources in order to complement insufficient knowledge and obtain new knowledge [3]. The architecture developed for the KF system (called system "KSNet") is based on this approach and utilizes such technologies as ontology management, intelligent agents, constraint satisfaction, soft computing, and groupware.

Intelligent agents and multi-agent systems are research topics which significantly changed the functioning of distributed systems. Multi-agent systems offer an efficient way to understand, manage and use the distributed, large-scale, dynamic, open, and heterogeneous computing and information systems [4,5]. In agent-based systems, an agent must represent its knowledge in the vocabulary of a specified ontology [6,7]. Ontology is a technique of semantic knowledge representation for further processing, and is considered as content theories of the kinds of objects, properties of objects and relations between objects possible in a specified knowledge domain, i.e. ontologies provide potential terms describing knowledge about the application domain [8]. An object-oriented constraint network paradigm was proposed as a general model of ontology representation in the KF system "KSNet" based on the KSNet-approach [9].

This paper has the following structure: (i) it shortly describes the state-of-the-art of the knowledge management areas related to KF; (ii) presents major technologies for KL (KF operations and an ontology-driven methodology), a knowledge repository structure and a multi-agent architecture of the system "KSNet;" and (iii) describes the system research prototype under development of this system and some case studies for KF in configuration tasks.

## 2. Knowledge Management: State-of-the-Art

We define *Data* as a set of facts which can exist in multiple forms at different locations and are often not interconnected and *Information* as a set of facts which are interconnected and delivered in a clear context and time. Information is more than a sum of its raw elements, which can be considered as data. *Knowledge* is defined as a set of relations (constraints, functions, rules) by which a user/expert decides how, why, where and what to do with the information to produce in a timely fashion adaptive actions meeting a goal or a set of goals. Knowledge may be considered as a high-value form of information that is ready to be applied to decisions and actions.

One of the research topics dealing with knowledge is Knowledge Management (KM). It is defined as a complex set of relations between people, processes and technology bound together with cultural norms, like mentoring and knowledge sharing, which constitute an organization's social capital [10]. KM consists of the following tasks: knowledge discovery (knowledge entry, capture of tacit knowledge, KF, etc.), knowledge representation (KB development, knowledge sharing and reuse, knowledge exchange, etc.), knowledge mapping (identifying KSs, indexing knowledge, making knowledge accessible) [11–15]. A number of different approaches have been proposed and tools have been developed to solve these tasks based on the algorithms of data searching and retrieving in large databases, technologies of data storing and representation, etc. Among them the following can be pointed out: Microsoft SharePoint Portal [16], SearchServer/ KnowledgeServer [17], Lotus Discovery Server [18], Text-To-Onto [19], etc. (knowledge searching and retrieving from different types of documents); Disciple-RKF [20], EXPECT [21], Trellis (EXPECT's successor) [22], COGITO [23], TKAI [24], OntoKick [25], etc. (knowledge acquisition from experts and tacit knowledge revealing); OntoEdit [26], Protégé [27], OntoLingua [28],

etc. (ontologies engineering); HPKB [29], AKT [30] etc. (KBs organization and development); KRAFT [31], InfoSleuth [32], Observer [33], etc. (knowledge and information integration). The above approaches are targeted at pertinent, clear, recent, correct information and knowledge processing, and timely delivery to locations of need for global situational awareness and the ability to predict the development of ongoing processes at the level of understanding. From this point, there arises a need for KL [34].

The possible application domains of KL belong to the following areas:

- large-scale dynamic systems (enterprises) with distributed operations in an uncertain and rapidly changing environment, where the collection, assimilation, integration, interpretation, and dissemination of information are required [35,36];
- focused logistics operations and/or Web-enhanced logistics operations addressing the sustainment, transportation and end-to-end rapid supply to the final destination, where the distributed information management and real-time information/KF to support continuous information and knowledge integration of all participants of the operations are needed [37];
- markets via partnerships with different organizations, where the dynamic identification and analysis of information sources, and providing for the interoperability between market participants (players) in a semantic manner are required [38–40].

For all of the above areas it is possible to describe management systems as an organizational combination of people, technologies, procedures and information/knowledge.

The KL is based on individual user requirements, available KSs, and content analysis in the information environment. Hence, systems operating in this area must react dynamically to unforeseen changes and unexpected user needs, keep up-to-date resource value assessment data, support rapid execution of complex operations, and deliver personalized results to the users/knowledge customers. Here the proposed approach to KL is realized through KF – the integration of knowledge from different sources (probably heterogeneous) into a combined resource in order to complement insufficient knowledge and obtain new knowledge. Development of this scientific direction has come a long way from data fusion (Figure 1), which arises from multisensor data fusion, in which information from a number of sources is integrated to form a unified picture [42].

## 3. KSNet-Approach

### *3.1. Knowledge Sources*

The following KS types were identified: (i) experts, who directly enter knowledge related to user requests using built-in mechanisms, (ii) KBs, (iii) databases, (iv) structured documents – text, HTML, XML, etc. documents (the relevance of a document to a request can be estimated using indexed keywords) and (v) other sources, for which mechanisms of knowledge recognition and capturing are available.

KSs fall into two groups: (i) passive sources (available external data and KBs, structured documents, other sources with some developed mechanisms of interaction) providing knowledge “on demand;” and (ii) active sources (experts, KM tools) provid-

ing knowledge "on demand" and pro-activity functions "Just-Before-Time"– support for request processing.

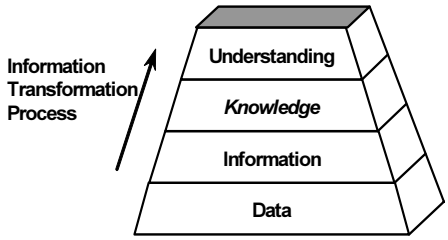



**Figure 1.** Conceptual framework of information support (adapted from [41]).

### *3.2. Knowledge Fusion*

In [43] the most complete sequence of main operations for KM referred to as a knowledge chain was proposed. It was used as a basis for the development of a KF process structure (Figure 2) consisting of: (i) capturing knowledge from KSs and its translation into a form suitable for supplementary use, (ii) acquisition of knowledge from external sources, (iii) selection of knowledge from internal sources (local KBs), (iv) knowledge generation: producing knowledge by discovering or deriving from existing knowledge, (v) internalization: changing system knowledge by saving acquired, selected and generated knowledge, (vi) externalization: embedding knowledge into the system's output for release into the environment, (vii) KF management: planning, coordination, collaboration, and control of operations constituting the KF process. In Sect. 4.4 the user processing scenario using most of these operations is presented.

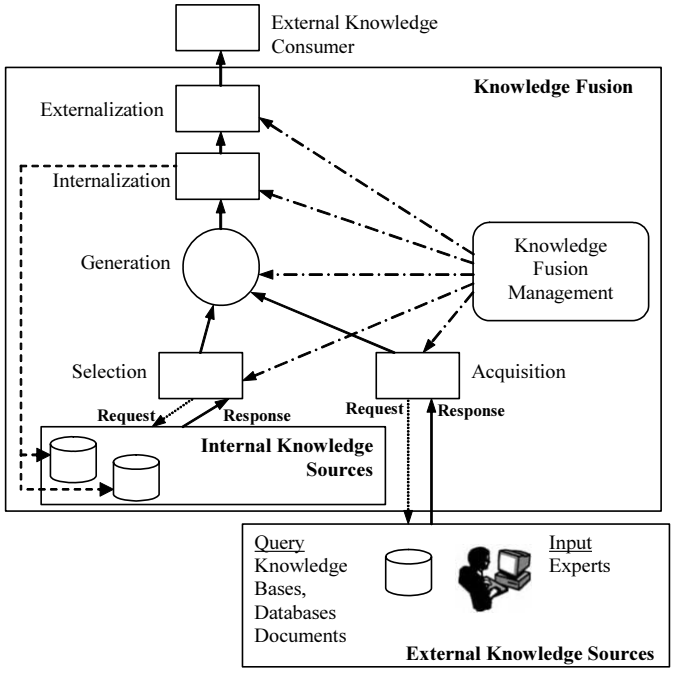



**Figure 2.** Operations of the knowledge fusion process.

To increase the KF rapidity it is necessary not only to find required sources but also to identify their usefulness for solving a particular problem. For this purpose it is reasonable to: (i) use a user profile (structured information about the user), (ii) offer tips and hints to the user to reveal tacit user interests, (iii) utilize techniques of knowledge/ontology reuse, (iv) perform indexation of stored knowledge, and (v) increase intelligibility of knowledge representation for the users involved in the processes of development, edition, update, etc.

### *3.3. Knowledge Source Network*

The network of loosely coupled sources located in the information environment is referred to as "Knowledge Source Network" (KS network). The term KS network originates from the concept of virtual organization based on the synergistic use of knowledge from multiple sources. Figure 3 roughly explains the basic concepts of the KS networks and their multi-level configuration. The upper level represents a customer-oriented knowledge model based on a fusion of knowledge acquired from KS network units (KSs), which constitute the lower level and contain their own knowledge models.

### *3.4. Ontology-Driven Methodology for Knowledge Fusion*

The following ontology types for the "KSNet" systems were defined: (i) top-level ontology describes notation for application domain description, (ii) application ontology (AO) contains terms and knowledge structures describing a particular application domain, (iii) preliminary KS ontology (KSO) contains KS knowledge terms and structure in the top-level ontology notation, (iv) KSO contains correspondence between terms of KS and AO, (v) preliminary request ontology contains terms which can be used by a user for request input and structure in the top-level ontology notation, (vi) request ontology contains correspondence between terms of preliminary request ontology and AO. The ontologies are stored in the common ontology library (OL) where they can be reused.

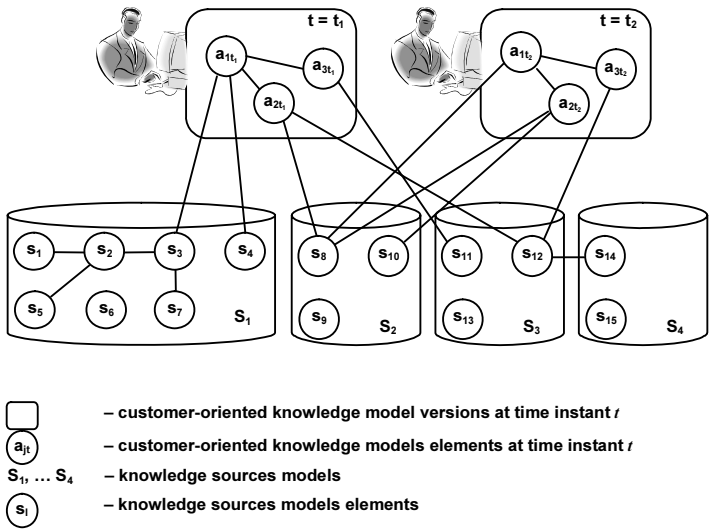



**Figure 3.** Distributed multi-level knowledge fusion management as the KS network configuration.

The system works in terms of a common vocabulary. AO is based on domain, task and method ontologies also stored in OL. Each user/user group works in terms of an associated expandable request ontology and thereby with a part of the AO pertinent to the user/user group. *User profiles* are used during interactions to provide for an efficient personalized service. Every user request consists of two parts: (i) a structural constituent (containing the request terms and relations between them), and (ii) a parametric constituent (containing additional user-defined constraints). For request processing, an auxiliary KS network configuration is built defining when and what KSs are to be used for the request processing in the most efficient way. For this purpose a *knowledge map* (see Sect. 4.2) including information about locations of KSs is used. The translation between the system's and the KS' notations & terms is performed using KSOs. A conceptual scheme of the user-oriented ontology-driven KF methodology is presented in Figure 4.

The formalism of object-oriented constraint networks has been chosen for the ontology representation. An abstract KS network model is based on this formalism. This solution was mainly motivated by such factors as the support of declarative representation, efficiency of dynamic constraint satisfaction, and problem modelling capability, maintainability, reusability, and extensibility of the object-oriented technology.

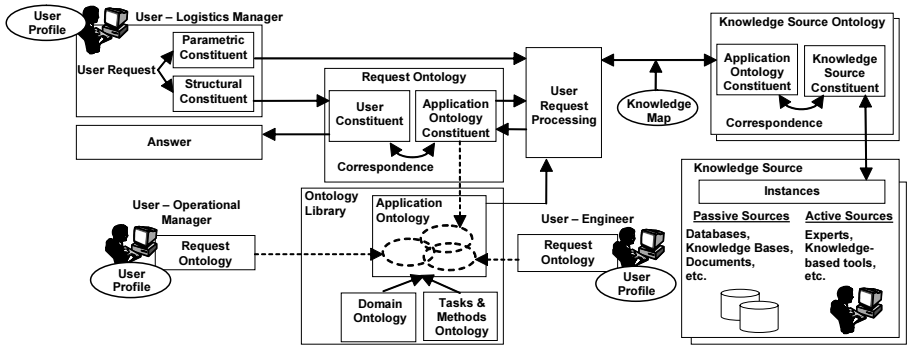



**Figure 4.** Conceptual scheme of the user-oriented ontology-driven KF methodology.

According to the chosen formalism, an ontology ($A$) is defined as: $A = (O, Q, D, C)$, where $O$ – a set of *object classes* ("*classes*"). Each of the entities in a class is considered as an *instance* of the class. This set consists of two subsets: $O = O^I \cup O^{II}$, where $O^I$ – a set of *non-primitive* classes i.e. classes which can have instances ($O^I = \{o: \exists\ instance(o)\}$), $O^{II}$ – a set of *primitive* classes i.e. classes which cannot have instances ($O^{II} = \{o: \neg\exists\ instance(o)\}$); $Q$ – a set of class attributes ("*attributes*"); $D$ – a set of attribute domains ("*domains*"); $C$ – a set of *constraints*.

For the chosen notation the following six types of constraints have been defined: $C = C^I \cup C^{II} \cup C^{III} \cup C^{IV} \cup C^V \cup C^{VI}$, where $C^I$ – accessory of attributes to classes ($C^I = \{c^I\}$, $c^I = (o, q)$, $o \in O$, $q \in Q$); $C^{II}$ – accessory of domains to attributes ($C^{II} = \{c^{II}\}$, $c^{II} = (o, q, d)$, $o \in O$, $q \in Q$, $d \in D$); $C^{III}$ – classes compatibility (compatibility structural constraints) ($C^{III} = \{c^{III}\}$, $c^{III} = (\{o\}, True \vee False)$, $|\{o\}| \geq 2$, $o \in O$); $C^{IV}$ – hierarchical relationships (hierarchical structural constraints) "is a" defining class taxonomy ($type$=0), and "has part"/"part of" defining class hierarchy ($type$=1).

($C^{IV} = \{c^{IV}\}$, $c^{IV} = \langle o', o'', type \rangle$, $o' \in O$, $o'' \in O$, $o' \neq o''$); $C^V$ – associative relationships ("one-level" structural constraints) ($C^V = \{c^V\}$, $c^V = (\{o\})$, $|\{o\}| \geq 2$, $o \in O$); and

$C^{VI}$ – functional constraints referring to the names of classes and attributes ($C^{VI} = \{c^{VI}\}$, $c^{VI} = f(\{o\}, \{q\}) \rightarrow True \vee False$, $|\{o\}| \geq 0$, $|\{q\}| \geq 0$, $o \in O$, $q \in Q$).

The most abstract class of the ontology (the top of the ontology's taxonomy) is "Thing." $o \in O$, $q \in Q$, $d \in D$, $c \in C$ are considered as ontology elements.

## 4. Knowledge Fusion System "KSNet"

### *4.1. Organizational Principles*

As a result of the analysis of modern systems for KM and information/knowledge fusion [31–33,44,45] the major organizational principles of the "KSNet" system based on the KSNet-approach have been formulated as follows: (i) scenarios and procedures of the system are developed independently of application domain; (ii) the system must deal with a specific application domain; (iii) the system must provide an interface for request input and result representation; (iv) the system must perform a translation of the entered request into application domain terms, decomposition of the request into its components (subrequests), recognize the subrequests and send them to processing (identifying suitable KSs and creating a special configuration of the KS network, querying identified sources, filtering them according to user-defined constraints, fusion of knowledge from different sources, validation, check for meeting requirements, present results to the user), (v) subrequests are processed simultaneously, (vi) the request can be passed to experts specializing in the application domain, (vii) results must be recorded, internal information components of the "KSNet" system have to be changed for supplementary reuse in similar requests.

### *4.2. Main Components*

In accordance with the above organizational principles the following components of the "KSNet" system were identified:

1. software components: (a) methods, (b) agents, (c) interface for user request input, for new knowledge entry by an expert, for operations with application domain (import and creation of AO, searching and ranking of KSs, preparation of special interface forms (request templates) for knowledge customers and administrators), for OL support (ontologies import and export, maintenance, diagnostics), etc;
2. repository: (a) ontologies (top-level ontology, AO, preliminary KSO, KSO, request ontology, and preliminary request ontology) and (b) information components (internal KB, knowledge map, user profiles);
3. service tables.

The above components are related to each other. Information about all the elements dealt with by the system (KSs, experts, users, tools, etc.) and main terms describing the application domain are stored in the service tables, and all active elements (users, active KSs, etc.) access them via specially designed software interfaces. Service tables are created and maintained via a database management system (DBMS).

Service tables are meant to store information describing the application domain, user profiles, parameters of KSs, namely: (i) contents of ontologies, user profiles, internal KB, knowledge map, (ii) links to methods and KSs, and (iii) information about the "KSNet" system users, wrappers, and auxiliary reference information.

Figure 5 presents the "KSNet" system repository structure. In the repository structure three components were defined. A *semantic component* is used for knowledge representation in a common notation and terms.

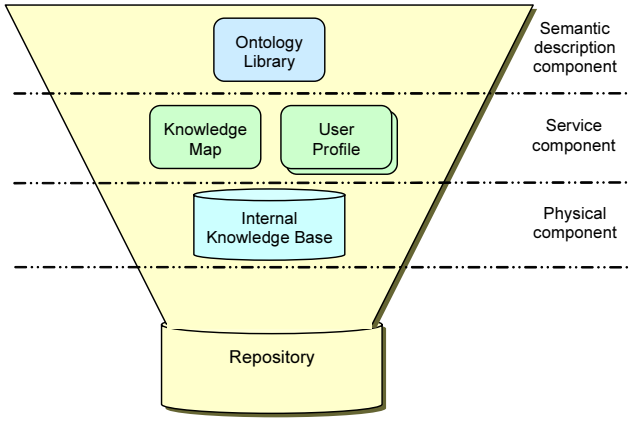



**Figure 5.** The "KSNet" system repository structure.

A *service component* is used for knowledge indexing and search and contains the following components:

- the knowledge map includes information about locations of KS network units utilized during problem solving and information about alternative sources (KS network units) containing similar information and KSs characteristics. Monitoring tools perform permanent checking of KSs availability and perform appropriate changes in the knowledge map. The knowledge map is meant to facilitate and speed up the process of the KSs choice;
- the user profile is an organized storage of information about the user, his/her requests history, etc. This component is used for a number of purposes (faster search due to analyzing and utilizing request history and user preferences, Just-before-Time request processing, etc.).

A *physical component* contains internal KB, used for storage, verification and reuse of knowledge (i) entered by experts, (ii) learned from users (knowledge consumers), (iii) obtained as a result of the KF process, (iv) acquired from KSs which are not free, not easily accessible, etc.

### *4.3. Multi-Agent Architecture*

Like some other KM systems, the "KSNet" system uses intelligent software agents to provide access to distributed heterogeneous KSs [46–48]. Table 1 describes some special features of the agents, used in the "KSNet" system.

**Table 1.** Features of the agents of the "KSNet" system

| agent | life time | quantity | general tasks |
|---|---|---|---|
| wrapper | KSs life time | number of KS types | translates knowledge from source terms into the AO terms and sends requests from system to sources. |
| mediator | Task execution time | number of tasks being processed | tracks out task processing step-by-step from input to result. Provides negotiations with the expert assistant agents during alternative KS ranking. Stores temporary results. |
| facilitator | system life time | 1 | provides a "yellow pages" directory service for the agents. |
| user agent | as long as user is registered in the system | number of registered users | provides a user personalization service: provides a set of functions for the user profile processing, facilitates request input, provides a set of tips and hints for the user, and passes messages and information from the system to the user. |
| translation agent | system life time | number of request input interfaces | provides for translation of terms between the users and the system, between the application domain and KSs. Uses the request ontology, and AO. |
| expert assistant agent | as long as the expert is registered in the system | number of registered experts | facilitates the process of expert knowledge entry into the system. Supports the process of alternative KS ranking. Updates the user profiles. |
| configuration agent | system life time | 1 | configures KS network using the knowledge map and the user profiles. Performs scheduling functions. Negotiates with the KF agents and the wrappers. |
| knowledge fusion agent | system life time | varying | obtains knowledge from the mediator and processes it. Generates new knowledge. Validates it. Interacts with the monitoring agent. |
| monitoring agent | system life time | 1 | provides a set of functions for diagnostics of the system repository base and external KSs. |
| ontology management agent | system life time | 1 | provides a set of functions for ontology engineering and operation – creation of ontologies for new KSs, modification of AO etc. Checks correspondence between KS and request ontologies and AO. |

Multi-agent system architecture, based on the Foundation for Intelligent Physical Agents (FIPA) Reference Model [49] as an infrastructure for the definition of agent properties and functions, was chosen as a technological basis for the "KSNet" system since it provides standards for heterogeneous interacting agents and agent-based systems, and specifies ontologies and negotiation protocols to support interoperability in specific application areas. FIPA-based technological kernel agents used in the system are: wrapper (interaction with KSs), facilitator ("yellow pages" directory service for the agents), mediator (task execution control), and user agent (interaction with users). The following problem-oriented agents specific for KF, and scenarios for their collaboration were developed: translation agent (terms translation between different vocabularies), KF agent (KF operation performance), configuration agent (efficient use of KSNet),

ontology management agent (ontology operations performance), expert assistant agent (interaction with experts), and monitoring agent (KSs verifications).

A major set of agents is represented in Figure 6 according to the above described principles and functions of the "KSNet" system.

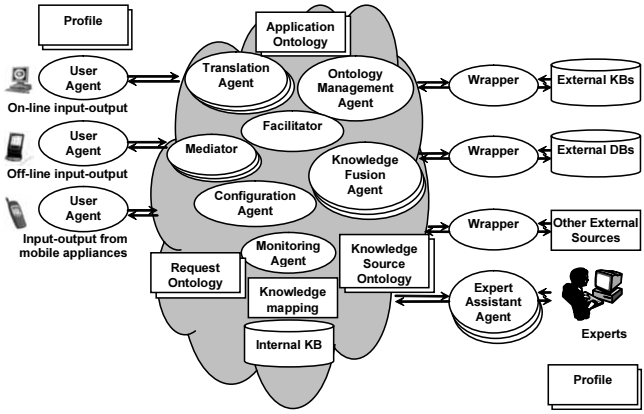



**Figure 6.** Basic components of the multi-agent "KSNet" system.

Each agent of the "KSNet" system contains the following modules [50]: (i) identifying, (ii) functional, and (iii) repository. The identifying module contains such parameters as unique identifier, creation date and time, etc. This module structure depends on agent type (some agents do not need this module). The Functional module contains a set of procedures to be executed by the agent. The Repository contains special information, such as agent's knowledge, history of the agent's contacts, temporary results, new knowledge, etc. Identifying and functional agent modules are shown in Figure 7. The agents' connectivity matrix is presented in Figure 8.

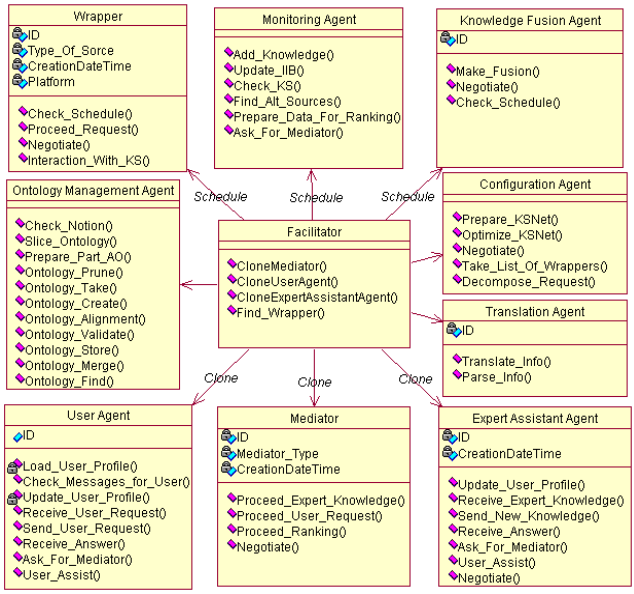



**Figure 7.** Class diagram of KSNet agents with main properties and functions.

| callee / caller | wrapper | mediator | facilitator | user agent | translation agent | expert assistant agent | configuration agent | KF agent | monitoring agent | ontology management agent |
|---|---|---|---|---|---|---|---|---|---|---|
| Wrapper | | P | P | | P | | | | | P |
| Mediator | P | | | P | P | P | P | P | P | |
| Facilitator | | P | | | | | | | | |
| user agent | | M/P | P | | | | | | | |
| translation agent | | | | | | | | | | P |
| expert assistant agent | | M/P | P | | | | | | | |
| configuration agent | M/P | P | P | | | | | M/P | | |
| KF agent | | P | P | | | | | | P | |
| monitoring agent | | M/P | | | | | | | | |
| ontology management agent | P | | | | P | | | | P | |

**Figure 8.** Class agents' connectivity matrix. P – peer-to-peer interaction, M – mediating interaction.

### *4.4. Major Scenarios*

The "KSNet" system life cycle consists of two major phases:

- the *preparation phase*: it includes the following tasks: (i) study of the application domain, creation of AO based on existing OL and description of the application domain; (ii) search for KSs related to the application domain and creation of KSOs and (iii) configuration of the KS network – knowledge distribution within KSs;
- the *operation phase*: it includes the following tasks: (i) offer of end-user interface for the entry of knowledge search requests by knowledge consumers; (ii) selection of KSs related to user requests and configuration of the KS network; (iii) selection, acquisition, fusion and verification of acquired and generated knowledge; (iv) presentation of results to the user; (v) storage of results. The configuration of the KS network consists of: (i) selection of KSs which are to be included in the KS network; (ii) negotiation between the KS network units; and (iii) scheduling and coordination of the KS network.

Below, one of the major "KSNet" system scenarios of user request processing during the operation phase (Figure 9) is presented. When a user request is received by the "KSNet" system the request terms are translated into system terms using the request ontology. Based on the translated request a part of the AO is formed which describes an object-oriented constraint network for user request processing.

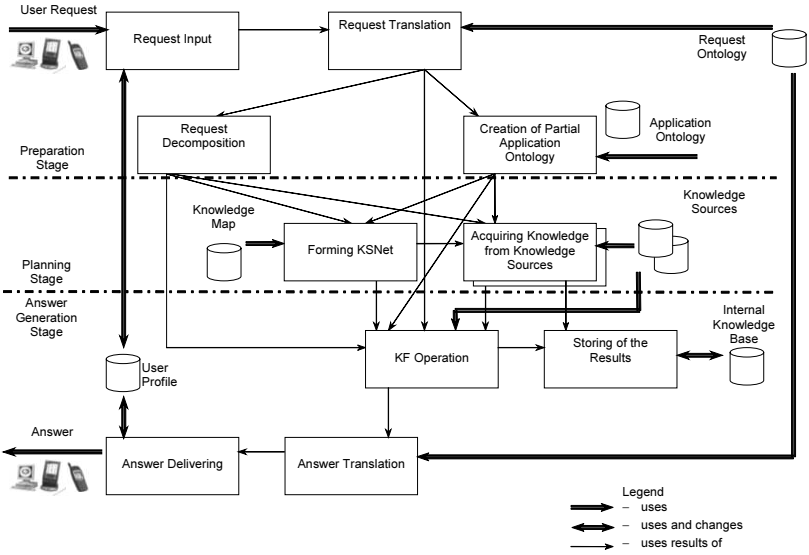



**Figure 9.** Collaboration diagram of user request processing scenario.

This object-oriented constraint network is a basis for building requests to KSs. A request to a KS is translated into the terms (using KSO) and notations (using preliminary KSO) of a KS. An answer from a KS is translated back into the notation and terms of the "KSNet" system and passed to the constraint network of the user request processing. The results of the processing are analyzed by the system and can be added to the internal KB for their possible reuse and/or in AO. The user request processing is completed by translation of the request processing results into user terms and presenting them to the user. All translation operations are performed using appropriate ontologies.

This step by step scenario supported by multi-agent architecture is presented below (Figure 10).

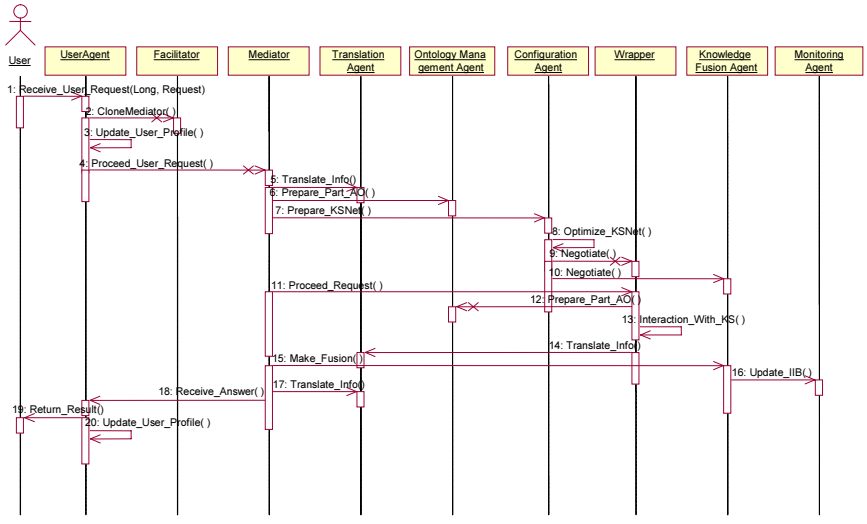



**Figure 10.** Sequence diagram of user request processing.

### *4.5. Knowledge Fusion Patterns*

In the "KSNet" systems the process of KF takes place during the performance of different tasks. The analysis of major system scenarios has made it possible to select of a list of generic KF patterns for these operations (Figure 11).

A definition of the developed KF patterns can be illustrated via the following example. Two initial KSs (A and B) with some structures of primary knowledge units are given. There is a tacit relation between two primary knowledge units, namely a3 from A and b2 from B. It is necessary to fuse two sources while preserving the internal knowledge structure and revealing the above tacit relation.

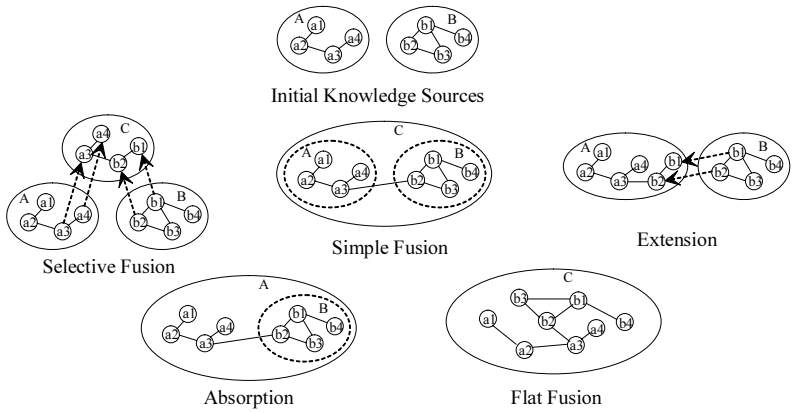



**Figure 11.** Knowledge fusion patterns.

- selective fusion (AO and KSO creation). A new KS is created, which contains required parts of the initial KSs. Initial KSs preserve their internal structures and autonomy.
- simple fusion (OL creation and maintenance). A new KS is created, which contains initial KSs. Initial KSs preserve their internal structures and lose (partially or completely) their autonomy.
- extension (*knowledge map* and *internal KB* maintenance). One of the initial KSs is extended so that it includes the required part of other initial KSs, which preserves its internal structure and autonomy.
- absorption (a new KS connection to the system). One of the initial KSs is extended so that it includes other initial KSs, which preserves its internal structure while losing (partially or completely) its autonomy.
- flat fusion (KF at the operation phase). A new KS is created, which contains initial KSs. The initial KSs dissolve within the new KS and do not preserve their internal structures and autonomy.

Based on the definition of the KF patterns different patterns have been chosen for different tasks of the "KSNet" system (Table 2). The use of KF patterns accelerated the KF process due to the typification of fusion schemes.

## 5. Prototypes and Examples

The main goal of the case study described below is to test the implementation of the KSNet-approach for complex dynamic systems – "product-process-business organization (business)" systems – of different configuration types: (i) marketing/order configuration, (ii) product configuration, (iii) upgrade/add-on configuration, (iv) distributed process configuration, (v) business network unit configuration, and (vi) whole business network configuration.

**Table 2.** Usage of knowledge fusion patterns

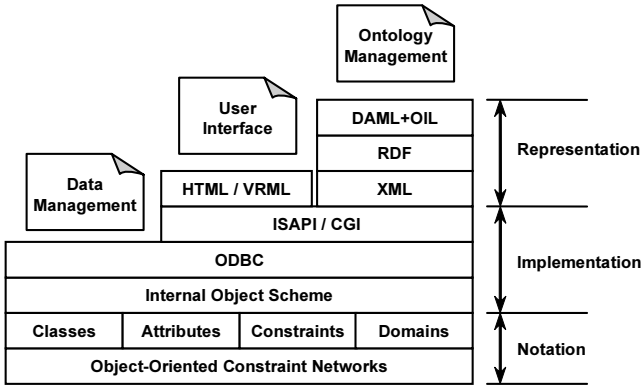
| Task | Description | KF patterns |
|---|---|---|
| AO and KSO creation during preparation and operation phases | new ontologies are built using elements of existing ontologies or KSs | selective fusion |
| OL creation and maintenance during preparation and operation phases | OL contains different ontologies | simple fusion |
| *knowledge map* and *internal KB* maintenance during operation phase | *knowledge map* and *internal KB* are extended when new KSs are connected to the system | extension |
| a new KS connection to the system | the new KS becomes a part of the system from the user point of view | absorption |
| KF at the operation phase | knowledge from different sources is used for generation of new knowledge | flat fusion |

### *5.1. Distributed Architecture of Prototypes*

The key points for the project to be tested and prototyped were conditioned by covering all the KF patterns (mainly concentrated on the preparation phase of the "KSNet" system lifecycle), KS network configuration (as a main element providing for the rapidity of the KF process) and constraint network processing (as a basis technology providing for the user request processing). In accordance with up-to-date technologies and standards, the information kernel for KL is built as shown in Figure 12.

As described above, the knowledge is represented by an aggregate of interrelated classes, their attributes, attribute domains and relations between them. An object scheme for working with the knowledge and database structure for its internal storage is designed based on this notation. Access to the database is performed via ODBC as a standard data access mechanism under the MS Windows operating system. Remote access to the stored knowledge is performed via the common HTTP Internet protocol. Knowledge is represented via either interactive HTML+VRML Java enabled pages for users or a DAML+OIL-based format for knowledge-based tools.

In order to increase the rapidity of the KF process in the "KSNet" system the following supporting tasks were defined (Figure 13): (i) the knowledge map creation utilizing alternative KS ranking, (ii) KS network configuration based on the task of efficient KS choice, and (iii) user request processing based on constraint network processing.




DAML - The DARPA Agent Markup Language
OIL – The Ontology Inference Layer
RDF – Resource Description Framework
XML – Extensible Markup Language
HTML – Hyper Text Markup Language

VRML – Virtual Reality Marjup Language
ISAPI – Internet Server Application Programming Interface
CGI – Common Gateway Interface
ODBC – Open DataBase Connection

**Figure 12.** Standards of knowledge logistics information kernel.

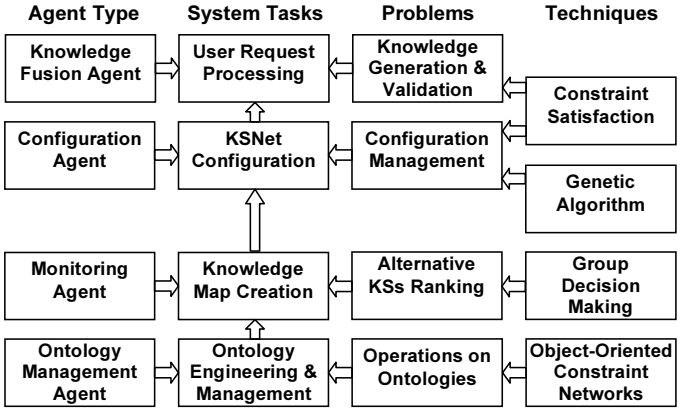



**Figure 13.** Main system tasks and techniques.

These tasks require the development and application of appropriate mathematical mechanisms (models and methods). The "MultiExpert" system, based on group decision support techniques, is used for the knowledge map creation. An application based on a genetic algorithm is used for the KS network configuration. An application based on ILOG constraint satisfaction technology is used for constraint network processing. The architecture of the "KSNet" system research prototype is presented in Figure 14.

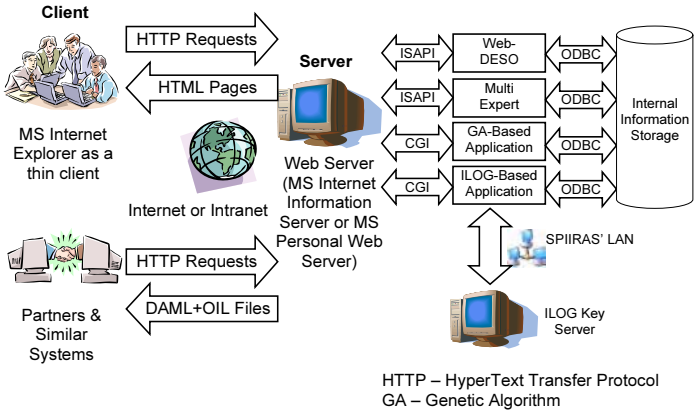



**Figure 14.** The architecture of the research prototype of the "KSNet" system.

### *5.2. Utilizing ILOG for Knowledge Fusion*

For the implementation of constraint networks, utilizing the features of the ILOG Configurator [54] package is proposed for representing the task in the object-oriented form (Figure 15).

In order to verify the application of ILOG for KF the following two prototypes of configuration tasks were developed: (i) a resource allocation example – a production system for car assembly and (ii) a product configuration example – a car configuration. The main idea of the system configuration task (in the above example, a car) is to obtain a feasible configuration for a system meeting specified requirements, with the system structure being known. The task of resource allocation assumes that there is some amount of work to be done and some facilities which can perform this work. The work consists of several operations (parallel and/or sequential) and each facility is capable of performing some of the operations.

At the preparation phase, an ontology engineer creates the AO "Supply Chain Management" for configuration task solving. A supply chain consists of production units capable of performing a number of operations. Every component (node) is described as a set of attributes/properties and a set of possible solutions/templates. Both products and units are described in a domain ontology.

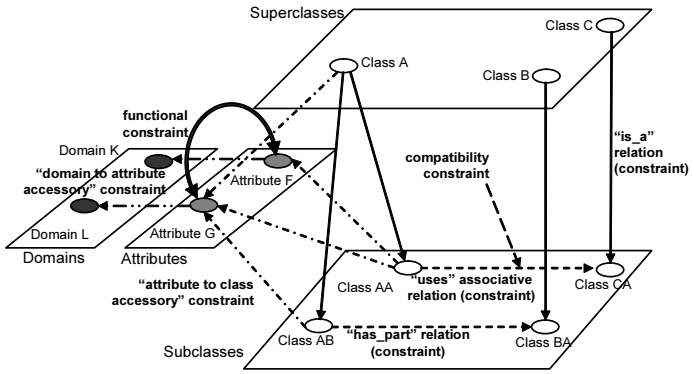



**Figure 15.** ILOG constraint satisfaction model adapted to the knowledge representation formalism.

In this example the system's OL contains two domain ontologies ("Management" and "Supply Chain"), and a tasks & methods ontology (Figure 16). Based on these ontologies the "Supply Chain Management" AO was developed (Figure 17).

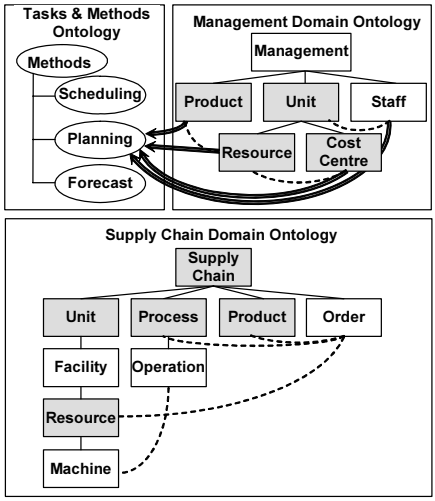



**Figure 16.** Ontology library containing task & methods and domain ontologies.

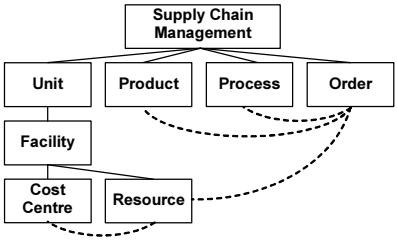



**Figure 17.** Supply Chain Management ontology.

In the "KSNet" system, users can input their requests in two main ways: (i) in a free form, and (ii) using specially developed templates (dynamic software forms with fields for entering request terms, constraints and criteria). Requests entered in a free form are passed to the system as plain text, e.g.: "Configure a supply chain (SC) in the part of car components production allocation to SC participants (SC participants are to be found). In accordance with the order the production costs must be minimal, the car's engine volume must be 2.0 l., and the total cost is equal to or less than $25000." Then this request is recognized by the translation agent. The ontology management agent finds the correspondence between ontology elements and user request terms. The configuration agent extracts information from the knowledge map; it finds KSs containing information for user request processing. It negotiates with wrappers (price, schedules, capabilities, etc.), defines appropriate KSs and prepares the KS network configuration for the user request. The wrappers define parts of the request related to their KSs and pass them to the KSs, receive a response from the KSs, and transform the response into the system's notation. The KF agent performs the fusion of received knowledge: it pre-

pares input data for the ILOG Configurator, calls its functions and receives an answer. The user agent returns the results to the user, updating the user profile.

Figure 18 presents an example sequence of templates for user request input and answer representation considering the configuration of a car and a supply chain for its production in accordance with user preferences (based on the free form request given above). The production process consists of three parallel tasks: (i) body production, (ii) engine production, and (iii) transmission production. The facilities are the plants, with known capacities and such characteristics as production cost and time. The goal is cost minimization within a time limit or time minimization within a cost limit.

Since the examples were implemented as web-based applications they can also be considered as prototype of the "Multi-component product e-configuration tool" [57].

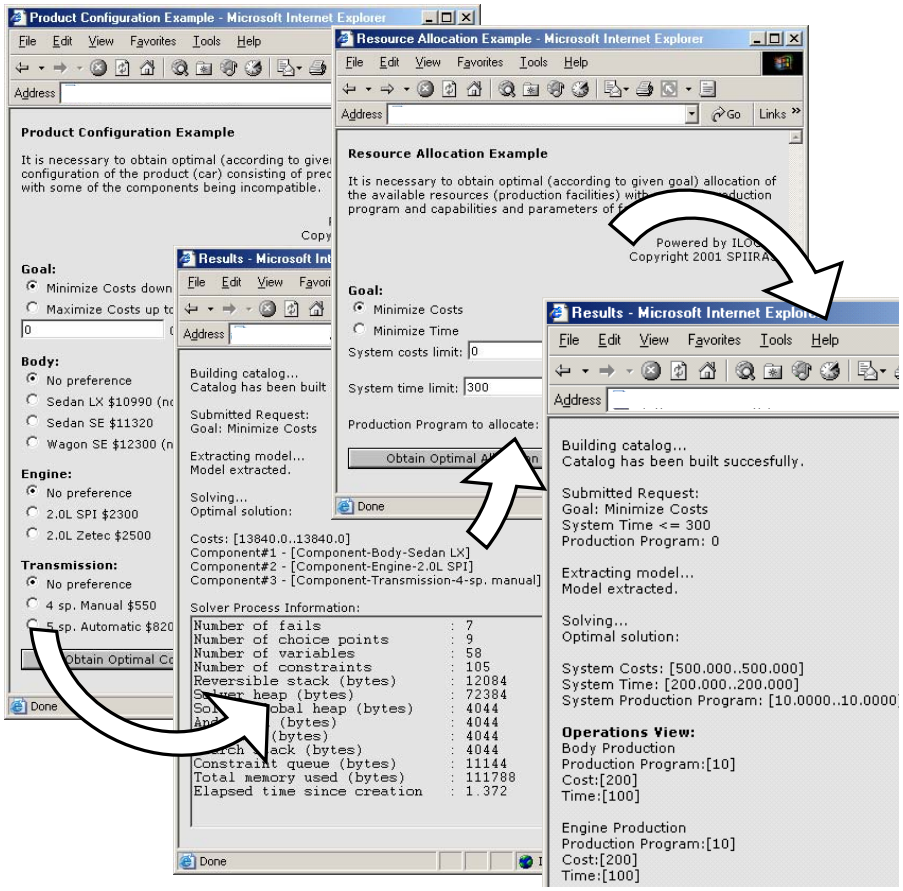



**Figure 18.** User request input and output forms for car configuration and resource allocation examples.

Both types of configuration tasks described above were incorporated in a prototype of a mobile hospital formation. The Binni scenario was chosen as a case study for this prototype. It is a hypothetical scenario based on the Sudanese Plain [58]. The aim of the Binni scenario is to provide a multi-agent environment, focusing on new aspects of coalition problems and new technologies demonstrating the ability of coalition-oriented agent services to function in an increasingly dynamic environment [59,60]. The ex-

perimentation with the Binni scenario is intended for a demonstration of how the developed KSNet-approach can be used for supply chain management, logistics and other coalition operations problems.

The general problem considered in this case study for the system "KSNet" has the following formulation: "Define suppliers, transportation routes and schedules for building a hospital of given capacity at given location by given time." The following subproblems were selected:

- hospital related information (constraints on its structure, required quantities of components, required delivery schedules);
- available suppliers (constraints on supplier capabilities, capacities, locations);
- available providers of transportation services (constraints on available types, routes, and time of delivery);
- geography and weather of the Binni region (constraints on types, routes, and time of delivery, e.g. by air, by trucks, by off-road vehicles);
- political, economical, or social situation, e.g. who occupies the territory used for transportation, existence of military action on the routes, etc. (additional constraints on routes of delivery).

As a result of the analysis of these problems, the following modules were defined:

1. *portable hospital allocation*: this subproblem is devoted to finding the most appropriate location for a hospital to be built considering such factors as location of the disaster, water resources, nearby cities and towns, communications facilities (e.g., locations of airports, roads, etc.) and the decision maker's choice and priorities;
2. *routing problem*: this subproblem is devoted to finding the most efficient ways to deliver hospital components from available suppliers, considering such factors as communications facilities (e.g., locations of airports, roads, etc.), their conditions (e.g., good, damaged or destroyed roads), weather conditions (e.g., rains, storms, etc.) and the decision maker's choice and priorities;
3. *hospital configuration*: this subproblem is devoted to finding the most efficient components for the hospital considering such factors as component suppliers, their capacities, prices, transportation time and costs and the decision maker's choice and priorities.

Input data for user request input is prepared by an expert team using specially developed screen forms. Experts' tasks included a list of suppliers, a specification of dependencies between the weather and delivery types (routes), and the analysis of hospital supplies delivery costs. Parts of ontologies corresponding to the described task were found in the Internet's ontology libraries [61–66]. The application ontology of this humanitarian task was built and a connection of found sources was performed. Three wrappers were developed to process information about: (i) suppliers, (ii) transportation service, and (iii) weather conditions and prepared HTML forms for user request input.

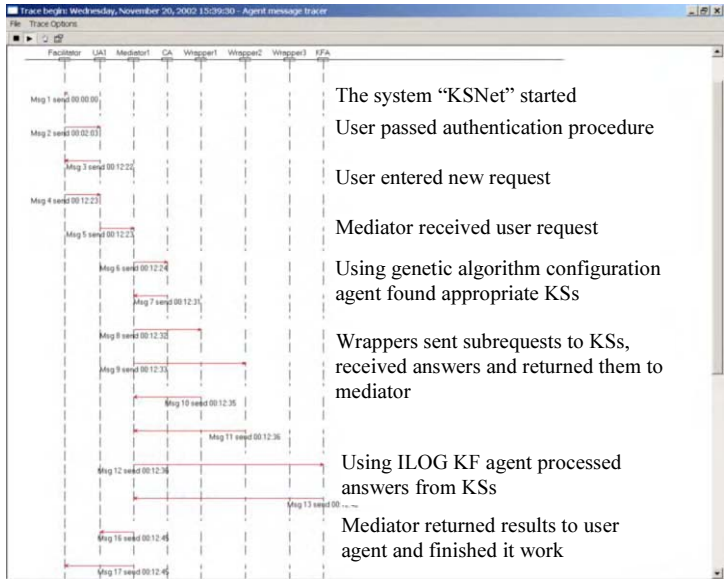



**Figure 19.** An example of agents' interaction.

One of the scenarios of agent community interaction during user request processing is given in Figure 19. According to the prototype scenario, the user request processing starts with pointing out on the map of the Binni region a desirable location of the hospital to be built. The map is updated and possible locations closest to the one pointed out by the user are shown. These locations were entered into the system by experts, taking into account such facts as availability of water resources, roads, surrounding areas and cities. The user selects a desirable destination among those suggested by the system. Besides the hospital destination, the system requests additional hospital characteristics such as hospital capacity or furniture, and medical equipment volume. The characteristics are the result of the ontologies analyses and, as a rule, correspond to attribute values needed to solve the task. After the information is entered by the user a set of knowledge sources containing information relevant to the user request is formed.

The result (for target values of hospital location, furniture volume, and furniture suppliers) of processing the considered request is presented in Figure 20.

## 6. Discussion

Comparison of the "KSNet" system with some other existing systems/projects oriented toward KF is presented in Table 3. These systems are:

- KRAFT (Knowledge Reuse and Fusion/Transformation), a multi-agent system for the integration of heterogeneous information systems. The main aim of this project is to enable sharing and reuse of constraints embedded in heterogeneous databases and knowledge systems. It contains a hierarchy of shared ontologies for local resource ontology translation;
- InfoSleuth, a multi-agent system for retrieving and processing information in a network of heterogeneous information sources.

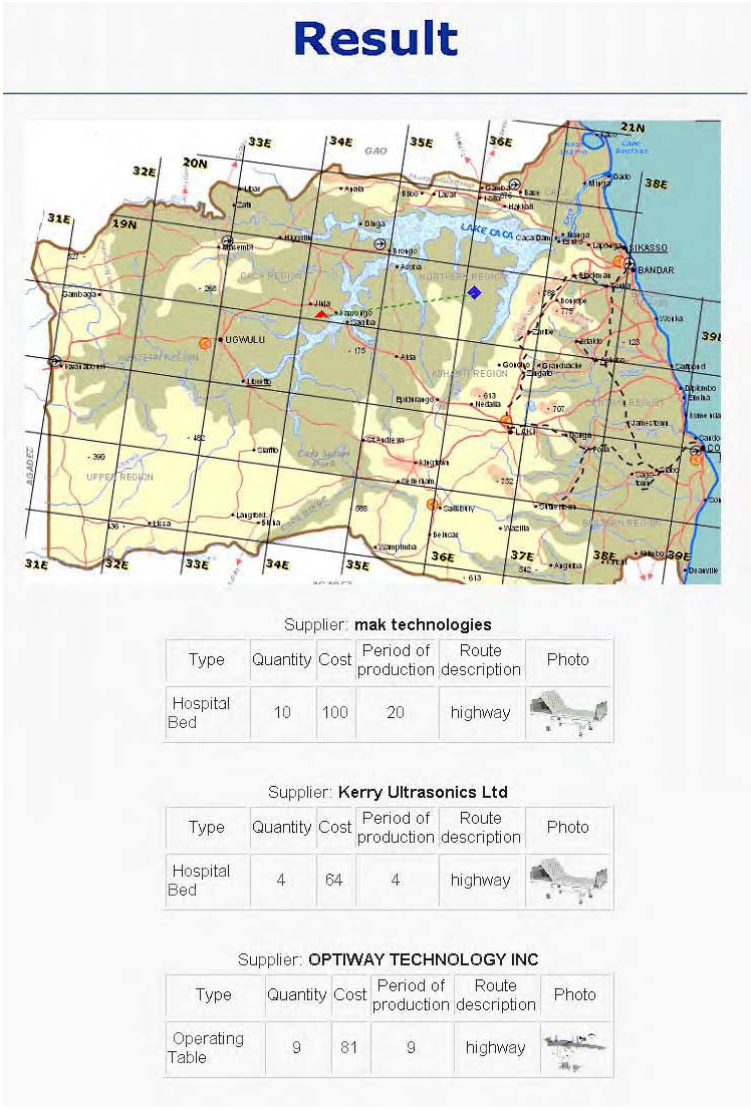



**Figure 20.** Example results of user request processing.

## 7. Conclusions

The paper discusses techniques, supporting procedures/tasks used for implementation of the knowledge fusion systems “KSNet” based on the KSNet-approach to knowledge logistics. A description of the multi-agent architecture of the “KSNet” system is given based on this approach.

**Table 3.** Comparison of the "KSNet" system with existing knowledge integration systems

| *Characteristic* | *KRAFT* | *InfoSleuth* | KSNet |
|---|---|---|---|
| languages and formats used | KQML, P/FDM, CoLan, CIF | initially KQML, KIF, currently OKBC. Initially ODBC; currently JDBC. LISP, CLISP, LDL+ Java, C/C++, NetScape | KQML, KIF, DAML+OIL MS Visual C++, ILOG, MS Access, HTML, JavaScript, PHP |
| supported sources | any available information sources for which appropriate processing mechanisms exist | initially databases; currently any available sources for which appropriate processing mechanisms exist | any available sources for which appropriate processing mechanisms exist |
| multi-agent architecture | FIPA-based with peer-to-peer interaction | FIPA-based with mediating interaction | FIPA-based with mixed peer-to-peer & mediating interaction |
| relationships between ontologies | Hierarchy | mapping of sources ontologies to the system ontology. | mapping of sources ontologies onto current application ontology. |
| peculiarities | processes data and constraints | the network of interacting agents is developed. Mechanisms of messages interchange in multi-agent systems are described | utilizes object-oriented constraint networks for knowledge representation |
| case study | virtual enterprises | Environmental Date Exchange Network (EDEN) project | E-business, virtual enterprises |

The structure and major features of the software prototype are presented. Given examples prove the applicability of the developed techniques to such areas as management, product configuration, and supply chains. Consequently, this approach could be useful for such fields as e-business, configuration management, strategic planning, etc. Utilizing ontologies and compatibility with modern standards (such as DAML+OIL) allows seamless integration of the developed approach into existing processes in the described areas. The components of the system repository enable utilizing heterogeneous KSs due to the application of a top-level ontology, provide scalability due to an expandable/renewable KB, and allow rapid knowledge search due to the application of knowledge maps and user profiles.

## Acknowledgments

The lecture is due to research carried out as a part of the ISTC partner project # 1993P funded by Air Force Research Laboratory in Rome, NY, the projects # 16.2.44 of the research program "Mathematical Modelling, and Intelligent Systems" & # 1.9 of the research program "Fundamental Basics of Information Technologies and Computer Systems" of the Russian Academy of Sciences, and the grant # 02-01-00284 of the Russian Foundation for Basic Research. Some prototypes were developed using software granted by ILOG Inc. and the Intelligent Systems Laboratory of SPIIRAS.

## References


[1] DAML – the DARPA Agent Markup Language web site. URL: www.daml.org, 2001.
[2] DAML+OIL web site. URL: http://www.daml.org/2001/03/daml+oil-index.html, 2001.
[3] Smirnov, A., Pashkin, M., Chilov, N., Levashova, T., Haritatos, F. Knowledge Source Network Configuration Approach to Knowledge Logistics. International Journal of General Systems. Taylor & Francis Group, 2003, 32 (3), pp. 251–269.
[4] Weiss, G.: Introduction. In: G.Weiss (Eds.): Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence. The MIT Press, Cambridge, Massachusetts, London, 2000. pp. 8.
[5] Franklin, S.: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. Proceedings of the Third International workshop on Agent Theories, Architectures, and Languages. Springer-Verlag, 1996.
[6] Huhns, M.N., Stephens, L.M.: Multiagent Systems and Societies of Agents. In: G.Weiss (Eds.): Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence. The MIT Press, Cambridge, Massachusetts, London, 2000, pp. 79–120.
[7] Jennings, N.R.: On Agent-based Software Engineering. Artificial Intelligence, Vol. 117, 2000, pp. 277–296.
[8] Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems & Their Applications, January–February 1999, pp. 20–26.
[9] Smirnov, A., Chandra, C.: Ontology-Based Knowledge Management for Co-Operative Supply Chain Configuration. Proceedings of the 2000 AAAI Spring Symposium "Bringing Knowledge to Business Processes." Stanford, California: AAAI Press, 2000, pp. 85–92.
[10] Rasmus, D.W.,: Knowledge Management: More Than AI But Less Without It, PC AI, Vol. **14**, 2000, No. 2, (Knowledge Technology Inc., Phoenix, AZ) pp. 35–44.
[11] Jarrar, Y., Schiuma, G., Zairi, M.: Defining organisational knowledge: A Best Practice Perspective. VI International Conference on "Quality Innovation Knowledge, Kuala Lumpur, 17–20 February, 2002, pp. 486–496.
[12] Dzbor, M., Paralic, J., Paralic, M.: Knowledge Management in a Distributed Organisation,: Proceedings of the BASYS'2000 – 4th IEEE/IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing, Kluwer Academic Publishers, London, September 2000, ISBN 0-7923-7958-6, pp. 339–348.
[13] Park, J.Y., Gennari, J.H., Musen, M.A.: Mappings for Reuse in Knowledge-based Systems. SMI Technical Report 97-0697, 1997.
[14] Ceruti, M., Anken, C., Lin, A. Rubin, S.: Applications of High-Performance Knowledge-Based Technology. Proceedings of IEEE Systems Man and Cybernetics, Vol. 3, Nashville, Tennessee, USA, October 8–11, 2000.
[15] Vail, E.F.: Knowledge Mapping: Getting Started with Knowledge Management. Information Systems Management, Fall, 1999, pp. 16–23.
[16] Microsoft Sharepoint web site. URL: http://www.microsoft.com/sharepoint/, 2002.
[17] Hummingbird web site. URL: http://www.hummingbird.com/products/eip/, 2002.
[18] IBM Lotus Discovery Server web site. URL: http://www.lotus.com/home.nsf/welcome/discoveryserver, 2002.
[19] Maedche, A. Staab, S.: Discovering conceptual relations from text. In: W.Horn (Ed.): Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000) (IOS Press, Amsterdam, 2000).
[20] Tecuci, G., Boicu, M., Marcus, D., et al.: Development and Deployment of a Disciple Agent for Center of Gravity Analysis, Proceedings of the Eighteenth National Conference on Artificial Intelligence and the Fourteenth Conference on Innovative Applications of Artificial Intelligence, Edmonton, Alberta, Canada, July 27 – August 1, 2002, pp. 853–860.
[21] Blythe, J., Kim, J., Ramachandran, S., Gil Y.: An integrated environment for knowledge acquisition. Proceedings of the International Conference on Intelligent User Interfaces, Santa Fe, New Mexico, January 2001, pp. 13–20.
[22] Trellis project web site. URL: http://www.isi.edu/expect/projects/trellis/index.html, 2002.
[23] COGITO: E-Commerce with Guaiding Agents Based on Personalised Interaction Tools. URL: http://ipsi.fhg.de/~cogito, 2002.
[24] Cheah, Y., Abidi, S.S.R.: Healthcare Knowledge Acquisition: An Ontology-Based Approach Using the Extensible Markup Language (XML). URL: http://www.geocities.com/yncheah/papers/ mie00_cheah_abidi_ont.pdf, 2002.
[25] Sure Y.: A Tool-supported Methodology for Ontology-based Knowledge Management. In: E. Stubkjaer (Ed.): The Ontology and Modeling of Real Estate Transactions in European Juristictions. International Land Management Series, Ashgate, URL: http://www.informatik.uni-bremen.de/~heiner/COST/Sure.doc, 2002.

[26] Ontoprise: Semantics for the WEB. URL: http://www.ontoprise.de/products/index_html_en, 2003.
[27] Protégé-2000 Project web site. USA, Stanford Medical Informatics at the Stanford University School of Medicine. URL: http://protege.stanford.edu/index.shtml, 2000.
[28] Ontolingua web site. Stanford University, Knowledge Systems Laboratory. URL: http://www-ksl-svc.stanford.edu:5915/&service=frame-editor, 2001.
[29] Pease, A., Chaudhri, V., Lehmann, F., Farquhar, A.: Practical Knowledge Representation and the DARPA High Performance Knowledge Bases Project, In A. Cohn, F. Giunchiglia, B. Selman (Eds.): Proceedings of the Conference on Knowledge Representation and Reasoning, KR-2000, Breckenridge, CO, USA, Morgan Kaufmann , pp. 12–15 April 2000.
[30] AKT project web site. URL: http://www.aktors.org/.
[31] Visser, P.R.S., Jones, D.M., Beer, M.D., Bench-Capon, T.J.M., Diaz, B.M., Shave, M.J.R.: Resolving Ontological Heterogeneity in the KRAFT project. Proceeding of the International Conference On Database and Expert System Applications, DEXA-99, Florence, Italy, Springer-Verlag, LNCS, 1999.
[32] Nodine, M.H., Unruh, A.: Facilitating Open Communicating in Agent Systems: the InfoSleuth Infrastructure. Technical Report MCC-INSL-056-97, Microelectronics and Computer Technology Corporation, Austin, Texas, 1997. 78759.
[33] Mena, E., Kashyap, V., Sheth, A., Illarramendi, A.: Estimating Information Loss for Multi-ontology Based Query Processing. Proceedings of the 13th Biennial European Conference on Artificial Intelligence, ECAI'98, Workshop on Intelligent Information Integration, III98, Brighton, UK, 1998.
[34] Smirnov, A., Pashkin, M., Chilov, N., Levashova, T. Multi-agent Architecture for Knowledge Fusion from Distributed Sources. Lecture Notes in Artificial Intelligence. Springer, 2002. 2296, pp. 293–302.
[35] Adams, M.B., Deutsch, O.L., Hall W.D., et al.: Closed-Loop Operation of Large-Scale Enterprise: Application of a Decomposition Approach. Proceedings of DARPA ISO Symposium on Advanced in Enterprise Control (AEC), San-Diego, USA, 2000, pp. 1–10.
[36] Howells, H., Davies, A., Macauley, B., Zancanato, R.: Large Scale Knowledge Based Systems for Airborne Decision Support. In: Knowledge-Based Systems, 12. pp. 215–222.
[37] DARPA Advanced Logistics Project web site. URL: www.darpa.mil/iso/alp, 2002.
[38] Cunningham, M.: Buyer (and Seller). Be Aware. In: *DB2 Magazine*, **6**, 2. pp. 33–37.
[39] Kim, J., Ling, R., Will, P.: Ontology Engineering for Active Catalog.: Proceedings of AAAI Workshop on Using AI in Electronic Commerce, Virtual Organizations and Enterprise Knowledge Management to Reengineer the Corporation, Providence, Rhode Island. URL: http://www.isi.edu/expect/link/-papers-ontos.html.
[40] Kurbel, K., Loutchko, I. A.: Framework for Multi-Aagent Electronic Marketplaces: Analysis and Classification of Existing Systems. Proceedings of the International NAISO Congress on Information Science Innovations, ISI'2001, Symposium on Intelligent Automated Manufacturing, IAM'2001, Dubai, U.A.E. Electronic Proceedings, 2001.
[41] Anken, C.S.: Information Understanding. Introduction for 5th Anniversary Information Workshop, Rome, NY. The Information Institute. Air Force Research Laboratory (AFRL) Information Directorate, 2002.
[42] Varshney, P.K.: Scanning the Special Issue on Data Fusion. Proceedings of the IEEE, 1997. 85. pp. 3–5.
[43] Holsapple, C.V., Singh, M.: The Knowledge Chain Model: Activities for Competitiveness. Expert System with Applications, 2001. 20, pp. 77–98.
[44] Aguirre, J.L., Brena, R., Cantu, F.J. Multiagent-based Knowledge Networks. Expert Systems with applications 2001. 20, pp. 65–75.
[45] Gray, P.M.D., Hui, K., Preece, A.D.: Finding and Moving Constraints in Cyberspace. AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace. Stanford University, California, USA. AAAI Press, 1999. pp. 121–127.
[46] Smirnov, A., Pashkin, M., Chilov, N., Levashova, T. Multi-Agent Knowledge Logistics System "KSNet": Implementation and Case Study for Coalition Operations. In: V. Maik, J. Müller, M. Pchouek (Eds.): Multi-Agent Systems and Applications. Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003, Prague, Czech Republic, June 16–18, 2003. Proceedings LNAI 2691, p. 292–303.
[47] Smirnov, A., Pashkin, M., Chilov, N. Agent Architecture and Negotiation Protocol for Knowledge Logistics in Distributed Intelligent Enterprise. In: A. Dolgui, J. Sofdek, O. Zaikin (Eds.): Production System Design, Supply Chain Management and Logistic. Proceedings of the 9th International Multi-Conference on Advanced Computer Systems (ACS'2002). Szczecin, Poland, 2002. Part 2. 379–386.
[48] Smirnov, A., Pashkin, M., Chilov, N., Levashova, T. Multi-agent support of mass customization for corporate knowledge management. Proceedings of the International Conference on Intelligent Manufacturing Systems, 2003, Budapest, Hungary, pp. 103–108.

[49] FIPA Documentation. Geneva, Switzerland, Foundation for Intelligent Physical Agents (FIPA). URL: http://www.fipa.org, 2000.
[50] Guan, S., Zhu, F., Ko, C.C. Agent Fabrication and Authorization in Agent-Based Electronic Commerce. Proceedings the International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce, MAMA'2000, Wollongong, Australia. Electronic Proceedings, 2000.
[51] Smirnov, A., Pashkin, M., Rakhmanova, I. Multi-agent WWW-based Environment for Distributed Team Decision Making Support. Proceedings of Benefit Day on Pan-European Co-operation and Technology Transfer, Budapest, Hungary, 1997, pp. 23–29.
[52] Rios, S.: Decision Theory and Decision Analysis: Trends and Challenges. Kluwer Academic Publisher, Boston Hardbound, ISBN 0-7923-9466-6 July 1994, 312 pp.
[53] Bach, E., Glaser, E., Condon, A., Tanguay, C. DNA Models and Algorithms for NP-complete Problems. Proceedings of the 11th Annual IEEE Conference on Computational Complexity, CCC'96, Philadelphia, Pennsylvania, USA, 1996. pp. 290–300.
[54] ILOG Corporate web site. URL: http://www.ilog.com, 2001.
[55] Chaudhri, V.K., Lowrance, J.D., Stickel, M.E., Thomere, J.F., Wadlinger, R.J.: Ontology Construction Toolkit. Technical Note Ontology, AI Center. Report, January 2000. SRI Project No. 1633. 85 p.
[56] MacGregor, R., Patil, R.S.: Tools for Assembling and Managing Scalable Knowledge Bases. URL: http://www.isi.edu/isd/OntoLoom/hpkb/OntoLoom.html, 1997.
[57] Akkermans, H. OBELIX Project Proposal: The EU e-Business Ontology Project. Vrije Universiteit. URL: http://www.ontoweb.org/workshop/ucecOntoWeb/obelix.ppt.
[58] Rathmell, R.A.: A Coalition Force Scenario "Binni – Gateway to the Golden Bowl of Africa. In: Proceedings on the International Workshop on Knowledge-Based Planning for Coalition Forces (ed. by A. Tate). Edinburgh, Scotland, 1999, pp. 115–125.
[59] Clement, B.J., Cox, J.S., Durfee, E.H.: Infrastructure for Coordinating Hierarchical Planning Agents. In: Proceedings of the 10th Annual IPoCSE Research Symposium, 1999.
[60] CoAX – Coalition Agents eXperiment. Coalition Research Home Page, 2002, URL: http://www.aiai.ed.ac.uk/project/coax/.
[61] Clin-Act (Clinical Activity), The ON9.3 Library of Ontologies: Ontology Group of IP-CNR (a part of the Institute of Psychology of the Italian National Research Council), December, 2000. http://saussure.irmkant.rm.cnr.it/onto/.
[62] Hpkb-Upper-Level-Kernel-Latest: Upper Cyc / HPKB IKB Ontology with links to SENSUS, Version 1.4, February 1998. Ontolingua Ontology Server. http://www-ksl-svc.stanford.edu:5915.
[63] Weather Theory. Loom ontology browser, Information sciences Institute, The University of Southern California, 1997, http://sevak.isi.edu:4676/loom/shuttle.html.
[64] North American Industry Classification System code, DAML Ontology Library, Stanford University, July 2001. http://opencyc.sourceforge.net/daml/naics.daml.
[65] The UNSPSC Code (Universal Standard Products and Services Classification Code), DAML Ontology Library, Stanford University, January 2001, http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml.
[66] WebOnto: Knowledge Media Institute (KMI), The Open University, UK, 2002, http://eldora.open.ac.uk:3000/webonto.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Multi-Agent Data and Information Fusion

## *Architecture, Methodology, Technology and Software Tool*

Vladimir GORODETSKY, Oleg KARSAEV and Vladimir SAMOILOV
*St. Petersburg Institute for Informatics and Automation of the Russsian Academy of Sciences*
*Phone: +7-812-2323570, fax: +7-812-3280685*
*E-mail: {gor, ok, samovl}@mail.iias.spb.su*
*http://space.iias.spb.su/ai/gorodetski/gorodetski.jsp*

**Abstract.** This paper introduces a present-day understanding of the data and information fusion problem and describes some aspects of the methodology, technology and software toolkit developed by the authors for the design, implementation and deployment of a class of multi-agent information fusion-related applications. The distinctive feature of the proposed technology supported by the software toolkit is that it is distributed and agent-mediated, i.e. it assumes a distributed mode of designer activity mediated by agents that perform most of the routine engineering work and support the coordination of collaborative designer activities. The above methodology, technology and software toolkit were implemented and practically used for prototyping several applications from the data and information fusion scope.

**Keywords.** Multi-agent system, information fusion, distributed data mining, distributed decision making, decision combination, agent mediated technology, multi-agent software toolkit

### 1. Introduction

The authors of [1] define Information Fusion (IF) as "the process of combining data to refine state estimates and prediction" [1]. It is assumed that these data specify either a complex system comprising a set of semi-autonomous objects operating according to a joint goal (intent) or a natural phenomenon evolving in space and time, which is not directed by any intent.

An extended description of information fusion was given by B. Dasarathy [2]: "Information Fusion … encompasses the theory, techniques and tools conceived and employed for exploiting the synergy in the information acquired from multiple sources (sensor, databases, information gathered by humans, etc.) such that the resulting decision or action is in some sense better (qualitatively or quantitatively, in terms of accuracy, robustness and etc.) than would be possible if any of these sources were used individually without such synergy exploitation."

Many important applications utilize IF technology. IF is also central to maintaining situation awareness, with the objective of providing a comprehension of "what is going on so I can figure out what to do" [3] and how to make a prediction of this comprehension in the nearest future [4].

The most commonly accepted model of IF is a JDL model proposed in [1]. It considers a five level structure of data and information processing (Fig. 1) intended to provide users with a comprehension of the situation associated with a mission and an assessment of the results produced by how certain planned actions affect it.

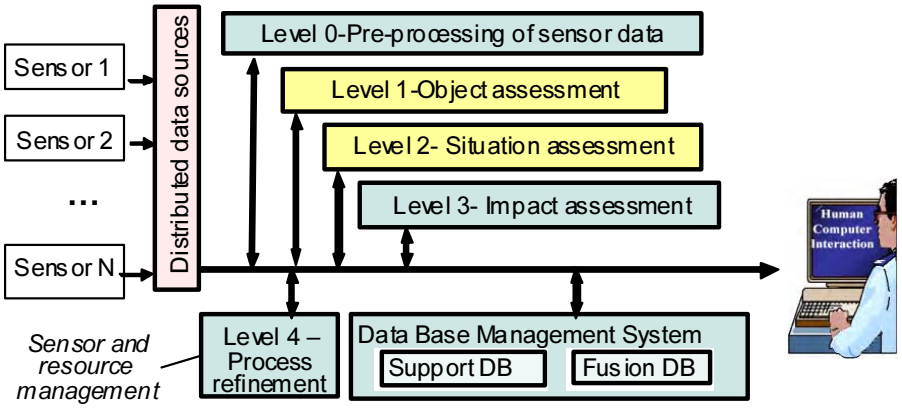



**Figure 1.** The JDL model of data and information fusion.

This paper is relevant to the tasks associated with two of five levels: "*Object assessment*" and "*Situation assessment*" (Fig. 1). Object assessment is a task of classification of the states of particular objects constituting a complex system of interest or phenomenon of the natural environment. This task is often referred to as *Data Fusion* although the boundary between data and information fusion tasks is too vague.

A given situation is specified in terms of the states and/or activities of objects constituting the situation as well as meaningful *relationships* between them. *Situation assessment* is a classification task aiming at the estimation and updating of the state of the situation. The situation is developing in time, i.e. this notion is of a dynamic nature. In many cases, situation assessment is considered a part of *Information Fusion*.

The objective of this paper is to describe basic multi-agent IF (MAS IF) system components, architecture and technology for the design, implementation and deployment of IF applications in a computer network, and also to introduce software tools supporting the above technology. Accordingly, section 2 introduces the problem statement and assumptions and describes two examples of IF applications. Section 3 motivates the use of a multi-agent architecture for IF systems intended for object and situation assessment. Section 4 presents a detailed description of the general methodology of data and information fusion design defining IF system components, their functionalities and interactions, and also presents the main algorithms used in data and information fusion. Section 5 introduces the generic multi-agent architecture of a multi-agent IF system and describes the functionalities of its basic components. Section 6 outlines the proposed technology for multi-agent IF system engineering and its supporting software tools. One of these tools, MASDK, supports the engineering of reusable components of Multi-Agent Systems for Information Fusion (MAS IF) while the second, IF Design Toolkit, supports the engineering of the problem and application-specific components of MAS IF. Section 7 describes MASDK and the IF Design Toolkit in detail. The conclusion summarizes the main ideas presented in the paper – multi-agent information fusion systems analysis, design and implementation.

## 2. Problem Statement

The primary objectives of the paper are to describe the methodology of information fusion for object and situation assessment as well as technology and software tools for MAS IF design, implementation and deployment. To expand on the exact nature of the IF applications considered in this paper, let us first examine two examples of such applications.

### *2.1. Intrusion Detection in Computer Networks*

At present, coordinated distributed attacks performed by a team of malefactors from spatially distributed hosts constitute the primary threats for computer networks and information. "Traces" of an attack are manifested in various data observed and processed by a computer network assurance system installed in different hosts of a computer network. For example (see Fig. 2, taken from [5] and slightly reworked), the traces of malefactors' attacks are displayed in a *tcpdump* file containing data resulting from input traffic preprocessing, in an audit data trail, in sequences of system calls made by the operating system, in short-term and long-term statistical data resulting from the monitoring of application servers, in queries to databases and directories, in data specifying user profiles, etc. To detect a broad variety of attacks against computer networks, including distributed attacks, it is necessary to make use of all available data and information sources [6]. Formally, intrusion detection is a classification task that utilizes a combination of alerts produced via analysis of data and information obtained by particular sources mentioned above. The assessment of the computer network security status presents a typical example of the IF problem.

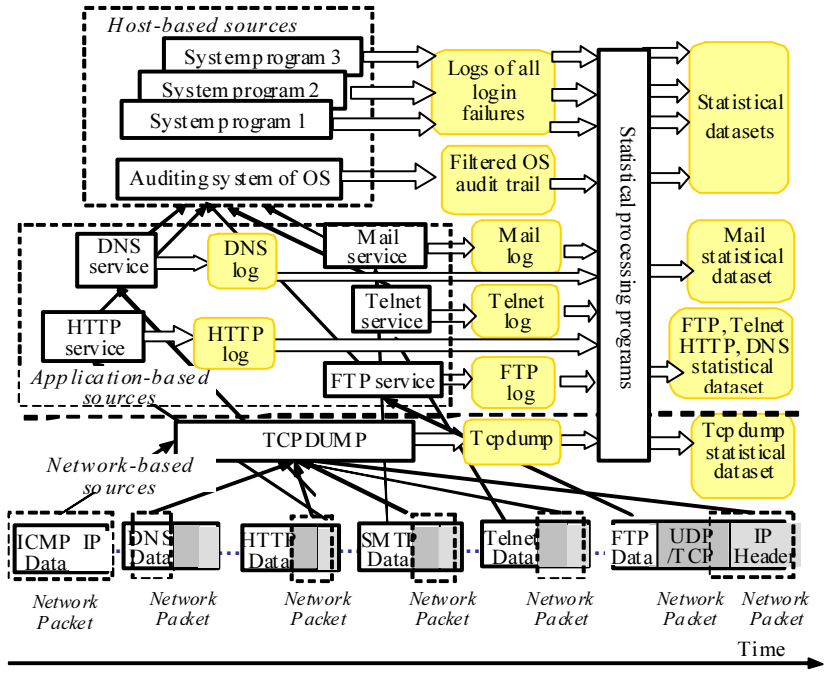



**Figure 2.** Multiplicity of data and information sources (given in grey) available for assessment of the security status of a host (the content of the figure is taken from [5]).

### 2.2. Analysis and Prognosis of Natural and Man-Made Disasters

Many different kinds of potentially dangerous situations often surface in the different regions of many countries. They can emerge due to natural disasters (earthquakes, floods, etc.), man-caused emergencies (chemical, nuclear, etc.) and so on. The specific features of such phenomena are a rapid and weakly predictable development in time and space, and a strong dependence on weather conditions, terrain, building infrastructures and so on. To assess the situation as a whole, in order to be able to predict its development and prevent undesirable or catastrophic consequences, it is necessary to make use of data and information collected from a variety of sources. A shortened example of data sources pertinent to a particular case of disaster, a flood, is presented in Figs 3 and 4. In this example, the sources of data and information and processing procedures are presented in two levels. The first level (Fig. 3) corresponds to the data and information specifying flood situation parameters in particular regions, provided by their own flood monitoring and data processing subsystems. At the second level, information characterizing situations in different regions is collected.

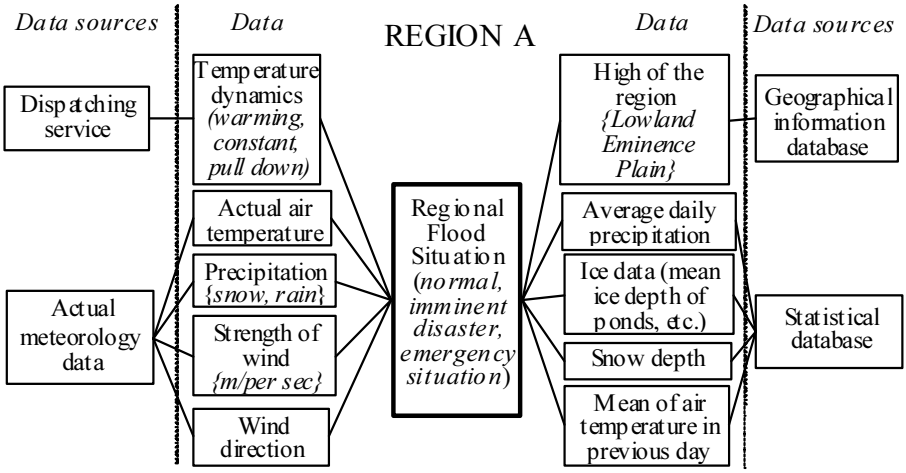



**Figure 3.** Data sources used in flood monitoring, prediction and related management.

It is processed together with additional aggregated information received, for example, from airborne equipment. Figure 4 demonstrates the second level of IF, as well as a closed loop of situation assessment, impact assessment and process refinement for flood situation monitoring and management, considered within the JDL model of information fusion (see Fig. 1).

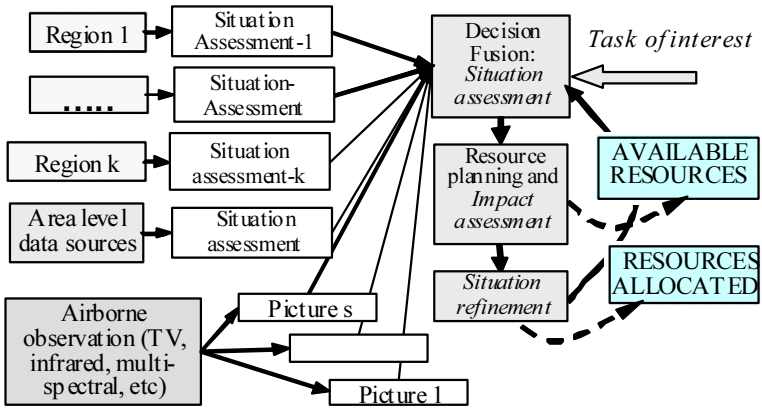



**Figure 4.** Closed loop of situation management: Situation assessment→Impact assessment→Situation refinement→…

### *2.3. Objectives and Primary Assumptions*

The objectives of this paper are to discuss the design and implementation issues of multi-agent IF systems for applications similar to those described above. The central task of an IF system is classification aiming at the assessment of the states of a complex object or situations constituted by semi-autonomous objects possibly operating according to a shared goal (intent). General assumptions are as follows. It is supposed that classification is performed on the basis of fusion of heterogeneous data and information obtained from multiple distributed sources. Data and information from different sources can be presented in different measurement scales and can be of varying accuracy. IF system input data can be incomplete, i.e. some sources may contain missing values of attributes and the total dimensionality of data can be large. It should also be assumed that some data can be unavailable for centralized processing. For example, data from particular sources can be private or classified. The set of possible states of the object of interest or situations to be assessed is known and finite. Thus, it is assumed that the IF system solves a *distributed classification task*.

An important assumption is that the design and implementation of the classification mechanism is performed on the basis of data mining and knowledge discovery technology, using interpreted historical datasets.

The task of data mining and knowledge discovery can be relegated to a separate system, the IF Learning System. The latter can also be designed as a particular component of the IF system as a whole, thus providing the latter with offline learning capabilities. Learning is based on the experience of the IF system which derives either positive or negative values from using the currently installed decision mechanism.

Other assumptions are of less importance and explained when necessary.

## 3. Why Multi-Agent Information Fusion? Why Multi-Agent Technology

The Multi-agent system (MAS) view represents an advantageous paradigm for the analysis, design and implementation of complex software systems. It proposes powerful metaphors for information system conceptualization, a range of new architectures,

techniques and technologies specifically destined for large scale distributed intelligent systems [7,8].

IF systems definitely fall into the class of potential MAS applications. Indeed, each IF application is naturally distributed: data sources are spatially distributed; data processing is performed in a distributed manner, the systems and/or users interested in the results of an IF system operation are distributed. If data coming from different sources are private or classified (military data, commercial data, etc.) then such data is not available for a centralized processing, in particular, the data holders do not render the accumulated datasets for the learning of classification. At the same time, they can make this data available for situated agents in order to process the private data locally, without revealing its content.

In most cases, IF systems are of a large scale, and it is exactly these kinds of applications that make the most of advantages provided by MAS architectures. In particular, MAS architectures are especially convenient for software implementation of such decomposable large scale applications.

A number of advantages can also be exploited if a software tool supporting the engineering and implementation of IF systems is also built as a multi-agent system. The most important advantage originates from the fact that in many cases IF systems are being developed by several spatially distributed designers. In such a case, agents of the software tool can take the roles of mediators among different designers by supporting their collaboration according to predefined protocols that provide the design process with both flexibility and automated support of the technology discipline. It is demonstrated below that most of the technological activities performed in a distributed manner can be coordinated with more ease if they are mediated by agents operating according to predefined protocols.

Recent research practices show that the popularity of multi-agent paradigms as applied to IF systems is continuously expanding and thus it is gaining a reputation as a very promising technology.

## 4. Basic Principles of the Information Fusion Methodology for Object and Situation Assessment

The IF system design process is determined by a number of basic conceptual solutions related to the following engineering aspects:

1. allocation of data and information processing functions to the data source level and to the meta-level responsible for generating source based decisions;
2. the structure of decision making and combining processes in an IF system, referred to hereinafter as *decision fusion meta-model*, or DF meta-model;
3. the structure and representation of the IF system knowledge base (KB) and ontology;
4. classes of techniques to be used for the engineering of IF system KB and respective source-based decision making mechanisms;
5. classes of techniques to be used for information fusion. Let us analyze the above aspects in order to justify the adopted solutions.

### *4.1. Allocation of Information Fusion Function*

There exist several variants of the allocation of functions to source-based and centralized levels of data and information processing proposed for IF [9]. Let us outline and evaluate them to justify the selection.

#### *4.1.1. Centralized Data and Information Processing*

This variant assumes the straightforward transmission of data from data sources into a central database for subsequent centralized data and information fusion. By default, both classification and learning are also performed in the centralized mode. This approach possesses very obvious drawbacks: (1) inefficiency in the case of the very high dimensionality of the entire data representation space; (2) very high communication overhead and data duplication; (3) inability to preserve data privacy both in learning and classification procedures. If various data structures are used by different sources then this methodology becomes practically infeasible.

#### *4.1.2. Combining Knowledge Bases of Data Sources*

Knowledge bases designed on the basis of particular data sources can be simply combined within a single KB, which afterwards is used as the KB of a centralized classification system. Although an obvious drawback of this model is that it supposes the use of centralized classification, it preserves source data privacy because while the agents have access to the private data, designers do not. This model is applicable when all local knowledge bases are represented in a common structure, e.g. they are all rule-based. An example of a relevant application is fraud credit card detection, if several banks agreed to use for this purpose a common protection system. Unfortunately, such an approach is not applicable if source data are of large dimensionality and knowledge representation in different sources differs greatly. Thus, such an approach is not the best choice, in most cases, for information fusion applications.

#### *4.1.3. Fusion of Decisions Produced by Base Classifiers*

In this model the decisions produced by local classification mechanisms (i.e. base classifiers [10]) are combined in the meta-level. This model is advantageous in many applications, in particular, if:

1. there are many data sources;
2. there are representative interpreted datasets sufficient for training and testing of both base and meta-level classifiers combining decisions produced by the base classifiers;

The advantages of this data and information fusion model are as follows:

1. it provides considerable decrease of communication overhead;
2. it is applicable when the structures of data produced by various sources are very diverse. In this case, independently of the data structures of particular sources, only decisions produced by source based classifiers are forwarded to the meta level and these decisions are represented in either binary or categorical measurement scales;
3. there exist several effective and efficient methods of combining such decisions in the upper level to obtain the combined decision;
4. it preserves source data privacy.

This model of data and information fusion outperforms the models described in subsections 4.1.1, 4.1.2 in many respects. It is used below as a component of the methodology of data and information fusion.

It should be noted that combinations of the fusion models described in subsections 4.1.1–4.1.3 can also be used in some specific cases. For example, in some cases it is preferable to forward to the upper level both the decisions produced by base classifiers and part of the data of some sources.

### *4.2. Decision Fusion Meta-Model*

The *meta-model of decision fusion* (*DF meta-model*) specifies structures of classifiers and their interaction in decision making and decision fusion. It comprises three types of hierarchically ordered particular structures: a structure of *source-based classifiers*, *a structure of decision combining process* and a structure of classification, henceforth referred to as *classification tree*. Let us consider these structures and their composition in the *DF meta-model*.

#### *4.2.1. Structure of Source Based Classifiers*

In the simple case where the dimensionality of the vector of data source attributes is small (about 20–25) and data representation structures are more or less homogeneous within the source, e.g. they are measured either in numerical or in discrete scales, then one base classifier for this source can be used (Fig. 5). In more complicated cases, if the dimensionality of the attribute vector is high enough and/or source data are heterogeneous (measured in different scales, are of different accuracies and reliabilities, have missing values of attributes, etc.), then it is reasonable to provide source data with several base classifiers. Each such classifier produces classifications using different sets of attributes and/or it has to be trained on the basis of different training and testing datasets. In this case, several base classifiers structured in this way have to be used in the single source. The decisions produced by these classifiers can be forwarded to the meta-level for combining with decisions produced by base classifiers of other sources (Fig. 6). An alternative is to combine these decisions and forward the combined decision to the upper level (Figs 7, 8). It is reasonable to use this model in the case where the number of source-based classifiers is too large. Example is given by an intelligent sensor network containing hundreds of multi-attribute sensors whose outputs are collected within a single data source.

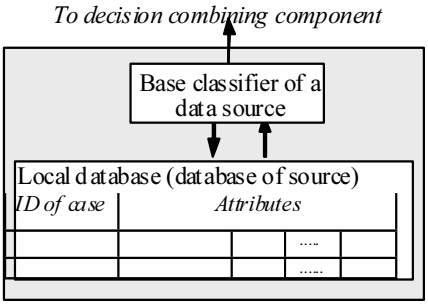



**Figure 5.** Data source decision making model: Variant 1.

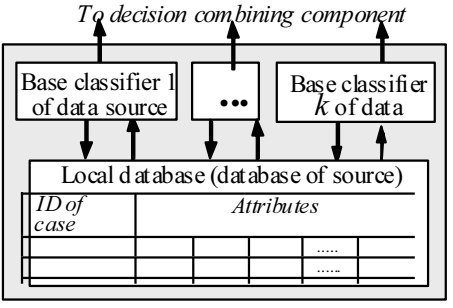



**Figure 6.** Data source decision making model: Variant 2.

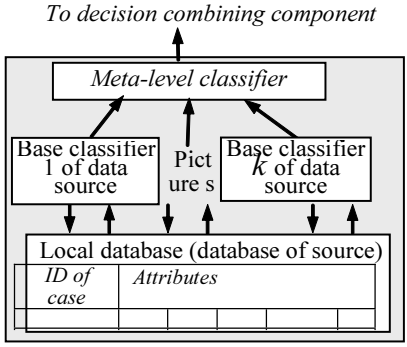



**Figure 7.** Data source decision making model: Variant 3.

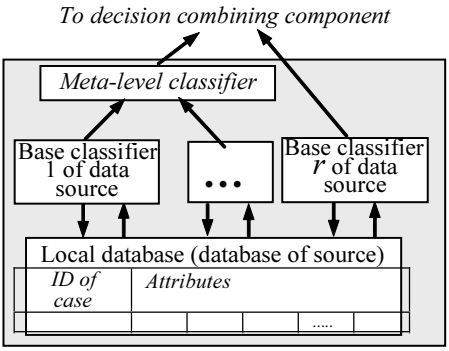



**Figure 8.** Data source decision making model: general case.

### *4.2.2. Structure of the Decision Combining Process*

The *structure of the decision combining process* specifies how decisions produced by base classifiers could be combined at the meta-level as a collective decision [11,9]. The variants of these structures are demonstrated in Fig. 5 through 8. Figures 5 and 6 present the cases where no intermediate level is used in the process of combining decisions. Fig. 7 presents an example in which decisions of base classifiers are combined prior to passing information to the meta-level. Fig. 8 presents a more general case where parts of base classifier decisions are combined, and the rest is directly relayed to the meta-level.

It is important to note that, if no uncertainty measures are assigned to the decisions of base classifiers, then all decisions relayed to the meta-level are represented either in binary or in a categorical scale. In the proposed technology, the classification task for multiple classes is reduced to a number of binary (pair wise) classification tasks. Hence the input data of a meta-classifier is represented as binary vectors. For brevity, the structure of the decision combining process is hereafter referred to as a *decision making tree.*

### *4.2.3. Structure of the Classification Tree*

The *structure of classification* ("*classification tree*") is a component of the *DF meta-model* used to reduce multiple classification tasks to a number of binary ("*pair wise*") ones. The nodes of the *classification tree* that are not leaves are called hereinafter *meta-classes*. An example of the classification tree for the task with 4 classes of situations is presented in Fig. 9, on the left. In this tree, the root node corresponds to the *meta-class* representing all possible solutions. The classification procedure in this node discriminates the situations of classes 1 and 3 from those of classes 2 and 4, i.e. discriminates instances of meta-class 1 from instances of meta-class 2. In the second step, the decision should correspond to a leaf of the classification tree indicating the final solution.

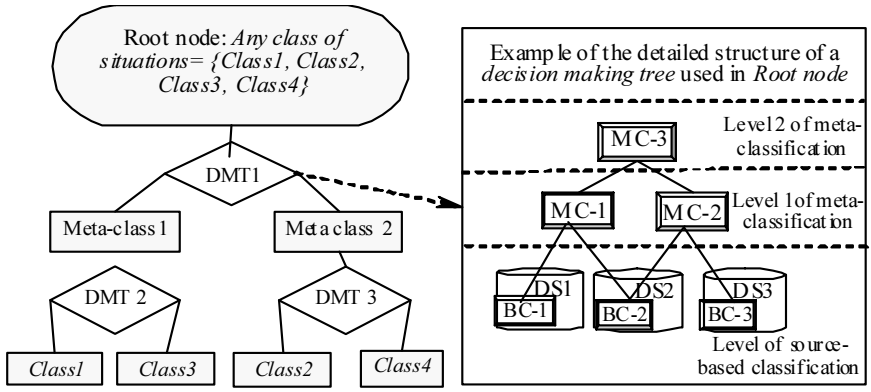



**Figure 9.** An example of the Data Fusion meta-model composed of classification and decision making trees. DMT: decision making tree; DS: data source; BC: base classifier; MC: meta-classifier.

### *4.2.4. Decision Fusion Meta Model*

On each node of the *classification tree,* a decision making task is mapped that involves in the decision making procedure all the base classifiers and meta-classifiers of the *decision making tree*. Thus, on each node of the classification tree, a *decision making tree* is mapped, which is composed of the *structure of the decision combining process* and the *structure of the classifiers* associated with data sources. Such a *decision making tree* can be different for different nodes of the *classification tree*.

An example of a *decision making tree* corresponding to the meta-class (node) *MC1* is presented in Fig. 9, on the right. This figure also illustrates the interconnection of the notions introduced in this subsection, i.e. the notions of *classification tree* (the upper level structure of data and information fusion), *decision making tree* (the lower level), *meta-class*, and also introduces the notions of *base level of classification* and *higher level meta-classification*.

Thus, the Meta-model of Decision Fusion, DF meta-model, is composed of *decision making trees* which are structured based on the *classification tree.*

### *4.3. Structure of an IF System Knowledge Base: Ontology*

The knowledge base of an IF system is composed of knowledge bases of particular classifiers, meta-knowledge base and ontology. A KB peculiarity is that it is *distributed* over hosts where the data of particular sources is stored, and particular fragments of the KB are situated in the hosts in which the combining of source-based decisions is performed. As a rule, the distributed components of the KB are *heterogeneous*.

The distributed character and heterogeneity of a KB considerably affect the methodology of its design. Additional peculiarities of KB design originate when data sources are private or classified and thus unavailable for centralized processing.

Several new problems arise from the abovementioned peculiarities. These problems have to be resolved within the design process, thus influencing the IF KB methodology [9].

The *first problem* is the necessity of providing the system with a *shared thesaurus* required for *monosemantic understanding of the terminology* used by distributed components (in our case, agents) of the IF system. This problem arises due to the fact that

specifications of data belonging to different sources can be developed independently by data holders. The latter can denote different attributes and domain notions by the same terms, and vice versa, they can denote the same notions by different terms, which obviously results in the inconsistency of knowledge representation and therefore in confusion on the part of the agents.

The problem of *non-coherency of data measurement* [9] results from the fact that different sources can contain overlapping sets of attributes and the same attributes from different sources can be measured in different units. In IF procedures the same units must be used. Thus, the problem is how to provide a KB with the capability of dealing consistently with such data.

The third problem is the "*entity instance identification problem*" [9]. Fig. 9 illustrates the essence of this problem. Indeed, due to the multiplicity of sources of information about a situation, an instance of a situation snapshot is represented by its fragments in several data sources. For example, the instance of situation #4 in Fig. 9 is represented in data sources ***A***, ***B*** and ***C***. To understand whether these fragments represent information about the same situation instance, an identification mechanism is required. In turn, a complete specification of situation instances is necessary for further training of the meta-classifier, which combines decisions produced on the basis of particular data sources.

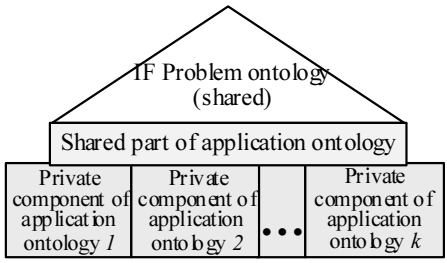



**Figure 10.** Tower of IF ontology components.

The aforementioned problems are effectively resolved on the basis of an ontology-centered distributed knowledge representation [16]. The focus of this approach is to explicitly represent all the notions (terminology) used within the system and the relationships between them as top level knowledge shared by all components of the IF system. This component of knowledge representation is referred to as a *shared ontology*. The typical structure of the ontology is explained in Fig. 10, where the upper level corresponds to the problem ontology that represents the common part of the ontology pertaining to all applications of the problem in question, in our case, the IF problem. The middle level represents a component of the ontology specifying notions and relationships particular to the application. This component of the ontology is the *shared* part of *application ontology*. This is exactly the level of ontology that must be developed to resolve the aforementioned problems (see section 7.3 describing how these problems are resolved in the proposed technology). The lowest level of the ontology, private ontology, specifies notions used only by particular source-based components of the IF system.

Thus, the structure of the IF system KB consists of ontology and distributed KBs of particular base and meta-classifiers that are structured according to the *DF meta-model* consisting of a *classification tree* (in the upper level) and the set of *decision trees* mapped to each node of the former.

## 5. Classification and Decision Combining Techniques

### *5.1. Techniques Used for Source-Based Classifier Learning*

The development of techniques for the training and testing of base classifiers is a subject of many fields of research, e.g. data mining and knowledge discovery. Let us just briefly indicate the particular classes of techniques implemented in the MAS IF Design Toolkit supporting the IF technology.

These techniques are designed to extract (1) production rules from continuous and ordered data (numerical, integer); (2) production rules from discrete data; and (3) association rules. Let us indicate particular techniques of the aforementioned classes implemented in the MAS IF Design Toolkit.

*Visual Analytical Mining* (*VAM*): for the extraction of production rules and/or decision trees from datasets containing attributes of numerical and linear ordered measurement scales [13,14]. VAM extracts production rules specified in terms of first order quantifier-free logic.

*GK2*: for the extraction of production rules from data represented in discrete scales, i.e. binary, categorical, integer and linear-ordered [15,16]. Conceptually, it is similar to the well known AQ technique [17], but uses different algorithms for the extraction of minimal rules.

The *FP-grows* algorithm of association rule mining is well recognized within the data mining and knowledge discovery community for its efficiency. Its formal and informal description can be found in [18].

The *VAM*, *GK*, and *FP-grows* techniques are implemented as classes of the *Library of training and testing methods* of the MAS IF Design Toolkit.

### *5.2. Classification Mechanisms of Base Classifiers*

The techniques described above are used for the semi-automatic engineering of the knowledge necessary for decision making by base classifiers, i.e. rule-based knowledge. Let us consider how base classifiers apply these rules for producing decisions based on newly incoming data.

It should be noted that according to the adopted IF methodology, the task of binary classification is solved in each node of the *classification tree*, and accordingly, in each node of the *decision making tree* (see section 4.2). Let us denote the classes (meta–classes) assigned to any given node of the classification tree as $Q$ and $\overline{Q}$. The production rule mining techniques *VAM* and *GK2* described in subsection 5.1 generate two sets of rules $\{R_1^+, R_2^+, ..., R_k^+\}$ and $\{R_1^-, R_2^-, ..., R_l^-\}$. The first set is such that it "argues in favor" of class $Q$, i.e. contains rules of the type $R_i^+ = F_i^+ \supset Q$, *i=1,2,...,k*, ($F_i^+$ is the rule premise represented by a conjunction of propositions), whereas the second set of rules "argues in favor" of class $\overline{Q}$, i.e. contains rules of the type $R_i^- = F_j^- \supset \overline{Q}$, *i=1,2,...,l*. ($F_i^-$ is the rule premise represented by a conjunction of propositions). Henceforth, these rules will be referred to as *arguments*. Taking into account application-dependent requirements, a metric for argument quality assessment can be introduced and computed for each of the extracted rules. Such metrics

are based on a *confusion matrix* containing the probabilities of correct classification and true and false positives.

Several decision making variants can be used if rules of base classifiers are presented in the above form. For example, the simplest variant consists in counting the weights of the "positive" and "negative" arguments in favor a particular decision, and the conclusion is made in favor of the class whose arguments are "stronger." In fact, this variant of decision making corresponds to a well-known "*weighted voting*" approach (see, e.g. [19], although this method was used decades ago). This method is nowadays widely used, and works relatively well. Other variants of the above method of argument combining are considered in the next section.

In the working version of the IF software tool, the implemented classification mechanism can be characterized as argumentation, based on notions drawn from the *Inferential Theory of Learning* [20]. This theory equates learning to knowledge mining through *knowledge space transformation.* From such a viewpoint, each hypothesis generated by an inductive learning procedure can be considered as twofold. On the one hand, such a hypothesis can be considered as a new generalized attribute specifying a new dimension of data specification, and a set of such hypotheses, in turn, can be considered as a new representation space determined by the primary set of attributes. On the other hand, a new hypothesis (for example, a rule) may represent a decision procedure intended to discriminate the situation of a category from that of another. Thus, in the latter case, the set of hypotheses is viewed as a decision structure [21].

In the implemented model of decision making, the rules (arguments) extracted through the use of *VAM* and/or *GK2* techniques are considered as new coordinates of the representation space computed via a transformation of the primary space. All of the newly computed coordinates are binary. Accordingly, the initial training and testing datasets, after they are mapped into the new space, are subsequently used for the training and testing of base classifiers. In most cases, this representation space transformation and subsequent rule mining results in the extraction of more "strong" "*pro*" and "*contra*" arguments of the class *Q*. Experience has confirmed that this process, possibly repeated more than once, results in a situation where the decision making procedure consists of computing the truth value of a single rule given over the truth values of lower level rules and/or attributes.

It should be noted that this technique is similar to the meta-classification method, which was developed concurrently as an independent project. The latter is described in the next section.

### *5.3. Techniques for Combining Decisions in Data Fusion Systems*

According to the proposed methodology, the strategy of decision combining consists of a hierarchy of multiple classifiers producing decisions on the basis of particular data sources followed by combing these decisions in the meta-level (see section 4.2 above). Let us briefly review the existing decision combining approaches and techniques and justify our preferences.

To date, several techniques of decision combining have been proposed. These techniques can be grouped as follows:

1. *voting* algorithms;
2. *probabilistic* and *fuzzy* algorithms;
3. *meta-learning* algorithms based on *stacked generalization*;
4. *meta-learning* algorithms based on *classifier competence evaluation*.

*Voting methods* have been in use for at least four decades while nevertheless preserving popularity [19]. The main drawback of these types of algorithms is that they actually have no theoretical ground and there is no guarantee that the chosen variant of voting will work perfectly in a particular application for all possible input data. The advantage is mainly in the simplicity of such techniques.

Methods of the *second group* are based on the use of different uncertainty models. Among these models, Bayesian "a posteriori" probability-based models (e.g., different kinds of Bayesian networks), possibility theory, Dempster-Shafer theory of evidence and fuzzy set-based models are the most commonly used (for details see [9]). These "classical" models have a long history and are well published. However, they are applicable in practice to low scale applications, provided that enough expert information and/or historical data are available for an empirical assessment of the respective uncertainty measures with satisfactory accuracy and reliability.

Two new groups of approaches to decision combining have recently been developed. The first is based on the "*stacked generalization*" concept. The second approach consists in the *assessment of base classifier competence*.

*Stacked generalization* [22] is very simple, but it spawned several distinct techniques of decision combining. Its most advantageous variant, "*meta-classification*," was proposed in the early 1990s [10].

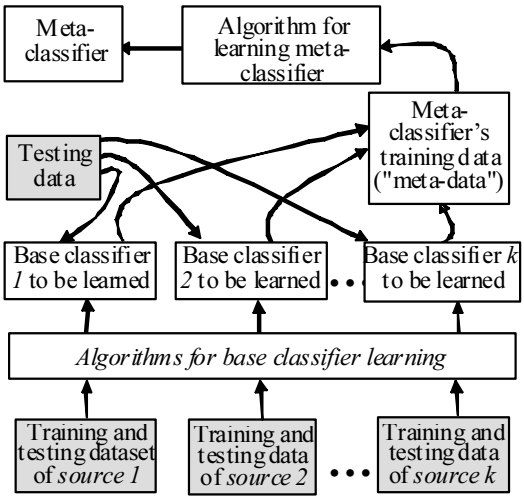



**Figure 11.** Meta-classification scheme.

A generalized structure of the decision combining process based on "*meta-classification*" is illustrated in Fig. 11. The meta-classifier is a conventional rule-based classifier, which takes as input the row of outputs from source-based classifiers. To train the meta-classifier a two-step technique is used. In the first step, the *meta-data sample* is computed based on source-based classifier testing. For this purpose, a special set of testing data is used. Let us note that a complete specification of a situation instance comprises fragments corresponding to the data of particular data sources. The main requirement to the above testing sample is that it must be complete, i.e. each instance of this sample must have no aforementioned missing fragments. The result of base classifier testing is represented as a row of classification labels produced by each

instance of the testing sample. These labels can be either correct, or incorrect. This row is augmented by the correct classification label and the resulting row in the next step is considered an instance of the *meta-data* used in the training and testing of the meta-classifier. The second step in designing the meta-classification algorithm consists in conventional training and testing performed based on the above meta-data sample.

In general, *stacked generalization*-based techniques of decision combining are effective and still under active research. A drawback these techniques is their inability to leave the pre-existing classifier unchanged if a new classifier is inserted into the classification system. In contrast, the competence-based group of techniques considered below is free of this drawback.

The competence-based group of techniques exploits the natural assumption that each classification algorithm is the most "competent" while dealing with a particular region of the representation space. Therefore *evaluations of classifier competency* for each particular record of input data is at the heart of such algorithms. This idea was first proposed in [23] and [24].

The focus of this approach is a special procedure dubbed "*referee*" (see Fig. 12) associated with each individual base classifier. This procedure assesses the competence of "own" classifier based on input data [25]. A learning procedure is used to provide the referee with this ability. Referee learning is a common learning task, which is solved on the basis of the same learning dataset that is used for the learning of its respective classifier. The only distinction is that when testing a base classifier, it is necessary to assign to each example of the testing data set a label indicating whether the classification of this particular example is true or false. As a result of testing, the examples of the testing data set are divided into two classes. The first corresponds to the class of examples classified correctly by the base classifier (area of classifier "competency") and the second corresponds to the class of examples classified incorrectly (area of classifier "incompetence"). This partition of the testing data set is further used in the training of the respective classifier referee. Referees employ certain measures for the assessment of classifier competency within particular areas.

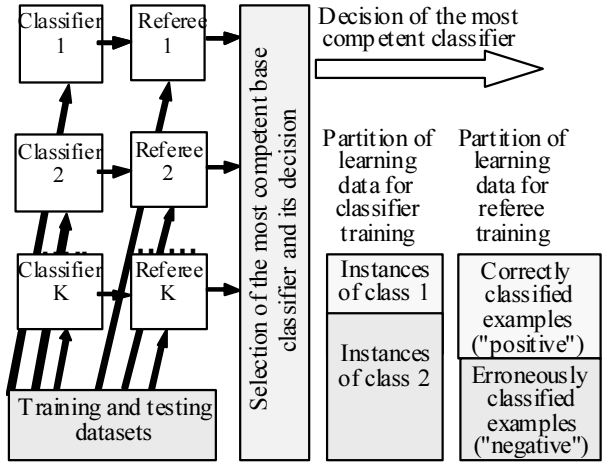



**Figure 12.** Explanation of the idea of a competence-based approach to combining decisions of multiple classifiers.

Thus, in competence-based techniques, the decision combining procedure is two-fold: (1) detection of the most competent classifier and (2) selection of the classification results produced by the most competent classifier.

See [26] for significant improvements of this method. The main advantages of the competence-based technique are higher accuracy (as compared with both voting and stacked generalization-based techniques) and the ability to preserve an already existing set of classifiers unchanged if a new classifier is inserted in the decision combining model.

Two types of methods discussed in this section, namely meta-classification and competence-based methods, are thus far used in the IF software tool as techniques for combining decisions.

## 6. Multi-Agent Architecture of the Information Fusion System

A multi-agent architecture is used for the learning and decision making components of the MAS IF considered in the paper. The learning and decision making components consist of two groups of software components (Fig. 13): (1) for handling the source-based data; and (2) for manipulating meta-data generated on the basis of source-based data. As a rule, the components of the first group are situated in the same hosts as the data source databases.

In Fig. 13, the learning components of MAS IF are referred to as *Training and Testing agents*, or *TT-agents.* Let us note that the learning components can be absent in the target MAS IF if they are not intended for further use in improving its performance.

A more detailed architecture of MAS IF is presented in Fig. 14, where the learning components (both source- and meta-levels) are depicted on the left and the components performing decision making functions (also both source- and meta-levels) are depicted on the right. Let us outline the structure and functionalities of components of the MAS IF architecture, most of which are agents of different classes having specific roles in the system.

The source-based components (Fig. 14, lower part) are common to each source. They are as follows:

*Data source managing agent*:

1. participates in the design of shared and private parts of the application ontology;
2. collaborates with meta-agents in managing training and testing datasets, participates in base classifier learning and in the processing of meta-data for meta-learning;
3. supports a gateway to its database via translation of queries from the language used in the ontology into SQL language;

*KDD data source agent*: trains and tests the base classifiers constituting the source-based classification agent and assesses the quality of their performance. Training and testing is performed using the library of learning methods, ontology, and source-based training and testing datasets.

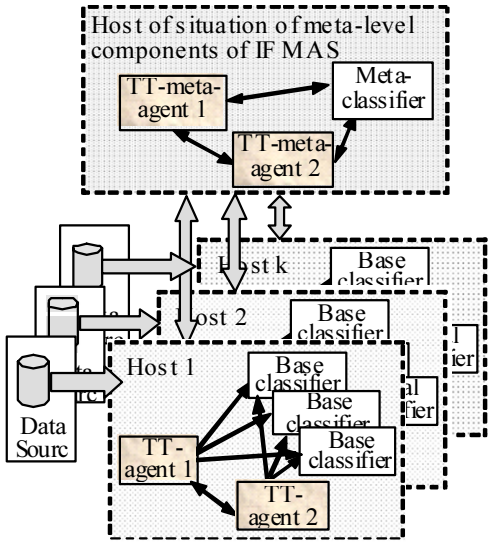



**Figure 13.** General view of the MAS IF architecture.

*Source-based classification agent*: produces decisions using source input. It is the subject of learning performed by the KDD data source agent. This agent is composed of several base classifiers structured according to the decision making model.

*Server (library) of training and testing (learning) methods*: this component is not an agent. It comprises a multitude of software classes implementing particular KDD methods, metrics, etc.

The Meta-level components of IF MAS (Fig. 14, upper) are, as follows:

*Meta-learning Agent Master ("KDD Master")*

1. manages the distributed design of the shared application ontology;
2. computes the training and testing meta-data samples;
3. manages the design of the IF meta-model (decision and classification trees).

*Meta-level KDD agent*: trains and tests the meta-level classification agent and assesses its quality.

*Meta-level agent-classifier*: performs decision combining using meta-level information. It is the subject of learning performed by the Meta-level KDD agent of MAS IF.

*Decision combining management agent*: coordinates the behavior of the meta-level agent-classifier and Meta-level KDD agent both in learning and decision combining modes.

*Communication environment*: the KQML language is used for the message content wrapper in the development of the agents' communication environment, while the content itself is specified with XML, which is used to represent the content within an application ontology. The transport level of the message wrapper uses standard TCP/IP protocol.

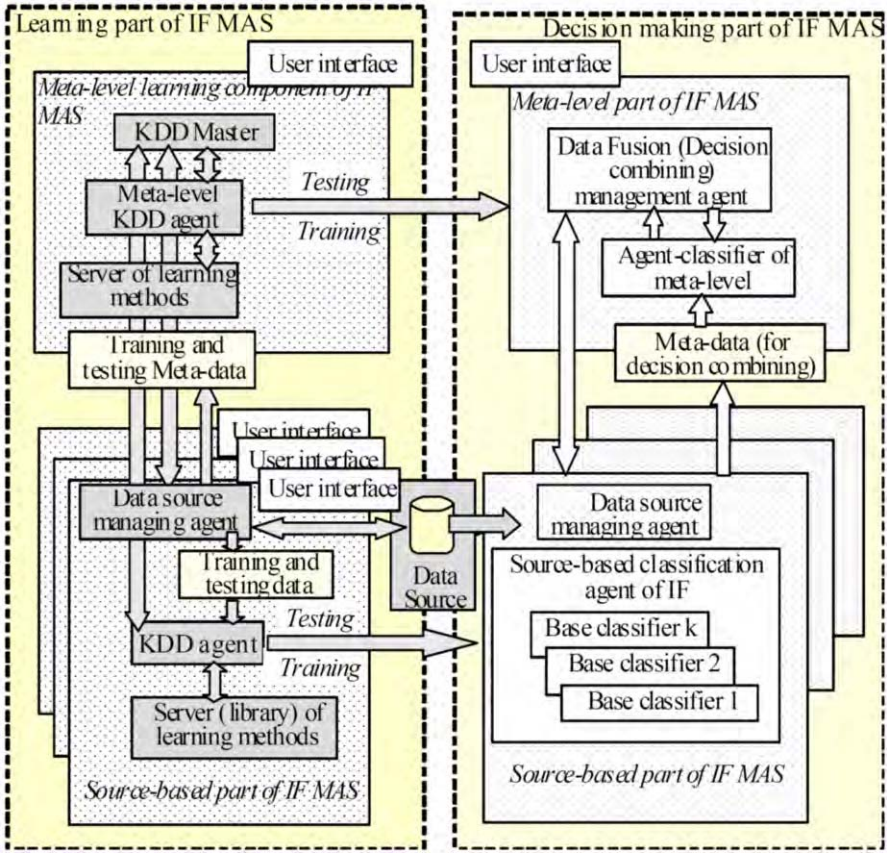



**Figure 14.** Architecture and interaction of MAS IF learning (left) and IF MAS decision making (right) components.

A conceptual view of the structure of the MAS IF agent communication environment is depicted in Fig. 15. Each communication act is supported by three intermediate components:

1. portal of the computer in which an agent sending a message is situated;
2. portal of the computer in which an agent-addressee is situated;
3. communication meta-agent (agent-facilitator) of the MAS IF supporting message addressing.

In MAS IF, these components provide complete transport services.
Protocols supporting agent message exchange are divided into three groups:

1. protocols that support agent *message exchange* in accordance with the commonly accepted three-level scheme in MAS: "message *transport* protocol" (message envelope) – "message *syntax* specification" –"message *content* specification."
2. protocols managing *semantically interconnected dialogs* (conversations) of agents that take place if agents need cooperation in task solving. These are meta-level protocols as compared to the first group of protocols.

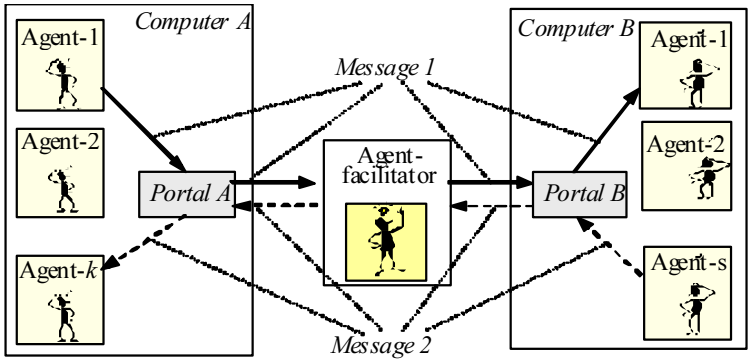



**Figure 15.** Architecture of the communication environment: message exchange routing.

3. protocols supporting agent interaction in cooperative design, learning and decision making procedures. They are considered in further detail in the subsequent section. These protocols are meta-protocols as compared to the protocols of the first and second groups.

## 7. Multi-Agent Technology for Information Fusion: Design and Implementation Issues

### *7.1. General View of the Multi-Agent Technology for IF*

The proposed MAS IF technology is based on the methodology presented in section 4. It consists of two basic phases supported by two types of software toolkits, respectively. Fig. 16 illustrates the design phase hierarchy used to create the technology.

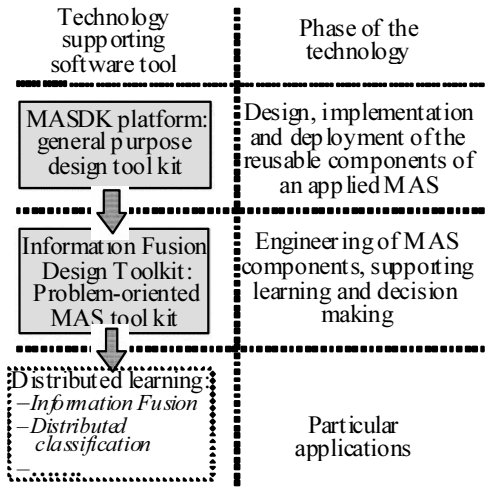



**Figure 16.** Explanation of the MAS IF engineering technology. DDM–Distributed Data Mining; DM–Decision Making.

The primary aim of the first phase[2] is the design, implementation and deployment of the *Generic MAS IF*, is part of the *application ontology, agent classes* and agent *instances* deployed in the network, and the *communication environment.* In fact, the first phase of the applied MAS IF technology was mostly intended for the engineering of its general-purpose components and also some of its problem-oriented components. In the *Generic MAS IF*, the agent classes and their instances are respectively provided with a part of IF problem-oriented functionalities and the problem ontology. This phase of MAS IF design is for the most part supported by a software toolkit referred to as the *Multi-agent System Development Kit*, or MASDK [27], a general purpose MAS software tool. The technical aspects of its practical use are outlined in this section.

The second phase is primarily intended for the *specialization* of the *Generic MAS IF*, which was developed in the first phase into an application of interest. In this phase, the technology is supported by the *Information Fusion Design Toolkit, IFDT*, developed by the authors. It is an IF problem-oriented software toolkit, which, together with MASDK provides complete support for the development and deployment of MAS IF applications.

The components of IFDT are presented in Fig. 17. They can be divided into three groups: (1) protocols supporting the collaborative engineering of the applied MAS IF and also the cooperative operation of its agents; (2) library of training and testing methods, and (3) user interfaces supporting designer activity. It should be noted that the *Information Fusion Design Toolkit* implements a new kind of agent-oriented software engineering that could reasonably be termed, *Agent-mediated software engineering*.

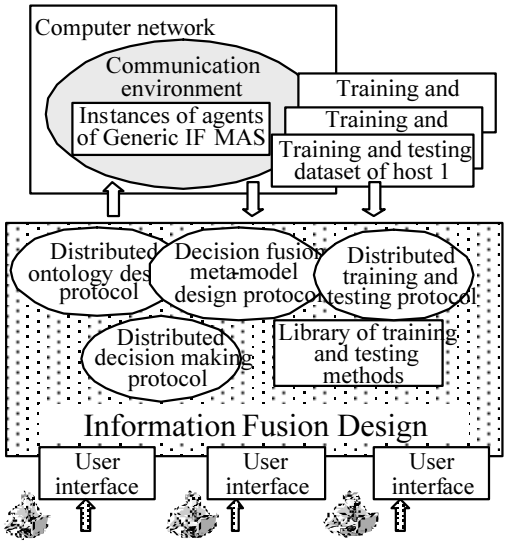



**Figure 17.** IF Design Toolkit.

As compared with existing variants of agent-oriented software engineering, the new features specific to the technology supported by IFDT are (1) the *distributed design* of an applied MAS IF that in some cases (private or classified training and testing datasets) is the only admissible one, and (2) use of *agents as mediators* of designers in engineering procedures. A high-level scenario (protocol) of applied MAS IF design supported by IFDT is presented in terms of the IDEF0 diagram in Fig. 18.

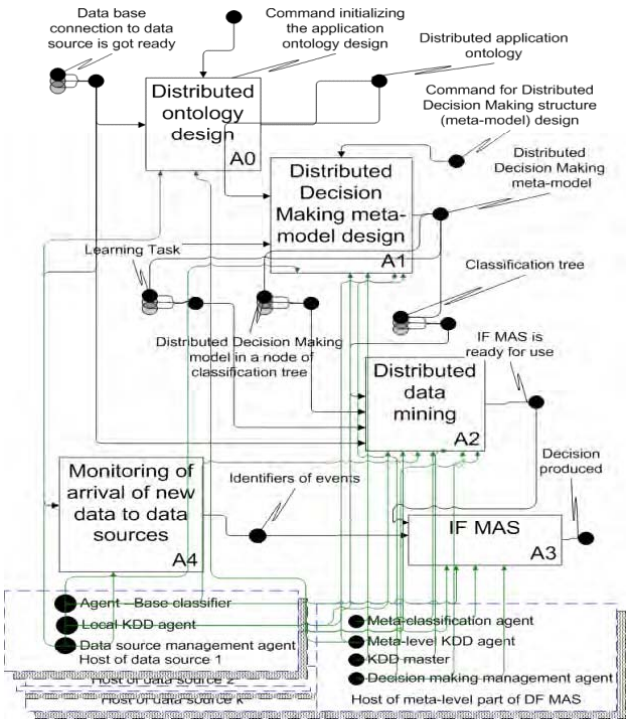



**Figure 18.** High-level protocol of the distributed design of MAS IF.

The main components of the above scenario are as follows:

- A0. Distributed engineering of shared and private parts of the application ontology;
- A1. IF meta-model design, i.e. design of decision making and classification trees;
- A2. Distributed data mining for the engineering of the IF system distributed KB used in decision making processes according to the IF meta-model;
- A3. Design MAS IF operation scenario for processing new input data, i.e. scenario of distributed decision making;
- A4. Monitoring arrival of new data to data sources.

Fig. 18 specifies the interaction of agents in design procedures, input, intermediate and final results and the order of activities processed. The core sub-protocols of the above technology are those providing MAS IF with learning capabilities, i.e. A0, A1 and A2 protocols. Sub-protocol A3 supports operations of MAS IF intended for the classification of new input data specifying a situation. The above sub-protocols are described in section 7.3.

### 7.2. MAS IF Technology Support: The Design and Implementation of Reusable Components

Until now, many multi-agent software tools and platforms (more than 70) have been developed. Among these tools, the most popular are JADE [28], Zeus [29], Bee-gent [30], MadKit [31], FIPA-OS [32], etc. However, despite the diversity of existing tools, most are still in the research stage and, in general, possess limited capabilities.

Most existing MAS software tools exploit the idea of reusability, assuming that, in different MAS, there exist many common functionalities and data structures that are, in practice, application independent [33] and thus can be used as generic classes and structures. This is also the basic idea behind the MASDK [27] used to the support the design, implementation and deployment of reusable components in the MAS IF.

MASDK consists of two parts: (1) the g*eneric agent* is considered an upper level (the most general) class of agents possessing the *reusable* classes and generic data structures of the MAS; (2) a number of individual editors for the partial specialization of the *generic agent* in the context of the intended application.

The *generic agent* is a nucleus that is bootstrapped by a designer, via user-friendly editors, into agent classes equipped with specialized data structures. In the next step, agent classes are replicated and specialized into instances of particular agents. MASDK also supports the design and implementation of the communication environment (see Fig. 15) and the deployment of MAS IF within a computer network. Further specialization of agent instances and MAS IF as a whole is supported by IFDT. Let us consider in further detail how the MAS IF technology is supported by MASDK.

As noted above, there are two types of components within each agent instance designed with MASDK: (1) the reusable components of the *generic agent*; (2) the specialization of the *generic agent*, designed using MASDK editors. Fig. 19 reveals the composition of these components. The *reusable* components of an agent class are shown on the left while the specialized (applied) components are shown on the right. It should be noted that a part of the applied components shown in Fig. 19 is further specialized using IFDT (see next subsection). Let us briefly describe this figure with particular focus on the functionalities of the particular agent components.

The meta-scenario of an agent's behavior is managed by the *Agent manager*, in effect allocating CPU time to the three main execution threads:

1. primary processing of incoming external events to the agent is performed by the *external event manager*;
2. analysis of the agent's current state and management of the agent's operation according to its state and incoming events is performed by the *agent operation manager*;
3. distribution of agent output messages, produced according to the agent's behavior scenarios and a running interaction protocol, is performed by the *output message manager.*

The *external event manager* thread identifies the external world model (thus forming the agent's beliefs). The external world model is further analyzed by the *agent operation manager*, who then allocates tasks to the *state machine manager.* In turn, the *state machine manager* initiates the proper agent behavior.

Scenarios of agent behavior are represented in terms of *state machines* [34]. A scenario selected for execution is processed by the *state machine manager*, who forms a sequence of sub-scenarios that are further allocated for execution to particular

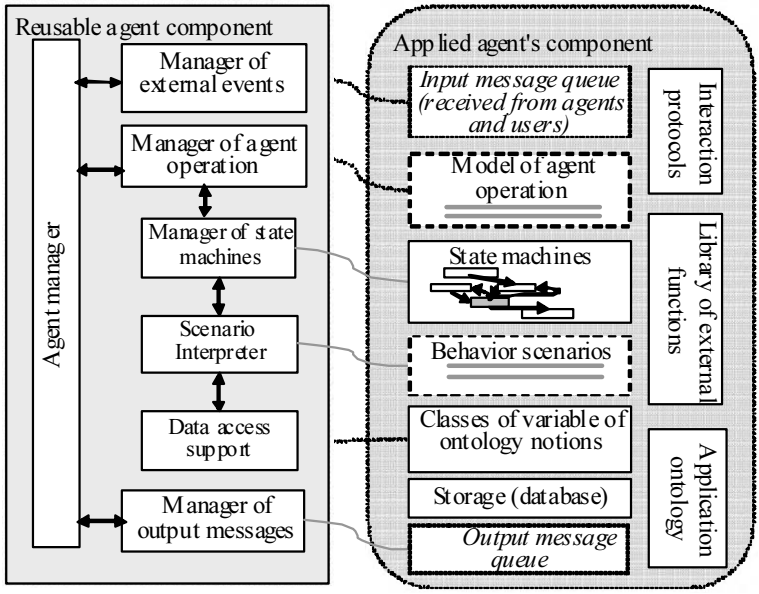



**Figure 19.** Model of reusable and applied agent's components.

*state machines*. Scenario execution is relegated to the *scenario* interpreter. Access to the agent database is concurrently provided by a *data access support* component when necessary.

*Applied agent* components are presented in Fig. 19, on the right. These components and their functions are as follows:

1. an application ontology representing the classes of notions of the application domain and the relationships that hold among them. This ontology is shared by all agents. The agents use ontology notions in two modes: (1) as a shared terminology providing unambiguous interpretation of messages that agents exchange; and (2) as a terminological basis for knowledge representation. It should be noted that agent knowledge is specified in terms of (1) *preconditions* and *transition rules* designated to every state (node) of *state machines*, (2) behavior scenarios assigned to the states (nodes) of *state machines*, and (3) behavior scenarios performed in the transition of a *state machine* from its current state (node) to a new state in accordance with transition rules;
2. *interaction protocols* specifying coordination of agent interaction and behavior in the joint execution of certain distributed algorithms;
3. *external function library* containing names of methods represented as executable code used for solving specific sub-tasks. External functions are invoked by state machines that represent agent behavior scenarios;
4. *input message queue* ordering messages in temporary storage;
5. *agent operation model* containing an upper-level specification of an agent's behavior, which is represented in terms of particular state machines. The *agent operation model* specifies function-determining rules that are executed depending on the current agent's state;

6. *state machines* specifying agent meta-scenarios, in particular, it specifies the states and transition rules to be selected for execution according to both the preconditions designated to state machine nodes and the input data;
7. *behavior scripts* specifying agent behavior according to its particular states and transitions;
8. the specialization of an agent's reusable components is supported by a number of dedicated editors equipped with user-friendly interfaces. It should be noted that design operations supported by the MASDK are interchangeable with operations supported by the IFDT (described below).

### *7.3. Information Fusion Design Toolkit*

The Information Fusion Design Toolkit (IFDT) is used together with the MASDK. It primarily supports the design of the application-specific MAS IF components and protocols for MAS IF agent interactions. The properties of the supported technology are as follows:

1. it supports a distributed mode of MAS IF design mediated by agents. In the design process, designers interact with each other according to a number of hierarchically specified protocols for monitoring and managing designer activity. The toolkit is primarily oriented toward engineering spatially distributed IF applications, with a focus on agent interaction protocols. It is also suitable for engineering MAS IF when data sources contain private data unavailable for centralized processing;
2. it provides support for sophisticated MAS IF design activities, such as maintenance of the consistency of shared and private application ontology components, IF meta-model design, learning of distributed classification and engineering of the decision fusion procedure itself.

A general description of the technology supported by IFDT is given in subsection 7.1. The collaboration of participating designers and the interaction of agents mediating designer activities are shown in Fig. 20. Designers, for the most part, initiate the design activity, acting under design protocol monitoring and management (see Fig. 18). The design protocol imposes on the designers a predefined design procedure.

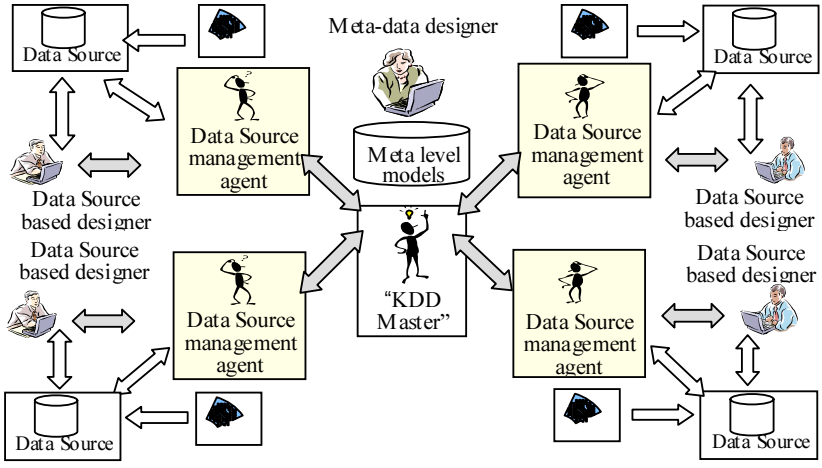



**Figure 20.** Explanation of the agent-mediated technology support provided by the IF Design Toolkit.

Let us consider the steps of the protocol given in Fig. 18 in a little more detail. The processes comprising this protocol are superficially described below, although each process is represented in several levels of detail up to the sub-processes that do not suppose distributed execution.

### *7.3.1. Design of Distributed Ontology*

Application ontology design is supported by both the MASDK and IFDT toolkits. A high level categorization of the design activities supported by the IFDT is represented in Fig. 21 in a standard IDEF0 diagram. It also indicates agents participating in the execution of particular activities. The particular design tasks to be solved by designers operating under this protocol aim at providing MAS IF application ontology with the properties outlined in subsection 4.3. The main steps of the ontology design procedure are briefly summarized below.

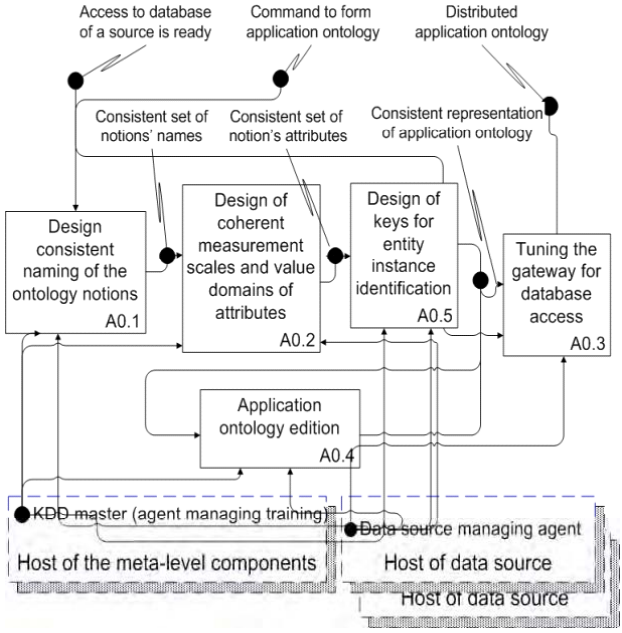



**Figure 21.** Protocol ordering ontology design activities supported by IFDT.

A0.1. Consistent naming of ontology notions: this activity is performed according to a sub-protocol (we omit its specification due to limited space), which specifies the A0.1 protocol in the next level of detail. The analysis and maintenance of the consistent naming of ontology notions is controlled by the meta-level designer, (see Fig. 20) mediated by the *KDD master agent* (see Figs. 14, 20), which in turn interacts with source-based designers via mediation of *data source management agents*.

A0.2. Design of coherent measurement scales for ontology notion attributes: the motivations for this task and its objective were outlined in subsection 4.3. Let us describe how this task is solved. Let $X$ be an attribute measured differently in different sources. In the shared application ontology, the type and the unit of attribute $X$ measurement are determined by the meta-level designer. At the next step, in all the sources

in which this attribute is found, the meta-level designer determines expressions for it so that the attribute can be converted to the same scale in all sources. Thus, values of attributes in the meta-level are used regardless of their measurement in the given data sources from which they originated. Agreement on common measurement units used for attribute *X* is reached via negotiations under a IFDT component protocol.

A0.3. Design of keys for entity instance identification: the entity instance identification task (see subsection 4.3) is solved as follows. For each entity in the application ontology, the notion of entity identifier ("ID entity") is introduced. This ID entity functions as the entity primary key (an analogy to the primary key of a database table). For each such identifier, a rule on the attributes of application ontology notions is defined. For example, in some cases, when data fragments specifying a situation snapshot (instance) are time stamped, a time interval can be used as attribute of the predicate determining the aforementioned rule. This rule is then applied to calculate the value of this key. A specific rule is defined for each particular data source so that a unique connection is specified for the entity identifier and the local primary key in this source. In a special case, it may be a list of pairs "value of entity identifier" – "value of local key." When such rules are determined for each source it is possible to form a list of all entity instances in the meta-level. This list identifies all fragments of the same instances existing in different data sources.

In current research specifications of the applied MAS IF the ontology is written in XML. In the next version of MASDK, the RDF and DAML+OIL languages [35] are considered. The use of XML, or any other pertinent language, to represent the ontology yields an additional IF specific problem. Indeed, instances of ontology notions are stored in databases accessed using standard SQL. Since databases are unable to process queries in XML or its derivatives, it is necessary to provide ontology-based systems with a special "gateway." The function of this gateway is to translate ontology language-based queries into SQL-queries. The A0.3 sub-protocol supports the design of such a gateway.

### *7.3.2. Design of a Meta-Model for Information Fusion*

According to the high level protocol (Fig. 18), the design of the IF meta-model is carried out in the second phase of the technology supported by the IFDT. Section 4.2 outlines the role and general structure of the IF meta-model in learning and decision fusion. It should be noted that at the core of the upper level of the IF meta-model is a classification tree (see Fig. 3). To each node of this tree, corresponding to either a meta-class or a class of situation, a decision tree is mapped, which can in turn branch into several levels (Fig. 9).

Fig. 22 depicts design procedures, their execution sequence and participating agents of the IF meta-model. The main steps of this phase are as follows:

1. specification of the decision making task (A1.1);
2. design of a meta-model of decision making (A1.2);
3. getting meta-properties of data source data (A1.3);
4. splitting the data into training and testing samples (A1.4);
5. design of the classification tree (A1.5);
6. specification of the learning task (A1.6);
7. forwarding the decision making structure (A1.7).

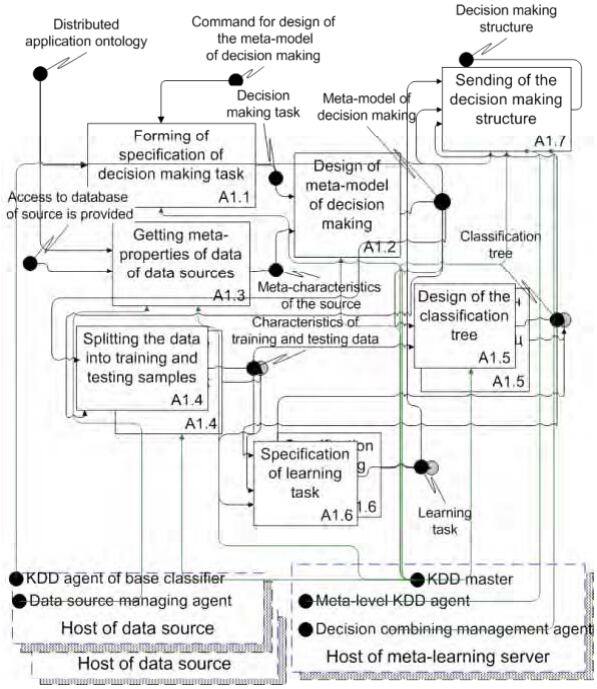



**Figure 22.** Protocol of IF meta-model design.

These processes are mediated by the *KDD master*, *meta-level KDD agent*, and *information fusion management agent*, which are meta-level agents, and also by the *data source managing agents* and *KDD agents of base classifiers* situated in the same hosts as data sources.

### *7.3.3. Distributed Data Mining Protocol*

This protocol is at the core of the IF system technology because it supports the basic functionalities of MAS IF design. These basic functionalities are the training and testing of particular classifiers and the design of decision combining procedures. An IDEF0 diagram of this protocol is presented in Fig. 23.

The distributed data mining protocol involves the interaction of all the agents of the MAS IF. Part of these agents are design mediators, the rest support the above mediators.

The basic processes of the distributed data mining protocol are:

1. selection of data for the training and testing of base classifiers (A2.1);
2. training and testing of classifiers (A2.2);
3. management of meta-classifier training (A2.3);
4. computation of data for training and testing the meta-classifier (A2.4);
5. training and testing of the meta-classifier (A2.5);
6. preparation of the IF system for use (A2.6).

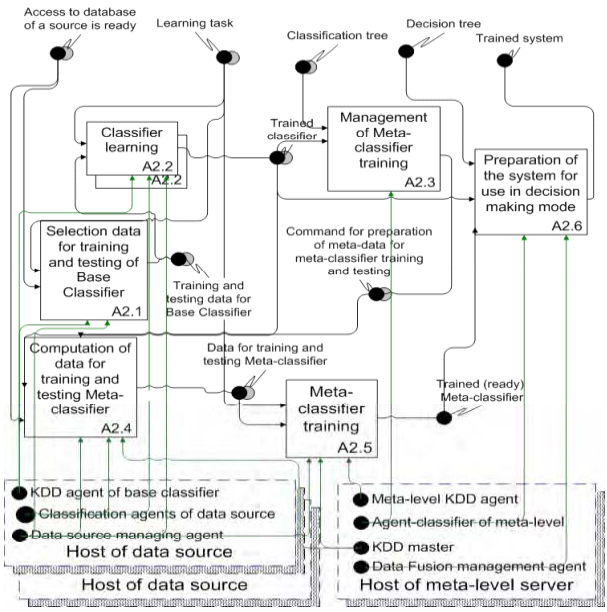



**Figure 23.** Distributed data mining protocol.

The sub-processes (sub-protocols) of distributed data mining in protocol A2 are specified in several levels of detail up to the level where processes are executed by particular agents.

### *7.3.4. Protocols for Distributed Decision Making (Decision Fusion)*

The components of this protocol (Fig. 24) are as follows:

1. analysis of the availability of new input data for processing (A3.1);
2. decision making management. This sub-protocol manages the production of decisions according to the *decision fusion meta-model* (A3.2);
3. preparation of data (A3.4);
4. decision making by base classifiers (A3.7);
5. management of meta-classifier decision making (A3.5);
6. decision making by meta-classifier (A3.6);
7. MAS IF decision making (A3.3).

It should be noted that the above sub-protocols are listed in an order corresponding to their execution in decision making procedures.

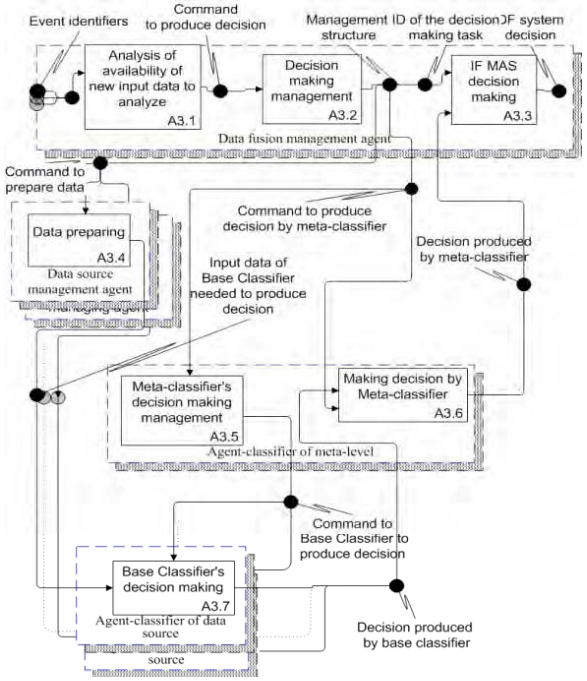



**Figure 24.** Distributed decision making protocol.

## 8. Conclusions

This paper presents a methodology of data and information fusion, a multi-agent architecture of the respective software system, the technology destined for the design, implementation and deployment of applied multi-agent data and information fusion systems, and also outlines software tools supporting the developed technology.

An application area of the above methodology and technology concerns the tasks of object and situation assessment corresponding to levels 1 and 2 of the JDF model of information fusion [36]. In turn, data and information fusion are the core tasks for providing situational awareness.

The multi-agent paradigm is used in the proposed methodology, technology and software tool. It provides designers with powerful tools for the conceptual modeling of data and information fusion problems, adequate architectures, and techniques for implementing the cooperation of distributed software components. This tool is specifically intended for modular large scale intelligent systems for data and information fusion.

It should be noted that in designing the MAS for Information Fusion, the multi-agent paradigm can be used in two distinct ways:

1. utilization of the *multi-agent architecture* for the software implementation of data and information fusion systems (MAS IF); and
2. utilization of the *agent-mediated software engineering* for MAS IF technology, in which agents function as mediators between distributed designers supporting a predefined design procedure specified formally in terms of interaction protocols, and perform numerous routine computations.

This paper covers both these aspects.

The developed methodology provides designers with a number of ready solutions concerning important methodological aspects of MAS IF design, such as:

1. how to allocate functions of data and information processing to the data source-based level and meta-level of fusion;
2. how to structure decision making and decision combining components of the IF system, i.e. how the information fusion meta-model should be organized;
3. how to structure an IF system distributed knowledge base, how to provide and maintain consistency of its distributed components, and how it interacts with the MAS IF ontology;
4. the data mining techniques used to train the MAS IF decision making and decision combining components.

The proposed technology consists in two phases: (1) the design of the MAS IF using the general purpose MASDK software tool (see section 7.2); and (2) the design of the application-specific MAS IF components and protocols for MAS IF agent interactions (see section 7.3) providing designers with a number of flexible and powerful software tools to support the engineering, implementation and deployment of applied MAS IF. In particular, engineering processes performed by a distributed team of designers support sophisticated design procedures, such as the design of information fusion meta-models. These processes also support the engineering of a distributed MAS IF knowledge base, whose upper level consists of an application ontology, and the design of the protocol for decision fusion produced by source-based distributed classifiers.

The proposed methodology, technology and software tools were validated using two well-known case studies: multi-spectral image classification (Landsat Scanner image dataset, GRSS_DFC_0010, [37]) and KDDCup99 [38].

This technology and both aforementioned software tools are currently under further development. Current efforts are focused on IF technology validation and the accumulation of experience of the MAS IF design through the development of various applications from a data and information fusion scope.

## Acknowledgements

This work is supported by the European Office of Aerospace Research and Development (EOARD, department of US AFRL/IF) and Russian Foundation for Basic Research (Project #04-01-00494a).

## Notes

1. The meta-classification approach can also be considered as a special case of the proposed approach in the “Inferential Theory of Learning” [24] mentioned in subsection 5.2.
2. The “first” and “second” phases of the applied MAS IF design described below overlap, and some design operations referred to as “first phase operations” can be carried out after some operations referred to as second phase operations.

## References


[1] A.N. Steinberg, C.L. Bowman, and F.E. White. Revision of the JDL Data Fusion Model. Proceedings of the SPIE Sensor Fusion: Architectures, Algorithms and Applications III, 1999, 430–441.

[2] International Society of Information Fusion. The Fusion Problem. *http://www. inforfusion.org/mission. htm*.

[3] E. Adam. Fighter Cockpits of the Future. Proceedings of 12-th IEEE/AIAA Conference on Digital Avionics Systems, 1993, 318–323.

[4] M.R. Endsley and D.G. Garland (Eds.). Situation awareness analysis and measurement. *Mahwah, NJ: Lawrence Erlbaum*, 2000.

[5] V. Gorodetski, I. Kotenko and O. Karsaev. Multi-agent technologies for computer network security: Attack simulation, intrusion detection and intrusion detection learning. *International Journal of Computer Systems Science & Engineering*. Vol. 18, No. 4, July 2003, pp. 191–200.

[6] T. Bass. Intrusion Detection and Multisensor Information Fusion: Creating Cyberspace Situational Awareness. *Communication of the ACM*, vol. 43 (4), 2000, 99–105.

[7] G. Weiss (Ed.) Multiagent Systems. A modern Approach to Distributed Artificial Intelligence. *MIT Press,* 1999.

[8] M. Wooldridge. An Introduction to Multi-agent Systems. *John Wiley&Sons*, 2001.

[9] Goodman, R. Mahler, and H. Nguen. Mathematics of Data Fusion. *Kluwer Academic Publishers,* 1997.

[10] A. Prodromidis, P. Chan, and S. Stolfo. Meta-Learning in Distributed Data Mining Systems: Issues and Approaches, *Advances in Distributed Data Mining*, AAAI Press, Kargupta and Chan (eds.). 1999. http://www.cs.columbia.edu/~sal/hpapers/ DDMBOOK.ps.gz.

[11] L. Rastrigin and R. Erenstein. Methods of collective pattern recognition. Moscow, *Energoizdat Publishers*, 1981. (In Russian).

[12] *http://www.semanticweb.org.*

[13] V. Gorodetski, V. Skormin, L. Popyack. Data Mining Technology for Failure Prognostics of Avionics, *IEEE Transactions on Aerospace and Electronic Systems*. Volume 38 (2), 2002, 388–403.

[14] V. Gorodetski, V. Skormin, L. Popyack, and O. Karsayev. Distributed Learning in a Data Fusion System. *Proceedings of the Conference of the World Computer Congress (WCC-2000)* "*Intelligent Information Processing*" *(IIP2000),* 2000, 147–154.

[15] V. Gorodetski, O. Karsayev. Mining of Data with Missed Values: A Lattice-based Approach. *International Workshop on the Foundation of Data Mining and Discovery in the 2002 IEEE International Conference on Data Mining*, Japan, 2002, pp. 151–156.

[16] V. Gorodetski and O. Karsayev. Algorithm of Rule Extraction from Learning Data. *Proceedings of the 8th International Conference* "*Expert Systems & Artificial Intelligence*" (EXPERSYS-96), 1996, 133–138.

[17] R.S. Michalski. A Theory and Methodology of Inductive Learning. *Machine Learning,* vol. 1, Eds. J.G. Carbonel, R.S. Michalski and T.M. Mitchel, Palo Alto, pp. 83–134, 1983.

[18] J. Han and M. Kamber. Data Mining. Concept and Techniques. *Morgan Kaufman Publishers*, 2000.

[19] T. Dietterich. Machine Learning Research: Four Current Directions. *AI magazine*. 18(4), pp. 97–136, 1997.

[20] R. Michalski. The Inferential Theory of Learning: Developing of Foundations of Multistrategy Learning. *Machine Learning: A Multistrategy Approach*, *volume 4* (Eds. R.S. Michalski and G. Tecuci), *Morgan Kaufman Publishers*, 1993.

[21] R. Michalski and A. Kaufman. Data Mining and Knowledge Discovery: A Review of Issues and Multistrategy Approach. *Machine learning and Data Mining: Methods and Applications* (Eds. R. Michalski, I. Bratko and M. Kubat) John Wiley&Sons, Ltd., 1997.

[22] D. Wolpert. Stacked generalization. *Neural Network*, 5 (2), 1992, 241–260.

[23] K. Ting. The characterization of predictive accuracy and decision combination. *Proceedings of 13th International Conference on Machine Learning*, Morgan Kaufman, 1996, 498–506.

[24] C.J. Merz. Combining classifiers using correspondence analysis. In *Advances in Neural Information Processing*, 1997.

[25] J. Ortega, M. Coppel, and S. Argamon. Arbitraining Among Competing Classifiers Using Learned Referees. *Knowledge and Information Systems*, 4, 2001, 470–490.

[26] L. Todorovski and S. Dzeroski. Combining classifiers with meta-decision trees. D.A. Zighen, J. Komarowski and J. Zitkov (Eds.) *Proceedings of 4th European Conference on Principles of Data Mining and Knowledge Discovery* (PKDD-00), France, Springer-Verlag, 2000, 54–64.

[27] V. Gorodetski, O. Karsaev, I. Kotenko. Software Development Kit for Multi-agent Systems Design and Implementation. In B. Dunin-Keplicz and E. Nawareski (Eds.) "*From Theory to Practice in Multi-agent Systems*." *LNAI*, vol. 2296, Springer-Verlag, 2002, 121–130.

[28] F. Bellifemine, A. Poggi, and G. Rimassa. JADE – A FIPA-compliant agent framework. *Proceedings of PAAM'99*, London, UK, 1999, http://sharon.cselt.it/projects/jade.
[29] J. Collis, D. Ndumu: The Zeus Agent Bilding Toolkit. ZEUS Technical Manual. *Intelligent Systems Research Group*, BT Labs. Release 1.0. 1999. http://193.113.209.147/ projects/agents/index.htm.
[30] Bee-gent Multi-Agent Framework. *Toshiba Corporation Systems and Software Research Laboratories*. 2000, *http://www2.toshiba.co.jp/beegent/index.htm*.
[31] *http://www.madkit.org.*
[32] S.J. Poslad, S.J. Buckle, and R. Hadingham: The FIPA-OS agent platform: Open Source for Open Standards. *Proceedings of PAAM 2000*, Manchester UK, 2000. http://fipa-os.sourceforge.net/.
[33] A. Sloman: What's an AI Toolkit For? *Proceedings of the AAAI-98 Workshop on Software Tools for Developing Agents*. Madison, Wisconsin, 1998.
[34] G. Booch, J. Rumbaugh, and A. Jacobson. The unified modeling language user guide. *Adison Wesley Longman*, 2000.
[35] *http://www.w3.org/DesignIssu es/Semantic.html.*
[36] J. Salerno. Information Fusion: A High-level Architecture Overview. CD *Proceedings of the Fusion-2002*, Annapolis, MD, 2002, 680–686.
[37] *http://www.dfc-grss.org.*
[38] *http://kdd.ics.uci.edu/databases/kddcup99/.*

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Architecture Analysis and Demonstration of Distributed Data Fusion

Elisa SHAHBAZIAN, Louise BARIL and Guy MICHAUD
*Lockheed Martin Canada*

**Abstract.** Information and Data Fusion is a discipline that provides methods and techniques to build Observe-Orient-Decide-Act (OODA) capabilities for various applications. There are many ways in which these methods and techniques can be chosen to provide capabilities in each phase of the OODA decision making cycle, and there are different fusion architectures, i.e., ways these methods and techniques can be applied, grouped and integrated. How one chooses the most appropriate set of methods, techniques and fusion architecture for an application depends on a number of factors. Additional factors have to be considered in the case when decision making is performed through a collaboration of a number of fusion centres on a network, defined as Distributed Data Fusion, in the context of this lecture. This lecture describes the choices for fusion architectures, the factors leading to the selection of a fusion architecture, and proposes a model to help make these choices in the case of distributed data fusion.

**Keywords.** Information fusion, data fusion, data fusion architecture, data fusion levels

## 1. Introduction

About 15 years ago, a LM Canada Research and Development (R&D) department was tasked to propose an approach to develop OODA decision support capabilities for Canada's HALIFAX Class frigates 20 years into the future. The effort started with a survey of the published literature on the technologies, techniques and algorithms used in OODA capabilities, namely Data Fusion (DF), Imaging, Artificial Intelligence, Operations Research, etc. The survey showed that there was disproportionally more research performed (and published) on understanding the theory (e.g., behaviour of algorithms, techniques and methods, models to classify the processes within the OODA loop, architectures for data flow between the processes) of these technologies, techniques and methods than their application to real systems. Of course it is not possible to build an application without understanding the theory; however, very little was found on how to choose methods, techniques and architectures for the systems of interest. The survey also found very little on the analysis of the feasibility of the methods, techniques and architectures, given a system with its sensors, missions and operational environments.

The next step in the effort was the identification of the theoretically most optimal methods, techniques and architectures for the HALIFAX Class frigate, leveraging the knowledge and experience gained from the reviewed literature. Many aspects of this optimal configuration were then modified/relaxed to take into account the design/implementation trade-offs due to the:

1. characteristics of input data;
2. computer resource constraints;
3. timing constraints;
4. availability requirements;
5. environmental and mission constraints;
6. cost versus performance trade-offs, etc.

An incremental approach was chosen where trade-offs of certain sub-set OODA capabilities were analyzed. These capabilities were then iteratively validated and augmented in simulation and real environments. Then, the incorporation of additional capabilities was analyzed.

In this incremental path of evolving these OODA capabilities the most significant qualitative and quantitative shift in the choice of methods, techniques and architectures occurred when the transition from a single HALIFAX Class frigate to a team of collaborating allied naval, airborne, land based platforms and Command and Control Systems (CCS) was considered.

This lecture uses the example of the Level 1 Data Fusion capability (Observe capability in the OODA loop) development, analysis and validation process for the HALIFAX Class frigate to demonstrate the selection and evolution of the DF architecture from a single fusion centre to a network of collaborating fusion centres.

## 2. Theoretical Background and Definitions

Before proceeding with the discussion of choices for building and evolving a DF system, a brief summary of definitions and basic theoretical background in DF is provided in this section. A detailed discussion of Data Fusion theory is provided in [1].

Data fusion is a multi-level, multi-faceted application dealing with:

1. the alignment, association, correlation, and combination of data and information from multiple sources to achieve refined position and identity estimation;
2. the complete and timely assessment of situations and impact;
3. the complete and timely assessment of the significance of situations and impact;
4. continuous refinement of processes.

Figure 1 shows the mapping of the OODA loop phases and DF levels as defined by the Joint Directors of Laboratories (JDL).

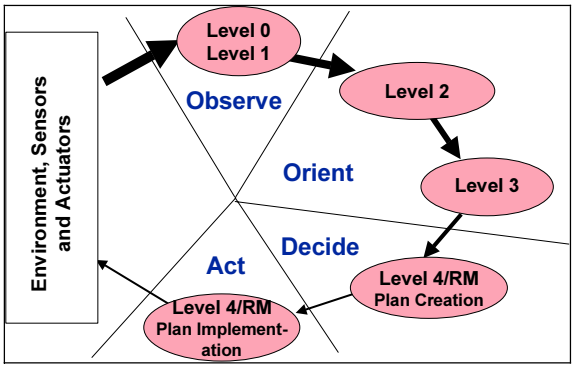



**Figure 1.** Data fusion phases and JDL levels.

The most recent definitions of the JDL DF levels are listed below [2]:

Level 0: Sub-Object Data Association and Estimation (pixel/signal level data association and characterization)
Level 1: Object Refinement (object continuous state (e.g., kinematics) estimation, discrete state (e.g., object attribute type and identity) estimation)
Level 2: Situation Refinement (object clustering and aggregation, relational analysis, communications and contextual estimation from multiple perspectives)
Level 3: Significance Estimation (situation implication, event prediction, consequence prediction, opportunities and vulnerability assessment)
Level 4: Process Refinement (adaptive processing through performance evaluation and decision/resource/mission management).

The thickness of arrows in the figure depicts the transformation of large amounts of data/information entering Levels 0 and 1 processes and continuously being refined into a higher abstraction of knowledge through higher levels of fusion processing.

Data fusion system architecture options range from a fully centralised system to a fully decentralised system.

1. a fully centralised DF system receives data from all sensor nodes and fuses it at one site, sending back a single consolidated view to all participants;
2. a fully distributed system requires the nodes to broadcast all local data available. The data would be used by each node to produce the situation view at the node.

The advantages of a fully centralised system include the ability to ensure a single consolidated tactical picture to all participating nodes and to reduce CPU resource requirements in participating nodes.

The disadvantages of a fully centralised system include lack of survivability of the nodes when network communication fails, large communication bandwidth requirement (large volume of data transferred on the network), possible computational bottlenecks in the tactical fusion centre. Another drawback is the possible requirement of a redundant or alternative tactical fusion centre to incorporate redundancy and to ensure survivability for the information distribution network. Participating nodes do not need to have any global knowledge of the sensor network topology; and should only know about connections with the tactical fusion centre and a redundant tactical fusion centre.

The main advantage of a fully distributed system is increased survivability of each participating node since it would not rely on a central fusion centre to compile a tactical picture.

The drawbacks include possible large communication bandwidth requirement since local information needs to be sent to all participating nodes and complications with fusing redundant information. If communication paths are not strictly controlled, pieces of information may begin to propagate redundantly. When these pieces of information are reused (double counted), the fused estimates produced at different nodes in the network become corrupted. Pieces of information from multiple sources cannot be combined within most filtering problems unless they are independent or have a known degree of correlation (i.e., known cross-covariance).

In a distributed system local data can be pre-processed before broadcasting, reducing communication bandwidth requirements, however this may be at the cost of the overall fusion system performance. A node in a distributed system may provide hard

decisions (binary decision, without a measure of uncertainty or confidence) or soft decisions (with uncertainty or confidence). The overall fusion performance will be higher when fusing soft decisions.

A hybrid fusion system design would address more accurately the "real world" constraints. For example, in a hybrid version of the distributed/ decentralized architecture each node could co-ordinate which local information is fused locally, which is pre-processed and which is not pre-processed before transferring, based on reporting responsibility or by implementing a hierarchical fusion system including several levels of fusion functions requiring different level and rate of information. In any application the DF architecture needs to be selected to address:

- node survivability;
- network topology;
- communication protocol;
- tactical picture commonality, consistency;
- information exchange rules and redundancy;
- track management.

The information and DF algorithms used in a fusion node (for either fusion architecture) can be "single scan" where a decision is made after fusion of each new data/information update or "multi scan" (or multi hypothesis) where decisions are made after a number (design dependent) of data/information updates have been fused, choosing the best hypotheses. Theoretically the performance of the overall fusion system will be higher when "multi-scan" approaches are used, since decisions are made based on more information.

## 3. Selection of the Fusion Architecture in a Single Node

This section discusses the selection of the fusion architecture in a single node on the example of the development of the fusion system, fusing the on-board sensor data on the HALIFAX Class frigates. The HALIFAX Class frigate was designed in 1983 and, for its time, had very modern system architecture. Its Command and Control System (CCS) was built within a distributed network of over 30 computers on a 10 megabits per second databus. The CCS decision support capabilities are integrated: information from all subsystems is maintained in one distributed Global Database (GDB) and provided to all subsystems and operators. The results of processing and decision making by the subsystems and operators are reflected in the distributed and synchronized GDB. On a high level such architecture and level of integration would be appropriate for any modern decision making system. However this very advanced system was built within a closed militarised hardware of the 80s and was limited by its data processing and communication bandwidth, as well as its growth potential.

The HALIFAX Class frigate has a sensor suite to detect and track air, surface and underwater targets, and tactical datalink systems for sharing data and commands in joint missions with other allied vessels. For the discussion of the single platform fusion solution, the fusion of the data from the datalink systems is not being addressed here. Characteristics of input data, computer resource constraints, timing constraints, availability requirements, environmental and mission constraints and cost versus performance trade-offs are discussed below for the choice of the DF architecture fusing on-board sensor data of the HALIFAX Class frigate.

Every sensor within the frigate has been designed to have a pre-processing subsystem, which provides tracks to the CCS. The Data Fusion architecture fusing these tracks would be similar to distributed data fusion system architecture. However these pre-processing subsystems are not designed to feed a DF system, since they do not provide covariances of the track estimates they provide. Thus the DF system will have difficulty dealing with redundancy (cross-covariance) of the tracks being fused. Therefore the DF architecture should be designed to fuse hard decisions from the sensor subsystems, using some sort of phenomenological approach for estimating the cross-covariances.

In terms of a choice of fusion algorithms, although muti scan approaches are expected to have higher performance, they would require considerably higher processor power, they would take longer to bring to a mature state and they would cost considerably more. Considering that the sensors perform tracking, it is anticipated that much of the background noise has been removed by the tracking software, hence there would not be significant number of false hypotheses that a multi scan approach would have to resolve, therefore a single scan approach is expected to perform sufficiently well. Therefore to ensure timely availability of a fusion system to be demonstrated onboard the HALIFAX Class frigate within cost constraints of the project a single scan approach has been selected. However, a modular software architecture which would permit easy upgrade to a more sophisticated multi scan approach has been ensured, to be able to enhance the DF system into the future, if any new sensor or existing sensor modification makes un-processed data available to be fused.

Considering roles that Canada's frigates are likely to play in the world, environmental and mission constraints trade-offs dictate emphasis on the object identification capability in the DF system. Table 1 summarizes the selection of choices for each design trade-off.

**Table 1.** Selection of data fusion choices for each design trade-off

| trade-off | choice to meet current requirements |
|---|---|
| characteristics of input data | — use single-scan algorithm (JVC-IMM [3])<br>— phenomenological cross-correlation resolution<br>— use Truncated Dempster- Shafer [4], select ID attributes |
| computer resource constraints | — no issue for single-scan<br>— design for future growth |
| timing constraints | — no issue for single scan<br>— design for future growth |
| availability requirements | — chose a low-risk available solution that can evolve |
| environmental and mission constraints | — emphasize object identification capability (Truncated Dempster-Shafer) |
| cost versus performance trade-offs | — aim for sufficient (not the best) performance<br>— enhance incrementally |

As one can see in this table, for Level 1 Data Fusion for the Halifax Class frigate a single scan high density target associator (Jonker-Volgenant-Castanon (JVC)) and a tracker capable of reacting to target manoeuvres (Interactive Multiple Model (IMM)) has been implemented for target state estimation. For ID estimation the Truncated Dempster-Shafer with selected and derived attributes available to the DF system that most characterise targets that truncates and resets ignorance has been chosen. Both of

these approaches have been demonstrated to have very good performance [3,4], however the overall system architecture is designed to accommodate evolution into a multi-scan solution, if judged necessary.

## 4. Multi-Platform Architecture Analysis

The single platform DF architecture was built using the analogy of a distributed DF architecture, however only addressing the issues of fusion of incoming information and none of the issues regarding the information sharing with other DF nodes needed to be addressed. When analysing the architecture for the collaboration of multiple such platforms it is necessary to address both how the information is shared between these platforms as well as how this information should be fused considering how the fused information is used in these platforms. As mentioned above there are specific requirements that the fusion architecture design in a specific node should be able to address:

- node survivability (how important is the node survivability for the overall mission success);
- network topology (hierarchy, where does the node fit in the joint operations context in terms of operational/tactical picture hierarchy);
- communication protocol (volume of data transfer, rate of data transfer, push-pull mechanism, broadcast, point-to-point communication);
- tactical picture commonality, consistency (how real-time and at what level of detail should the tactical picture be);
- information exchange rules and redundancy (how fusion algorithms handle data incest);
- track management (mechanisms used to maintain a unique track tag/number and keeping it up-to-date in all nodes of the network).

The network topology will have the most impact on how complex the DF architecture will have to be. Figure 2 shows an example of a symmetric network topology, where each node has exclusive control over its sensors and broadcasts all its fusion estimates to all other nodes.

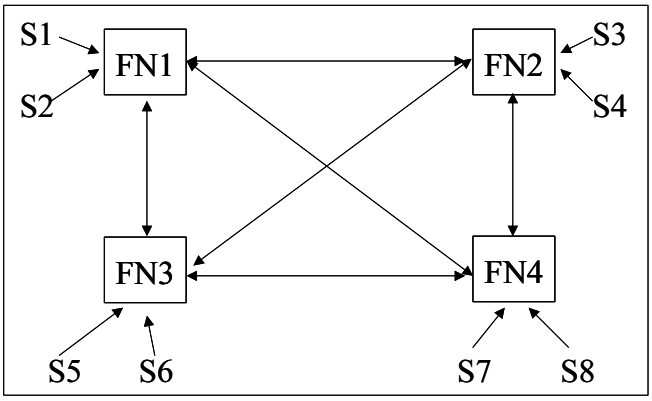



**Figure 2.** Symmetric node topology.

The data Fusion architecture in each node (FNx are the fusion nodes and Sx are the sensors) of such a topology can be identical, assuming all nodes have a consistent track management protocol. The specifics of that architecture will depend on all of the same constraints that were discussed for choosing an architecture for the HALIFAX Class frigates, except an approach for handling data incest, when fusing the information from the other nodes will be required.

It is much harder to design the DF architecture in each node in the case of a generic node topology shown in Figure 3.

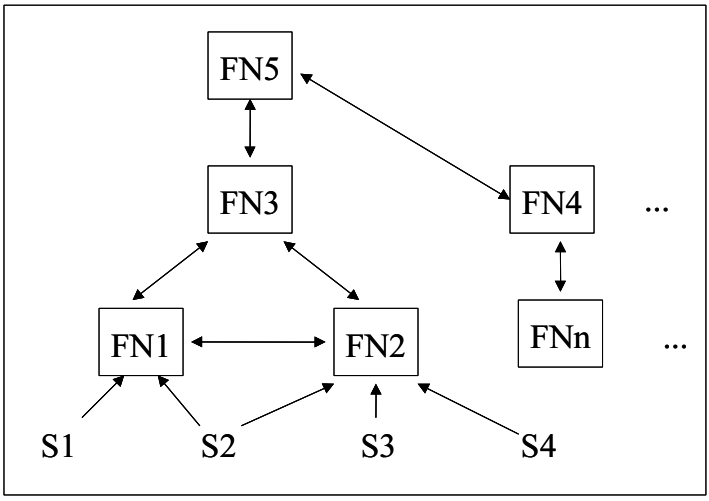



**Figure 3.** Generic node topology.

It is clear that the different nodes in this topology may have different architectures, furthermore there are many questions that need to be answered when designing the architecture for the various nodes in this figure, including:

1. what is tactical picture commonality for nodes FN1 and FN5 or FN4?
2. how does FN3 address data incest?
3. how do FN1 and FN2 address data incest?
4. how up-to-date and accurate the information should be in FN1 or FN2, versus FN3, FN5 or FN4?

The above mentioned requirements as well as the various trade-offs discussed on the DF architecture for a single node will guide the designer of the DF architectures in each node, however the answers to the above questions require significant research effort.

Additional considerations arise when looking at the evolution of warfare philosophy into the future. Military operations are moving from platform-centric warfare to Network-Centric Warfare (NCW). Traditionally, platforms own sensors and weapons, which own sensors. In NCW, sensors may not necessarily belong to the platforms or shooters. The greatest distinction between platform centric warfare and network centric warfare resides in the linkage between sensors, shooters, and decision makers. Platform centric warfare tightly links all three logically and physically, while network centric may separate these assets and then link them in different ways. NCW derives its power

from the strong networking of well-informed but geographically dispersed forces. In NCW, the enabling elements are:

- a high-performance information grid;
- access to all appropriate information sources;
- weapons reach and manoeuvring with precision and speed of response;
- value-added command and control processes to include high speed automated assignment of resources; and
- integrated sensor grids closely coupled in time to shooters and Command and Control (C2) processes.

NCW is applicable to all levels of warfare and contributes to the coalescence of strategy, operations, and tactics. It is transparent to mission, force size and composition, and geography. The net result is increased combat power [5].

Transitioning from platform centric operations to NCW will impose new requirements on the DF architecture within each node. Although the NCW vision may not be realised in the near future, the multi-platform DF system architecture must be able to address or at least be able to evolve to address the requirements of NCW.

To be able to analyse various DF architecture solutions, when transitioning from a single platform to multiple platform operations, for various node topologies and for transitioning into NCW, LM Canada R&D department has developed a multi-platform DF model and a Technology Demonstrator (TD), where different topologies and warfare philosophies can be experimented with and analysed.

## 5. Multi-Platform Data Fusion Model

LM Canada's multi-platform DF model is shown in Figure 4. It is currently anticipated that the DF architecture in each node in any network topology can be modelled consistently with the DF model. The underlying infrastructure for the implementation of this model in the TD is the distributed blackboard-based knowledge based system architecture (Cortex), developed at LM Canada [6], sponsored by Defence Research and Development Canada. The choice of the infrastructure is not pertinent to the discussion of the model, however it facilitates experimentation, as is explained later in this lecture.

This figure shows a set of processes and communication paths between these processes that can exist within a fusion node on a network. Depending on the network topology, the role of a node in the mission, and the characteristics of the input sources of information for that node, certain processes and certain communications paths may not be required and more than one layer of processing (multiple blackboards) in a specific process may be required. When building the DF architecture it is necessary to examine the requirements for each process and communication path in this figure.

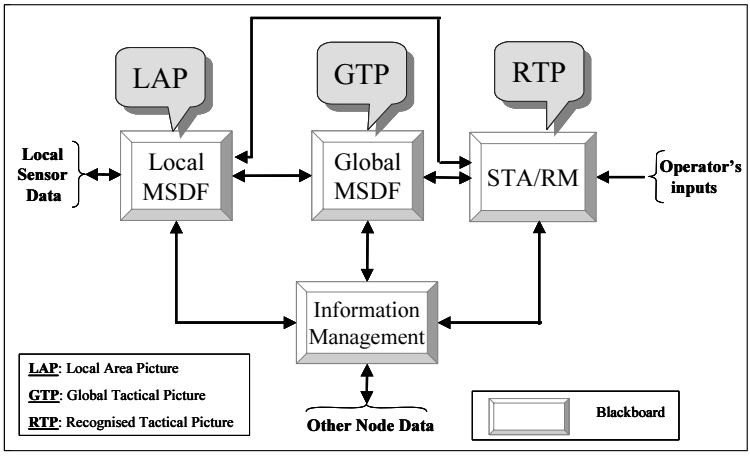



**Figure 4.** Multi-platform data fusion model.

A high level description of the processes and communication links is given below.

### *5.1. Processes*

There are 4 processes:

1. local MSDF – Level 1 Fusion engine which fuses primary (only I/O to that node) data/information – and generates the Local Area Picture (LAP);
2. global MSDF – Level 1 Fusion engine which fuses data/information from external (non primary, could be other fusion centres) sources with the LAP and generates the Global Tactical Picture (GTP);
3. STA/RM – Level 2, 3, 4, and resource management (RM) processes that provide decision support for:

    a) the interpretation of LAP and GTP – Level 2 and 3 DF processing,
    b) the MSDF process performance and refinement – Level 4 DF processing,
    c) Sensor Management– Level 4 DF processing,
    d) Weapons Management – RM processing,
    The modifications of GTP based on STA/RM reasoning as well as operator refinements of the GTP are maintained in the Recognized Tactical Picture (RTP);

4. Information Management (IM) – A process that manages the information flow between the Fusion Node and the network.

STA/RM internal process interfaces between levels 2, 3, 4 and RM are not detailed here, as this should be a topic of separate research. IM represents both node internal information management processes as well as inter-node communication management processes that are specific to the node.

In the TD there are also processes called Communications Manager (CM) and Network Manager (NM), which simulate the communications protocol and topology aspects of the inter-node communication.

### *5.2. Communication Links*

Local MSDF's LAP estimates are made available to:

- global MSDF to fuse with data/information form other nodes forming the GTP;
- STA/RM perform STA/RM functions for evaluation of Local MSDF performance;
- IM to provide the node LAP and/or sensor raw data to other nodes in the network.
- Local MSDF may receive input from:
- global MSDF (GTP) for algorithm and sensor cueing;
- STA/RM as refinements to the LAP estimates or as sensor management or process refinement recommendations;
- IM as contact-level data from other nodes or sensor management or process management requests from other nodes.

STA/RM recommendations are sent to Local MSDF to perform a number of actions to enhance Local MSDF performance including:

- select an alternate association mechanism for a subset of observed targets;
- select an alternate filtering approach for a subset of observed targets;
- modify MSDF parameters for processing a subset of targets;
- select different association, filtering, or parametric modifications for data/information coming form a specific source;
- recommend a Sensor Management action, e.g. provide sensor with target information to support its processing or request information of specific type, location, etc.

Global MSDF's GTP estimates are made available to:

- local MSDF for algorithm and sensor cueing;
- STA/RM to perform STA/RM functions and for evaluation of Global MSDF performance;
- IM to provide the node GTP to other nodes in the network.
- Global MSDF may receive input from:
- local MSDF to be fused with data/information from other nodes;
- STA/RM as refinements to the GTP estimates or as process refinement recommendations;
- IM as track data from other nodes or process management requests from other nodes.

STA/RM recommendations are sent to Global MSDF to also enhance the Global MSDF performance in the same way as for Local MSDF:

- select an alternate association mechanism for a subset of observed targets;
- select an alternate filtering approach for a subset of observed targets;
- modify MSDF parameters for processing a subset of targets;
- select different association, filtering, or parametric modifications for data/information coming form a specific source.

STA/RM sends recommendations for processes and information management (IM) in external fusion nodes STA/RM via IM.

IM may also provide Global MSDF with recommendations from other fusion nodes about fusion processing performance (e.g. track quality issues, track number or ID conflicts can lead to changes in MSDF algorithm or parameter modifications.

These are an initial set of possible information exchanges internal and external to the node in the context of DF processes. Depending on the network topology and the node role in the mission, only a sub-set of these interchanges will be pertinent.

There is a whole aspect of the inter-node communication that has not been detailed here. This is the information exchange that supports the task force operation based on a pre-defined communication protocol in a pre-defined topology, handled by the CM and NM processes, discussed further in the next section.

## 6. Technology Demonstrator Overview

The LM Canada TD is a versatile tool used to conduct investigations on DF architecture, algorithms and to perform data analysis. As shown in Figure 5, the TD consists of an Experimental Frame (EF) and DF applications.

The EF includes:

1. simulation/stimulation capability that permits to exercise the DF applications under test with simulated data from the CSAE-ATTI simulator developed at DRDC, or from trial data captured in at-sea trials of the HALIFAX Class frigates;
2. the Run Time Display provides the capability to display the decisions made by the DF applications under test;
3. System Manager provides capability to configure the tests being performed (e.g. selecting the scenario, number of nodes, etc.);
4. Comm Mgr/Net Mgr (CM/NM) simulate the pre-defined communication protocol in a pre-defined topology.

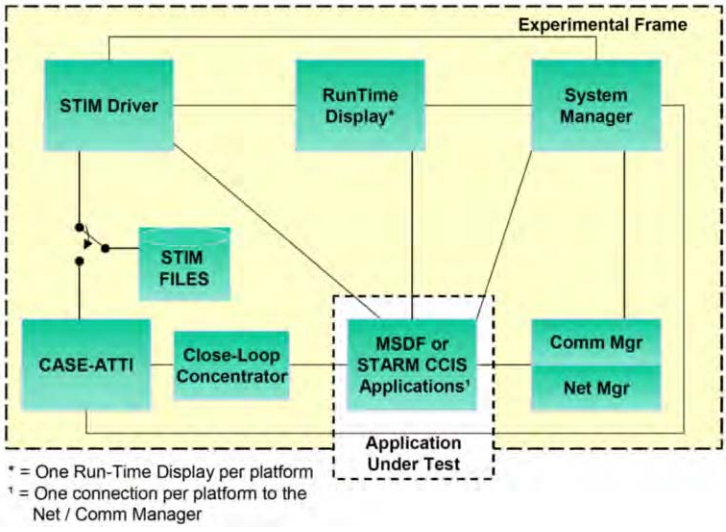



**Figure 5.** Technology demonstrator for multi-platform DF.

The DF applications under test include:

1. Multi-Source DF (MSDF) (Level 1 Fusion) from distributed fusion nodes; and
2. Situation and Threat Assessment (Level 2, 3, 4 Fusion) and Resource Management (STA/RM).

Currently the design of the DF applications under test is performed consistently with the DF model discussed above. The processes in each node are implemented as blackboards (or set of blackboards). The blackboard-based architecture was chosen to support the rule-based, concurrent and ad hoc reasoning requirements of higher level fusion (STA/RM). The distributed and modular nature of Cortex facilitates experimentation with various data fusion architectures and algorithms in each node, with various multi-platform topologies in a task force and with the evolution to NCW. Cortex also facilitates the modification and evolution of the DF capabilities and node topologies in the TD incrementally and concurrently.

Currently the application under investigation is a network of HALIFAX Class frigates and Airborne collaborating platforms such as the Aurora fixed wing aircraft and a maritime helicopter, operating in a symmetric network topology. As stated above, the requirement on the data fusion architecture in this topology is relatively easy to satisfy. Experimentation in the TD with simulated and trial data have demonstrated good performance for the currently implemented Level 1 DF capabilities. Incrementally, new capabilities are being validated and the existing capabilities are being matured [7]. It is true that the TD currently demonstrates the operation of a taskgroup collaborating to produce a common recognised tactical picture, this is far from being close to NCW requirements.

This TD is being used and will be used in the future to incrementally experiment with more complex network topologies, hoping to find answers to many questions in DF that are considered hard to answer. For Example, over the last 2 years, research is on-going to understand the requirements of the DF architectures and algorithms in a hierarchical topology necessary for the Army operations. This research could help validate and possibly modify the DF model and could become the next step toward understanding how the requirements for NCW should be satisfied.

## 7. Conclusions

The choices for DF architectures, the factors that lead to the selection of a fusion architecture in a single node and in a network of collaborating nodes have been discussed. A model to help make these choices in the case of distributed data fusion have been proposed and a Technology Demonstrator (TD) environment for the incremental development, analysis and demonstration of DF architectures and algorithms in various network topologies has been described. Currently the DF community is facing many unsolved problems when looking at the DF requirements for a task force involved in NCW. It is anticipated that the path to evolving the current maturity of DF (all levels) to a network of DF nodes of generic network topology involved in NCW will be very difficult. It is suggested that the DF model proposed here and the TD where the different problems can be incrementally implemented and analysed are necessary tools in this path.

## Acknowledgements

The authors would like to thank the LM Canada R&D department staff that participated in the research in DF and the development of the TD and the Defense Research and Development Canada (DRDC) researchers for contractual work leading to a significant portion of this research.

## References

[1] Edward Waltz and James Llinas, Multisensor Data Fusion, Artech House, 1990.
[2] Steinberg, A., Bowman, C. and White, F., Revisions to the JDL DF Model, SPIE vol. 3719, pp. 430–441, 1999.
[3] Jouan, B. Jarry, and H. Michalska, "Tracking Closely Maneuvering Targets in Clutter with an IMM-JVC Algorithm," Fusion 2000, Paris, France, July 10–14, 2000.
[4] P. Valin, and D. Boily, "Truncated Dempster-Shafer optimization and benchmarking," SPIE Aerosense Conf. on Sensor Fusion: Arch., Algorithms and Applications IV, Orlando, Florida, April 24–28 2000, pp. 237–246.
[5] Admiral Arthur K. Cebrowski and John J. Garstka, Network Centric Warfare: Its Origin and Future, (US Naval Institute Proceedings, Vol. 124/1/1,139, Jan 98), 32.
[6] Bergeron, P., Couture J., Duquet, J.-R., Macieszczak, M., and Mayrand, M., A NewKnowledge-Based System for the Study of Situation and Threat Assessment in the Context of Naval Warfare in FUSION 98, Las Vegas, 6–9 July 1998, Vol. II, pp. 926–933.
[7] Ménard E., Turgeon D., Michaud G., Shahbazian E., Comparisons of Track-Level Fusion Results with Tracklets Application within a Simulation Environment, Proceedings of NATO Advanced Study Institute, Armenia, Tsakhtadzor, 18–29 August, 2003.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Data Fusion Testbed for Recognized Maritime Picture

Eric LEFEBVRE
*Lockheed Martin Canada*

**Abstract.** This paper discusses a testbed for data fusion in the context of maritime surveillance. The testbed fuses non-real-time and near-real-time data from three different sensors, High Frequency Surface Wave Radar, Surveillance Aircraft, and ELINT data. The testbed correlator uses a Fuzzy Logic base clustering technique to associate ELINT data. For all data, the fusion process is performed at contact level and track level and resulting tracks are modeled with a genetic algorithm. The testbed interface allows the user to perform, among others, data files browsing, data contact edition and data visualisation on an interactive map. Current work on the implementation of level 2 fusion for situation awareness is also discussed. This testbed has been developed to test new techniques in the field of data fusion for maritime surveillance.

**Keywords.** Testbed, data fusion, maritime surveillance, clustering, genetic algorithm

## 1. Introduction

The Recognized Maritime Picture (RMP) is a result of all surveillance efforts, infrastructure, systems, plans or strategies from the maritime perspective. For national sovereignty purposes, RMP areas can include the 200 NM Exclusive Economic Zone (EEZ) and for defence purposes, extend well beyond. Civilian and Military maritime organisations may have access to a number of surveillance sources. A country's ability to make full use of these systems is limited by its ability to fuse the data from all data sources in a timely, accurate, and complete manner. These systems may include, among others, the High Frequency Surface Wave Radar (HFSWR) system, the Electronic Intelligence (ELINT) system, and Surveillance Aircraft (SA).

This paper presents a testbed that fuses data broadcast by the above cited information systems. The data that have to be correlated and combined do not follow exact and deterministic statistical models, as is the case with raw sensor data. They do not share the same kind of information and may be biased. The testbed makes full use of data parameters to reconstruct vessel tracks and to model them into mathematical objects that can be more easily handled for situation awareness.

Section 2 presents data characteristics that will drive the correlator process presented in Section 3. Section 4 provides an overview of the testbed interface while Section 5 presents some data fusion results. Section 6 discusses the current testbed development.

## 2. Data Characteristics

HFSWR, ELINT and SA data have different parameters. Table 1 shows parameters available for each data set. It should be noted that only time and position are common to all data sets. Since time and position may not be sufficient information for data fusion at contact level, especially if the update rate is low, the structure of the data favours a track reconstruction and data fusion at track level. This is done through the contact level association process with ID association, clustering and track modelling. ELINT data are received in batches every few hours while HFSWR and SA may be received in near-real-time but with a low update rate for SA.

**Table 1.** Data parameters

| | | HFSWR | ELINT | SA |
|---|---|---|---|---|
| geo- | time | * | * | * |
| positional | position | * | * | * |
| | area of uncertainty | * | * | |
| identity | ship name | | | * |
| | track number (UID) | * | | |
| | sensor name | * | | |
| EMAG | radio frequency | | * | |
| | pulse repetition interval | | * | |
| | pulse duration | | * | |
| update rate | high | * | | |
| | low | | * | * |

## 3. Correlation Process

The correlation process works in two stages. First a contact level association is performed using an incoming batch of contacts. At this stage contacts are grouped together to form tracks. When a track is formed it is queued in the second stage, which is the track level association. At this stage tracks are associated together based on a track's model instead of a contact's individual properties.

When a track does not have enough contacts to make a model and has not been merged to another track at track level association, its contacts remain available in a contact pool for the next contact level association process. The user triggers the contact level association process while the track level association process is triggered automatically when a new track is formed. The user can trigger contact level association repeatedly. Existing tracks will associate with newly formed tracks to update them at track level association.

### *3.1. Contact Level Association*

When contact level association (Figure 1) is performed, each contact is processed one by one. The first contact of the batch creates a new track. Then each subsequent contact is compared with previous contact properties. First, an Identification (ID) check is made. If a contact's sensor Unique Identifier (UID) or ship name is the same as that of a previously processed contact, both contacts are associated with the same track with-

out any other comparison. When four or more contacts are associated with the same track, the track is modelled into a collection of straight-line segments.

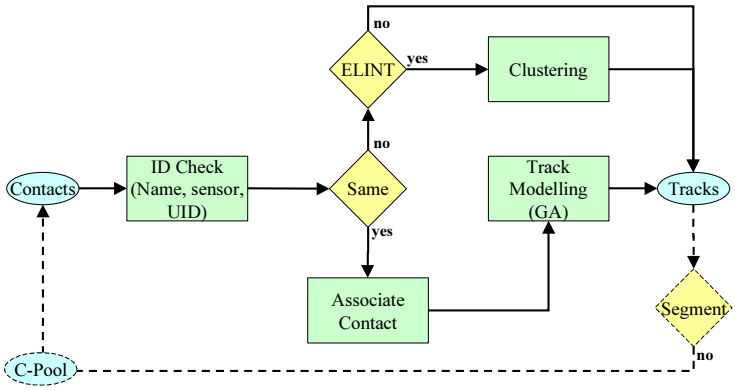



**Figure 1.** Contact level association process.

If the ID check is negative, a clustering process is used for contacts with ELINT information. All contacts belonging to the same cluster are associated with the same track. Finally, if a contact cannot be associated with a previous one by ID comparison or by a clustering technique, it creates a new track. Typically HFSWR contacts are associated based on their UIDs while ELINT, not having ID attributes, are associated using the clustering technique. Contacts provided by surveillance aircraft may be associated based on the ship name; otherwise, they will form single contact tracks.

Tracks with less than four contacts cannot be modelled and therefore cannot become segments. In this case, contacts belonging to these tracks are put into a contact pool, which will be added to the next contact batch to be processed if the tracks they belong to were not merged with other tracks at track level association.

### *3.1.1. Clustering*

Only contacts with Electromagnetic (EMAG) information are taken into account when clustering. Therefore, contacts must have a valid Radio Frequency (RF), Pulse Repetition Frequency (PRF), and Pulse Width (PW) or they will not be assigned to a cluster.

The clustering technique presented here is adapted from [1]. First the contact $j$ with the highest density is found to initialise the first cluster. The contact density is defined as the sum of membership multiplied by the geo-feasibility with other contacts:

$$density_j = \sum_{\substack{i=1 \\ i \neq j}}^{n} \mu_{ij} \delta_{ij}^{geo} \tag{1}$$

where $i$ and $j$ are the indices of the contacts, $n$ is the total number of contacts, and $\mu_{ij}$ is the membership between contact $i$ and $j$ given by

$$\mu_{ij} = \mathrm{MIN}(\mu_{ij}^{\mathrm{RF}}, \mu_{ij}^{\mathrm{PRF}}, \mu_{ij}^{\mathrm{PW}})$$
$$\mu_{ij}^{\mathrm{RF\,(PW)}} = \mathrm{MAX}\left(1 - \frac{\left|\mathrm{RF}_i(\mathrm{PW}_i) - \mathrm{RF}_j(\mathrm{PW}_j)\right|}{\sigma_{\mathrm{RF\,(PW)}}}, 0\right) \tag{2}$$
$$\mu_{ij}^{\mathrm{PRF}} = \mathrm{MAX}\left(1 - \frac{\left|B_{kl} \cdot \mathrm{PRF}_i - \mathrm{PRF}_j\right|}{\sigma_{\mathrm{PRF}}}, 0\right)$$

For $\mu^{\mathrm{PRF}}$, a basebanding [2] technique is used. $B_{kl}$ represents the elements of the basebanding vector:

$$B_{kl} = \frac{k}{l} \forall \left\{k, l \subset [1, m] : k, l, m \in \mathbb{N}\right\} \tag{3}$$

The geo-feasibility is calculated between two contacts by dividing the distance between contacts ($d_{ij}$) plus the radius of each error ellipse ($\sigma$) by the time difference between contacts.

$$v = \frac{d_{ij} + \sigma_i + \sigma_j}{\left|t_i - t_j\right|} \tag{4}$$

If the resulting speed $v$ is greater than 30 kts, positions are considered not geo-feasible and $\delta^{geo}$=0; otherwise, $\delta^{geo}$=1. If both contacts have the same time, $\delta^{geo}$=1 if $d_{ij} \leq \sigma_i + \sigma_j$, otherwise $\delta^{geo}$=0.

After the cluster is initiated, an iterative procedure is used to refine the cluster definition. Here is the outline of this procedure.

1. the cluster prototype $P_j$ for RF and PW is calculated using

$$P_j^{\mathrm{RF(PW)}} = \frac{\sum_{i=1}^{n} \mathrm{RF}_i\,(\mathrm{PW}_i) \cdot \rho_i^{(j)}}{\sum_{i=1}^{n} \rho_i^{(j)}} \tag{5}$$

where $\rho_i$ is the possibility that contact $i$ belongs to prototype $j$, which is given by

$$\rho_i^{(j)} = \frac{1}{1 + \mu_{ij}^2} \tag{6}$$

only contacts with $\rho^{(j)} > 0.5$ are considered for the computation of the prototype.

2. the membership $\mu_{ij}$ is calculated again with the prototype $P_j$ values calculated in equation 5.

Steps 1 and 2 are repeated until the normalised sum of the variation of possibilities between two iterations is lower than a fixed threshold

$$\frac{\sum_{i=1}^{n} \Delta\rho_i^{(j)}}{\sum_{i=1}^{n} \rho_i^{(j)}} < \varepsilon \tag{7}$$

then all contacts with $\rho^{(j)}>0.5$ belong to the cluster $j$.

When a cluster has been identified, a new high-density contact is found to initiate a new cluster $k$. Contacts with $\rho^{(j)}>0.5$ have their $\rho^{(k)}=0$ set for all iterations of the $k$ cluster. The algorithm ends when no contact remains with a density high enough to make a cluster ($\rho^{(k)}\leq 0.5$ for all contacts).

### *3.1.2. Modelling*

The algorithm used for modelling tracks is a hybrid optimisation algorithm based on both a genetic algorithm and simulated annealing, SAGACIA [3]. The different states of the track are represented with chromosomes (genetic algorithm). Chromosomes are made up from contacts ordered in time. Each gene represents a track's contact and has two possible values, 0 or 1. The value 1 indicates the beginning of a segment. The track's first and last contacts are always represented by 1. Moreover, each segment must have at least four contacts. Figure 2 shows a possible chromosome's representation for a 20-contact track. This track model has three segments and despite the last gene being equal to one, its corresponding contact belongs to Segment 3.

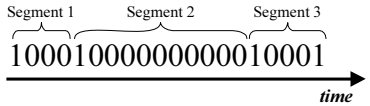



**Figure 2.** Chromosome representation of a track.

SAGACIA works from a population that takes advantage of the properties of genetic algorithms. The initial state of the population is randomly generated with the only constraint being that each segment must have at least four contacts. The state with only one segment is always present in the initial population.

The fitness or cost function is the sum of the chi-square per degree of freedom of all segments:

$$Cost = \sum_{s=1}^{ns} \left( \frac{\chi^2}{\nu} \right)_s \tag{8}$$

where $s$ is the segment number and $ns$ is the number of segments. The chi-square per degree of freedom is defined [4] by

$$\frac{\chi^2}{\nu} = \frac{1}{2n-4}\left\{\sum_{q=x,y}\sum_{i}^{n}\left(\frac{q_i - \left(\alpha^q + \beta^q t_i\right)}{\sigma_i}\right)^2\right\} \tag{9}$$

where $n$ is the segment's number of contacts, $q$ is one of the two orthogonal referential axes (here $q$ is either along the longitude or along the latitude at each contact position $(x,y)_i$), $t_i$ is the time of the $i^{th}$ contact, and $\alpha^q$ and $\beta^q$ are the adjusted parameters given by:

$$\alpha^q = \frac{\sum_{i}^{n}\frac{q_i}{\sigma_i^2} - \sum_{i}^{n}\frac{q_i t_i}{\sigma_i^2}}{\sum_{i}^{n}\frac{1}{\sigma_i^2} - \sum_{ii}^{n}\frac{t_i}{\sigma_i^2}} \;,\; \beta^q = \frac{\sum_{i}^{n}\frac{q_i}{\sigma_i^2} - \alpha^q\sum_{i}^{n}\frac{1}{\sigma_i^2}}{\sum_{i}^{n}\frac{t_i}{\sigma_i^2}} \tag{10}$$

With $\sigma_i$ defined by the maximum error projection on the orthogonal referential axes.

$$\sigma_x = \mathrm{MAX}\left(a\sin\theta, b\cos\theta\right) \;,\; \sigma_y = \mathrm{MAX}\left(a\cos\theta, b\sin\theta\right) \tag{11}$$

where $a$ and $b$ are respectively the semi-major and semi-minor axes and $\theta$ the error ellipse orientation.

The selection method for each generation is elitist and cost-proportionate "roulette-wheel sampling." The algorithm ends when the state with the lower cost is the same for 10 generations in a row. This state defines the number and the distribution of segments in the track model.

For this optimisation situation the genetic algorithm gives good results. Usual genetic algorithm downsides like cpu time consumption, possibility of being trapped in a local minimum, and difficult gene transposition do not apply here. Transposition into binary chromosomes is straightforward and track segments are well enough defined to allow small populations (around 30 individuals) and to prevent many local minima in the fitness function.

### *3.2. Track Level Association*

Track level association (Figure 3) is performed when the contact level association has created a new track. Each new track is compared with existing ones (track pool). If the track has no defined segment (track with less than four contacts) it is only compared with existing tracks with defined segments. In this case, to be eligible for a track association, the track's contacts must be position and ID compatible with the track model that it is being compared with. If both tracks (the new one and the one that it is being compared with) have defined segments, a segment-by-segment comparison is performed. Two tracks are associated only if the association is feasible and if each track of

the pair represents the best match for the other. When two tracks are merged into a single one, the resulting track is modelled and compared again with all the tracks in the track pool. When no further association can be performed, the track is put in the track pool.

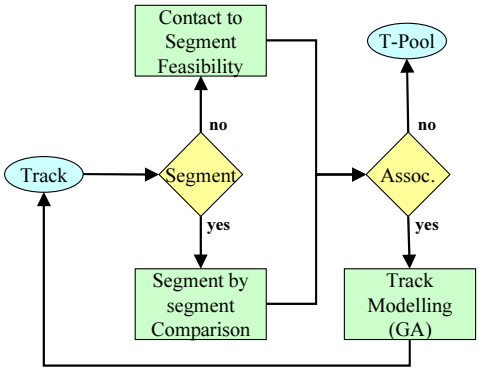



**Figure 3.** Track level association process.

*3.2.1. Segment-to-Segment Association*

This process determines the merging feasibility of two tracks by comparing their segments. When the two track segments do not overlap, they are extrapolated to find an intersection in space and time. In this case, tracks may or may not have the same course (Figure 4a). The merging feasibility is then evaluated with

$$feasibility = \mathrm{MAX}\left(0,\ 1-\frac{d_{12}}{\sigma_1+\sigma_2+bias}\right) \tag{12}$$

where $d_{12}$ is the distance between the two extrapolated track points and $\sigma_i$ is the radius of the error ellipse of point $i$ along $d_{12}$. The *bias* is the track bias given by the user.

If track segments overlap, the extremities of the overlap must be geo-feasible (Figure 4b). The geo-feasibility ($\delta^{geo}$) is evaluated by comparing track positions at common times as described in Paragraph 3.1.1. Here $d_{12}$ represents the two distances between the two tracks at both extremities of the overlap. The total merging feasibility is the average of the feasibility of both extremities.

When a track's overlap spawns over several segments (Figure 4c), the merging feasibility is the average of all segment overlaps.

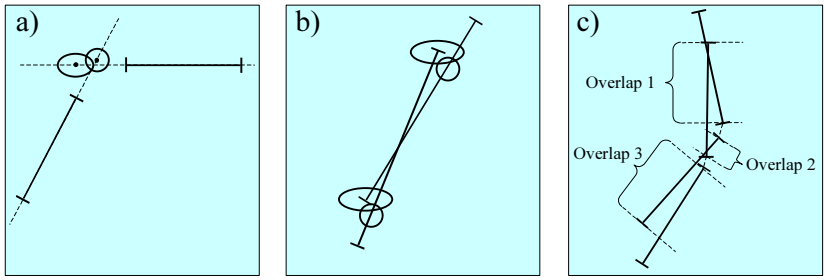



**Figure 4.** Possibilities for segment-to-segment association.

### *3.2.2. Contact-to-Segment Association*

This process evaluates the feasibility of a group of contacts (GC), or tracks without models, to be associated with a modelled track. The GC is evaluated against each track model in the track pool. To perform this, each track model is first aligned in time with each contact in the GC. Then a contact score representing the feasibility between each contact and the track model is evaluated using the geo-feasibility between the contact and the corresponding point on the track's segment. The contact score may have the following value:

a) if there is a time overlap between the segment and the contact but the contact is not geo-feasible with the segment, the score is –1;
b) if there is no time overlap between the segment and the contact and the contact is not geo-feasible with the segment, the score is 0;
c) if there is no time overlap between the segment and the contact but the contact is geo-feasible with the extrapolation of the segment, the score is 1;
d) if there is a time overlap between the segment and the contact and the contact is geo-feasible with the segment, the score is 2.

A contact matrix is built to record all contact scores for all tracks. The track score, which is the sum of all tracks' contact scores in the contact matrix is also computed.

A GC will merge *a priori* with all tracks that have their higher positive track score with that GC. However, to prevent the merging of two overlapping modelled tracks by this process (this must be exclusive to the segment-to-segment process) a contact-by-contact comparison is made in the contact matrix (Figure 5). If two tracks have a contact score of 2 for the same contact, the track with the lowest track score is discarded from the merge. If the two tracks have the same track score it is identified as a conflict and no merge occurs.

| GC | Track 1 | Track 2 | Track 3 | |
|---|---|---|---|---|
| Contact 1 | 0 | -1 | 1 | ⇒ Track 3 is kept, track score > 0 |
| Contact 2 | 0 | 1 | 2 | ⇒ Track 3 is kept, higher score than Track 2 |
| Contact 3 | 0 | 2 | 1 | ⇒ Track 2 is kept, higher score than Track 3 |
| Contact 4 | 1 | 2 | 0 | ⇒ Track 2 is kept, higher score than Track 1 |
| Contact 5 | 2 | 2 | 0 | ⇒ Track 2 is kept, Track 1 is rejected based on track score |
| Contact 6 | 2 | 0 | 1 | ⇒ Track 1 was rejected, Track 3 is kept |
| track score | 5 | 6 | 5 | ⇒ GC, Track 2 and Track 3 are merged, Track 1 was rejected |

**Figure 5.** Contact matrix and association results.

## 4. The Testbed Interface

The testbed interface is divided into four windows as can be seen in Figure 6.

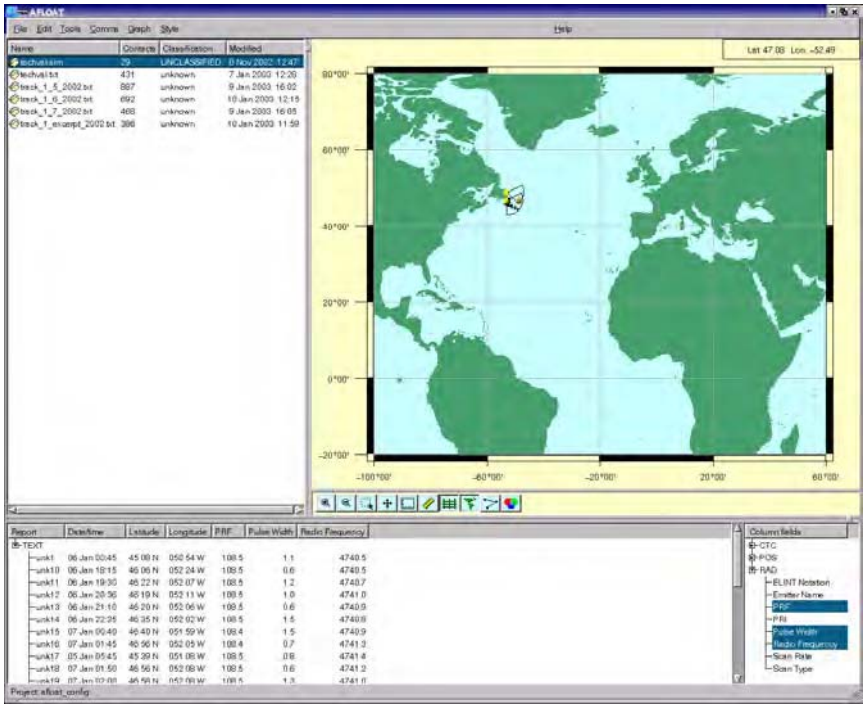


**Figure 6.** The testbed interface.

The upper left window is the list of loaded files. The upper right one is the display on an interactive map of contacts belonging to the selected file in the upper left window. The bottom left window is the editable list of contacts belonging to the current selected files. The bottom right window is the menu for displaying the contact's information in the bottom left window. Contacts can also be edited by clicking on them in the list or in the display window.

## 5. Results

The following scenarios show the testbed track association and modelling capabilities. The first scenario presents association of HFSWR and SA data. The second scenario presents HFSWR and ELINT data association.

### *5.1. HFSWR with SA*

Figure 7 shows contacts of both HFSWR and SA data sets.

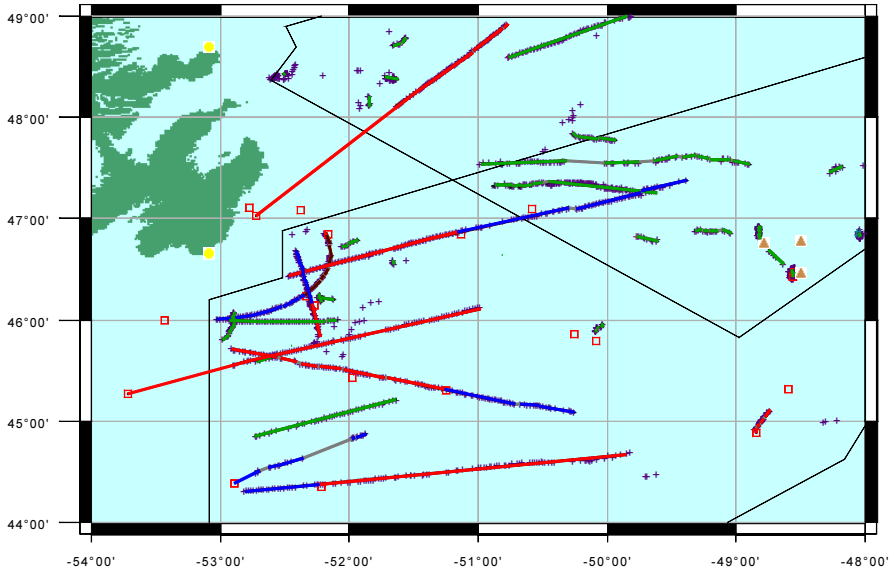



**Figure 7.** HFSWR-SA data fusion and modelling.

Squares represent SA contact positions while crosses represent HFSWR contact positions. Triangles are oilrigs and dots on the coast (46°38'N 53°05'W and 48°41'N 53°05'W) represent HFSWR location. Lines delimit radar coverage areas.

The resulting track models of the correlation are superimposed on the data as solid lines. Each model is composed of one or several straight-line segments.

Colour coded lines can be used to represent fusion information (e.g., red lines representing information from HFSWR before fusion with SA, blue lines representing HFSWR information after fusion with SA, green lines representing tracks with only HFSWR information, or purple lines representing information between first and last HFSWR-SA data fusion points). The superimposition of track models over initial data allows observing the data fusion due to the extrapolation of the HFSWR tracks. Two of these extrapolations permit the fusion of HFSWR tracks with data outside the radar coverage area.

### *5.2. HFSWR with ELINT*

In this example Global Positioning System (GPS) data were available for one ship, simulated ELINT has been generated from this ground truth using the testbed's Monte Carlo ELINT simulator. In Figure 8, squares represent this simulated ELINT while crosses represent three sets of HFSWR data.

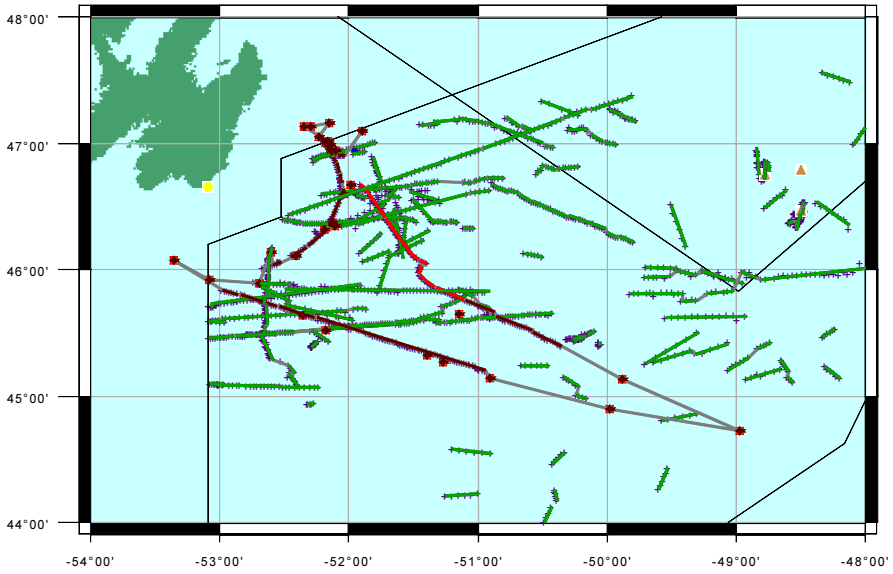



**Figure 8.** HFSWR-ELINT data fusion and modelling.

Again resulting track models are superimposed to the data. Dots overlaying the squares represent ELINT contacts that have been put into the same cluster by the clustering function. It should be noted that light grey lines do not represent a track model as dark lines do. They are only connectors to show temporal succession of ELINT contacts in the cluster. Track model exists only when HFSWR data are available.

The contact matrix generated by the contact-to-segment process allows more than one track association and several HFSWR tracks with large time gaps to be merged together. Each time new information is available it can be sent to the testbed to update the track models. This new information can be backwards in time or represent the newest information since the complete track model is always available for fusion.

## 6. Current Development

In [5], it is shown that HFSWR can provide substantial benefits to the RMP. Maritime vignettes were produced from the Forces Planning Scenarios to present sensor capabilities to the decision makers in a logical and intuitive way. These vignettes are used to evaluate the contribution of HFSWR to the RMP. Some vignettes also give indications of the vessel's behaviours. This knowledge of possible vessel behaviour may be used to perform automatic scenario recognition. Automated association of track information from sensor sources with non-sensor information is level 2 fusion.

General parameters can be extracted from the maritime vignettes and used to develop a plan editor. For example, the following three characteristics could be used for describing different plans:

- provenance or destination of vessel is known;
- vessel intercepts a known region;
- vessel course is known.

Figure 9 shows the testbed plan interface. The listed plans on the left, created by the user, are activated or deactivated depending on the user's needs. The right panel allows the user to edit specific parameters for each plan. These parameters can be used alone or in combination. For each plan, these parameters are used to define a set of rules. When activated, each plan selects tracks that match its rules. Then, only selected tracks are shown on the testbed display window – plans acting as filters.

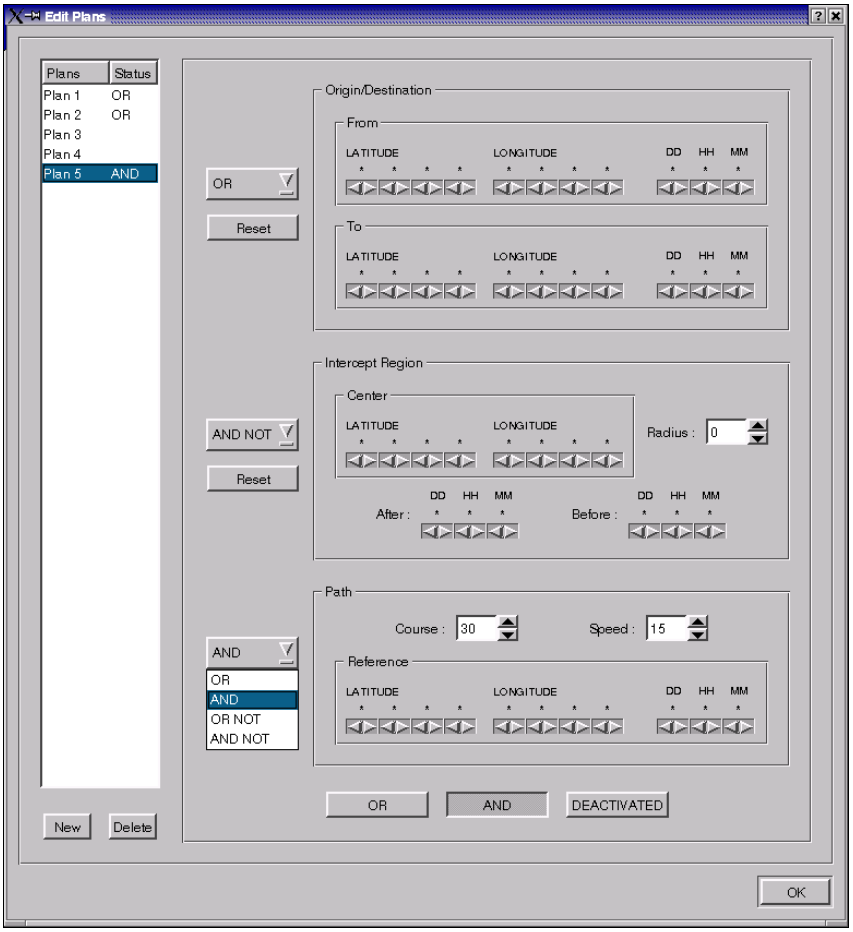



**Figure 9.** Testbed scenario interface.

## 7. Conclusions

A testbed for data fusion in the context of maritime surveillance has been presented. The characteristics of the data used favour a two stage association. First a contact level association performs track reconstruction from individual contacts and modelling of the tracks using a clustering technique and a genetic algorithm. The second stage involves track association based on track models. The testbed interface allows the display of contacts and track models on an interactive map. File browsing and contact edition is also possible.

Results show that data fusion may extend beyond the radar coverage region since track models can be extrapolated. Data fusion can also be done backward in time and is independent of the order of data received. Finally since track models reduce significantly the amount of information compared with all contacts taken individually, longer track history can be kept in information systems where the size of a database is limited.

Current development of the testbed implements automated ship scenario recognition to facilitate situation assessment and perform level 2 fusion.

## Acknowledgements

The testbed presented in this paper has been developed under funding from Defence Research and Development Canada Ottawa (DRDC-Ottawa).

## References

[1] C. Helleur, Maritime Track Recognition Through Sequential Possibilistic Clustering, to be published.
[2] R. Wiley, Electronic Intelligence: The Analysis of Radar Signals, Artech House, 1993.
[3] B. Li, W. Jiang, A Novel Stochastic Optimization Algorithm, IEEE Transactions on Systems, Man and Cybernetics – Part B : Cybernetics, 30(1), 2000, 193–198.
[4] N. Gershenfeld, The Nature of Mathematical Modeling, Cambridge University Press, 1999.
[5] S. Doré, C. Helleur, C.W. Van Fong, Scenario-Based Assessment of Sensors for the Canadian Recognized Maritime Picture, Proceeding of the 7th ICCRTS, September 2002.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Comparisons of Track-Level Fusion Results with Tracklets Application Within a Simulation Environment

Eric MÉNARD, Daniel TURGEON, Guy MICHAUD and Elisa SHAHBAZIAN
*Lockheed Martin Canada, 6111 Royalmount Avenue, Montreal, Quebec, H4P 1K6*

**Abstract.** Over the last years, Lockheed Martin Canada has developed a Testbed to regroup and analyze fusion architectures, algorithms and information sharing strategies. This Testbed is used to demonstrate practical implementations of distributed data fusion between multiple collaborating platforms. In a decentralized data fusion center, there are many algorithms that process positional information coming from a network to generate a Global Tactical Picture. These algorithms tend to remove or prevent cross-correlation from being part of the received data. This paper compares four different track fusion algorithms applied in a simulation environment. There are two implementations of the Tracklet fusion approach, an algorithm based on track quality and an algorithm based on the source of the information.

**Keywords.** Tracklet, data fusion, centralized, decentralized, Kalman filter

## 1. Introduction

Decentralized data fusion systems share information between local data fusion systems to build a Global Tactical Picture (GTP). There are many methods available to process remote tracks information received via a network. When the information is fused before being broadcasted on the network, it is important to carefully process the information in order to minimize cross-correlation during further fusion.

There are many methods to remove or prevent positional cross-correlation to be part of the received data. The first one is track quality. This method consists in resetting the covariance matrix to a predefined matrix, which is associated to a track quality value. The fused covariance matrix is never re-injected into the next fusion step. The second method is the Selective Position Fusion (SPF). This method decides, based on the source of information, whether the new information is fused with the old one or the new one just replaces the old information. The third method is one of the tracklet fusion methods: the Inverse Kalman Filter developed by G. Frenkel [1,2]. This method consists in computing an equivalent "contact" to be used by the fusion process. The last method is the Inverse Information Filter. The approach is similar to the Inverse Kalman Filter, but the tracklet is computed by inverting the Information Form [1,2].

The purpose of this paper is to present an application of tracklets within a simulation environment and assess its performance. The results obtained using the track quality, the selective position and the tracklet methods are compared with centralized contact fusion results in order to assess the accuracy of the targets' position and the estima-

tion of positional error. The comparison is based on scenarios involving linear and maneuvering trajectories.

Section 2 is an overview of the Lockheed Martin Canada (LM Canada) TESTBED [3]. Section 3 describes different fusion algorithms to remove or prevent cross-correlation. Section 4 explains the scenario used to compare track fusion algorithms. Section 5 presents the results and, finally, section 6 provides the conclusion.

## 2. Design Overview

The Multi-Platform R&D Testbed developed by LM Canada provides the infrastructure to support the algorithmic development, communication exchange between Command and Control Information System (CCIS) and agent based approaches for rule-based information management implementation [4]. The local Multi Sensors Data Fusion (MSDF) is a centralized fusion center for the sensors on the Platform Units (PU) generating the Local Area Picture (LAP). The global MSDF creates the Global Tactical Picture (GTP) by the merging and fusion of the LAP and the information coming from a LINK-11 like communication network (Figure 1).

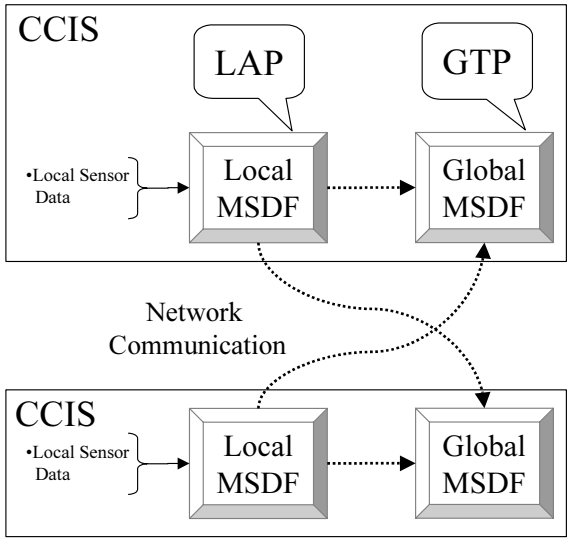



**Figure 1.** High level fusion architecture.

### *2.1. Reported Track Data Store*

In order to be as general as possible, each fusion node broadcasts tracks from its local track datastore: no tracklet is broadcasted. If required, the receiving fusion node then computes tracklets for each track received from each node. Tracklet computation is performed in the Global MSDF for each fusion node. To perform the various tracklet computations described above, Global MSDF needs to store the latest reported track state and covariance matrix, and the last computed tracklet, for each track reported by each fusion node. Reported tracks and their associated tracklets are stored in the Reported Track Data Store, a structure of the Global MSDF.

### *2.2. Tracklets Computation*

Tracklet methods are applied only when the received data contains positional cross-correlation. When data is received from its Local MSDF, the global fusion node replaces the related global track state with the received track state. In fact, fusion was already performed by Local MSDF and there was no need to compute an equivalent measurement and fuse it again. Performing these computations again would be time-consuming. At the beginning of the fusion process, the tracklet computation is performed using the new buffer of remote tracks. The tracklet is computed only if all information needed is available for the given remote track, i.e., the last reported track (for the Inverse Information filter) or the last tracklet (for the Inverse Kalman filter). If this information is not available, the system uses the latest report as a tracklet to initialize the tracklets' processing. In the Reported Track Data Store, the computed tracklet is stored together with the new reported track, replacing previously stored information. This information will be available during the next buffer processing. All this processing is performed during the data alignment step into the fusion process (Figure 2).

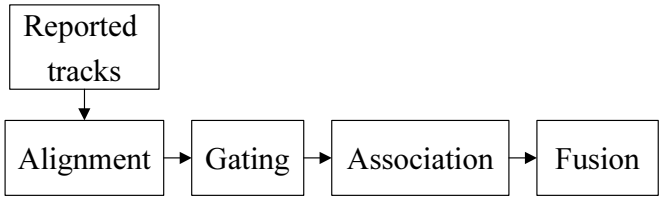



**Figure 2.** Global MSDF data flow of the fusion process.

Tracklet computation algorithms require that the covariance matrix be positive definite. To prevent loss of positive definiteness during matrix subtraction, the system validates the result of each computation. If the above subtraction is a non-positively defined matrix, a multiple of identity matrix is added to the result in order to reestablish the positive definiteness of the subtraction.

### *2.3. Gating and Position Fusion*

Figure 2 shows the data flow of the fusion process. Prior to positional fusion, the gating process is performed on reported tracks and global tracks. The computed tracklets are not used during gating. Track-to-track association is then performed to determine which reported track is associated with which global track. Then, each confirmed track/track pair is fused using one of the position fusion algorithms, e.g., the Extended Adaptive Kalman Filter (EAKF) or the Interacting Multiple Model (IMM). The inputs to these fusion algorithms are the computed tracklets and the time updated global tracks.

## 3. Track Fusion

When fusion nodes exchange track information (as opposed to contact data), track fusion algorithms are required. Track fusion differs from contact fusion due to the cross-correlation of tracks. With contact fusion, each contact provides independent informa-

tion and its positional information can therefore be fused with a standard filter like the Kalman or IMM filters. On the other hand, since a track is the output of a filtering process, it does not contain independent information and fusing it without pre-processing boils down to fusing identical information over and over. Eventually, such a filter will become overly confident and may diverge. The following paragraphs describe the design for the implementation of fusion algorithms in the LM Canada Testbed.

### *3.1. Selective Position Fusion with Covariance Matrix*

The Selective Position Fusion involves processing local and remote positional data differently to prevent data incest. Local positional data come from the Local MSDF, while remote data come from other CCIS on the network. Depending on the source, the position of a reported track may be fused to the global track's position or it may simply update (replace) the position of the global track.

Track position information from the Local MSDF and from the network is never fused into a global track since tracking has already been performed by Local MSDF or the remote CCIS. The complete covariance matrix is received by the Global MSDF. Fusing this information again will introduce cross-correlation and make the Kalman filter overly confident. The positional information is only added to the global track history.

### *3.2. Selective Position Fusion with Track Quality*

The Selective Position Fusion with track quality (TQ) method is like the Selective Position Fusion method with covariance matrix for the local positional data. However, for the remote positional data, a TQ is received and it is mapped into a circular area of uncertainty to quantize the covariance matrix. The uncertainty of a track is then reported as a number between, for example, 1 and 15. This approach is similar to using remote tracks as contacts during fusion for removing cross-correlation and allowing the fusion of reported tracks with standard filters. The inconvenience with the TQ is that the error on the track is generally over-estimated.

### *3.3. Inverse Kalman Filter*

The Inverse Kalman filter method consists in computing an equivalent measurement (tracklet) for each track update received from a particular source. The global tracker does not receive the original radar contact data, but the results of the tracking performed by the reporting unit. The new state vector of the reported track received at time $t_n$ is denoted by $X_n^j$, while the covariance matrix at the same time is denoted by $P_n^j$. The superscript *j* denotes the reporting source of the track. In order to compute the new tracklet at time $t_n > t_m$, the last computed tracklet state vector $u_m^j$ and covariance matrix $U_m^j$ from source *j* at time $t_m$ must be time updated to $t_n$ using the following equations

$$\begin{aligned}\Delta t &= t_n - t_m \\ u^j_{n|m} &= \Phi(\Delta t) u^j_m \\ U^j_{n|m} &= \Phi(\Delta t) U^j_m \Phi^T(\Delta t)\end{aligned} \tag{1}$$

where $\Phi(\Delta t)$ is the state transition matrix and the time updated tracklet is noted as ($u^j_{n|m}$, $U^j_{n|m}$). Note that the process noise is assumed null for the time update of the covariance matrix of the tracklet. The following equations compute the new tracklet state vector $u^j_n$ and the new covariance matrix $U^j_n$ for the source $j$ to the time $t_n$.

$$\begin{aligned}u^j_n &= u^j_{n|m} + U^j_{n|m}\left[U^j_{n|m} - P^j_n\right]^{-1}\left[X^j_n - u^j_{n|m}\right] \\ U^j_n &= U^j_{n|m}\left[U^j_{n|m} - P^j_n\right]^{-1} U^j_{n|m} - U^j_{n|m}\end{aligned} \tag{2}$$

Note also that a tracklet has the same dimension as a track, i.e., it has both position and velocity components.

### *3.4. Inverse Information Filter*

The Inverse Information Filter is similar to the Inverse Kalman Filter when computing an equivalent measurement. However, as opposed to the Inverse Kalman Filter, the Inverse Information Filter takes process noise into account. As with the Inverse Kalman Filter, the newly received information is the state vector $X^j_n$ and its covariance matrix $P^j_n$ from source *j* at time $t_n$. The tracklet computation needs the previously received information ($X^j_m$, $P^j_m$) at time $t_m$ to be propagated to time $t_n$. The time updated information state vector and its covariance matrix are noted $X^j_{n|m}$ and $P^j_{n|m}$, respectively. The following equations compute the tracklet's state vector $u^j_n$ and covariance matrix $U^j_n$ for the source $j$ at time $t_n$.

$$\begin{aligned}U^j_n &= \left[P^{j^{-1}}_n - P^{j\ -1}_{n|m}\right]^{-1} \\ u^j_n &= U^j_n\left[P^{j^{-1}}_n X^j_n - P^{j\ -1}_{n|m} X^j_{n|m}\right]\end{aligned} \tag{3}$$

where the time update is given by

$$\begin{aligned} \Delta t &= t_n - t_m \\ X_{n|m}^j &= \Phi(\Delta t) X_m^j \\ P_{n|m}^j &= \Phi(\Delta t) P_m^j \Phi^T(\Delta t) + Q \end{aligned} \tag{4}$$

where $\Phi(\Delta t)$ is the state transition matrix and Q is the process noise estimation for the reported track. Q is variable and is estimated based on the speed variance of the reported track. The speed variance is computed from linear regression of last "*N*" speed estimates of the reported track.

## 4. Test Scenario

The data used for the comparison of the fusion algorithms comes from a scenario (see figure 3) that has two reporting units of Halifax class ships. These two reporting units observe targets with their own sensors and broadcast fused track information to other PU on the network through a LINK-11 like system. This system interrogates each PU according to the network polling cycle time to get track information from a given PU and send it to other PUs. There are also four PUs that are so far from the targets maneuvering area that receive only information by the network, i.e. the targets are outside of the PUs sensors' range. To compare the above mentioned methods, each PU uses a different positional fusion algorithm.

The test scenario is divided into two different parts. The first one involves only the PU 1 as reporting unit, PU 2 being disabled for this test. The receiving units fuse the data received from the reporting unit with different fusion algorithms (SPF with TQ, Inverse Kalman Filter and Inverse Information Filter) to compare these algorithms with the reporting unit results.

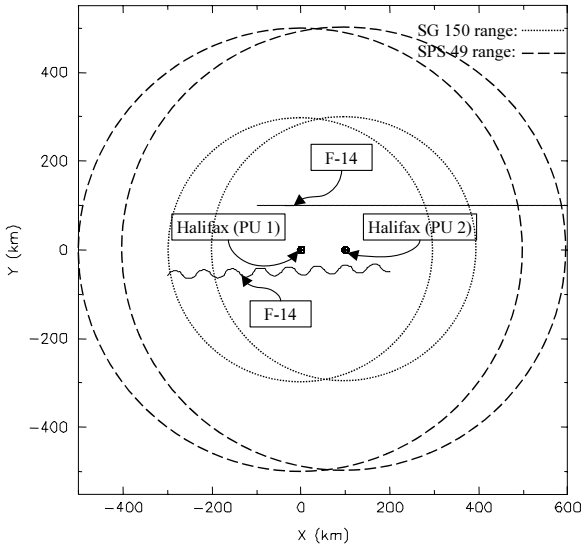



**Figure 3.** Scenario used to compare positional fusion methods.

The second part has two reporting units (PU 1 and PU 2) that observe all the targets and broadcast their track data to the four other PUs. These four 'listening" PU receive information only from the network and no information is available from their local sensors. This portion of the scenario is used to compare the fusion of two sources coming from the network with the local sensors of the two reporting units fused by a centralized data fusion center.

### *4.1. Observing PU Sensors*

Each Halifax class PU in this scenario was equipped with the simulated sensors listed in table 1.

**Table 1.** Sensors on the simulated Halifax class PU

| sensors | maximum detection range (km) | scan rate (RPM) | standard deviation in range (m) | standard deviation in bearing (rad) |
|---|---|---|---|---|
| SPS 49 | 500 | 12 | 250 | 0.03 |
| SG 150 | 300 | 60 | 100 | 0.01 |
| IFF 49 | 250 | 12 (coupled to SPS 49) | 500 | 0.087 |
| ESM | 500 | N/A | N/A | 0.087 |

The results were performed while the two PUs observed the targets simultaneously. This time frame is used to observe the fusion algorithm's behavior when different sensor reports are available and fused together.

## 5. Results

The comparison will be based on the position track state and the trace of the positional part of the covariance matrix. The trace is the sum of the semi-major axis and the semi-minor axis of the positional uncertainty and is used as a measure of the positional error.

This section compares track fusion algorithms, namely the Inverse Kalman Filter, the Inverse Information Filter, the Selective Position Fusion with TQ and the Selective Position Fusion with covariance matrix. For each of these algorithms, the Kalman filter is used to fuse either the reported track or the tracklet with the global track.

### *5.1. One Reporting Unit*

This scenario has only one reporting unit, PU 1. Therefore, the "optimal results" for the global MSDF must be similar to the local track state and covariance matrix since this is the only data available. The purpose of this test is to compare the consistency of the various fusion approaches.

The output of each tracking method is compared with the Local MSDF tracking of the reporting unit. The position track state and the trace of the covariance matrix are analyzed. In this scenario, there is no need to look at the results of the Selective Position Fusion with covariance matrix since they are identical to the local fusion node data.

By looking at the F-14 going a in straight line, figure 4 shows the difference of the trace of the covariance matrix with the reporting PU. The best methods are those for which the difference is close to zero, i.e., when the error is similar to the local track's error. The Inverse Information Filter outperformed the two other methods and corresponds to local data, while the Inverse Kalman Filter is close to the local data but not as close as the Inverse Information Filter. This difference is caused by the process noise, which is assumed null by the Inverse Kalman Filter. For the SPF with TQ, using TQ to generate contact-like covariance matrices causes the system to overestimate the fused covariance matrix. As seen in figure 4, the trace of the covariance matrix of the SPF with TQ is larger than the other algorithms, as expected.

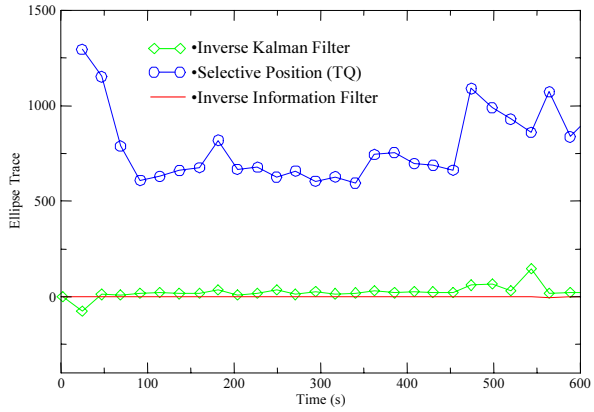



**Figure 4.** Non-maneuvering target, 1 PU reporting, ellipse trace difference with reporting PU.

Figure 5 shows the difference of the Y component of the state vector between the track fusion algorithm and the output of the reporting PU.

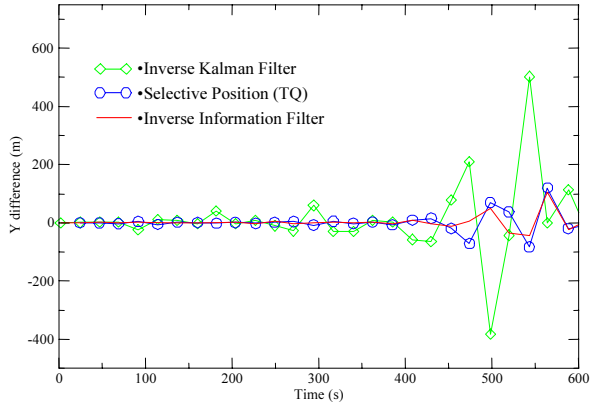



**Figure 5.** Non-maneuvering target, 1 PU reporting, Y difference with reporting PU.

The best methods are those for which the difference is close to zero, i.e., when the position is similar to the local track's position. All the algorithms performed well and were very close to the local fusion node. However, after 450 seconds, one sensor stops

reporting on the target and its position becomes less stable in the observing Local MSDF. This behavior is then reflected in the Global MDSF of the receiving PUs. Since the Inverse Kalman Filter neglects process noise, it becomes over confident and its estimated position is even less stable than the other track fusion algorithms.

By looking at the other F-14 doing an oscillatory maneuvering pattern, figure 6 shows the same behavior as for the non-maneuvering target, regarding the ellipse trace. The Inverse Information Filter is the best filter and follows the local fusion node. The selective position filter with TQ is the worst regarding the covariance matrix, because it has a larger uncertainty.

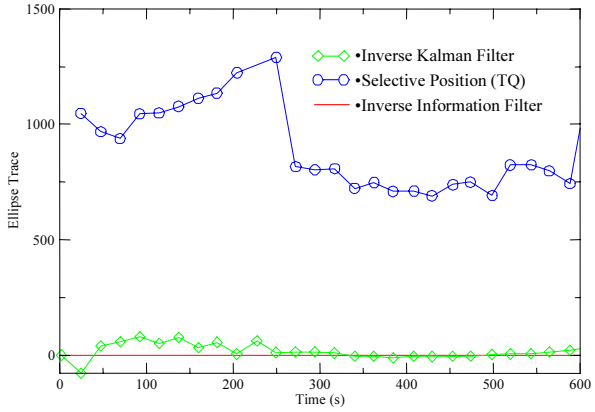



**Figure 6.** Sinusoidal maneuvering target, 1 PU reporting, ellipse trace difference with reporting PU.

Figure 7 shows the Y difference on the positional data. The Inverse Information Filter gets the best results in following the Local MSDF results. The Inverse Kalman Filter has the worst positional accuracy. These oscillations are due to the process noise that is assumed null in the Inverse Kalman Filter, while the target is maneuvering.

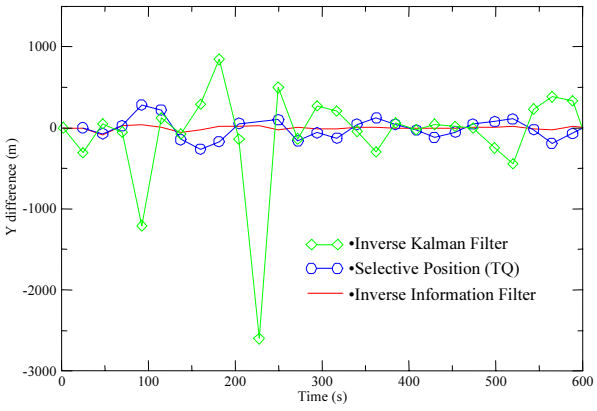



**Figure 7.** Sinusoidal maneuvring target, 1 PU reporting, Y position difference with reporting PU.

The SPF with TQ gets better results in position than the Inverse Kalman Filter, although it has a larger error. In the case of the SPF with TQ, the reported track is treated

as a contact with a large error, while the time updated global track has a small error. Consequently, the tracker "trusts" more the global track than the reported track when filtering. For the Inverse Kalman Filter, the computed tracklet has a small error due to the null Process Noise, and then the tracker "trusts," i.e. places more trust in the tracklet. Therefore, the position estimated by the SPF with TQ follows more closely the Local MSDF results. On the other hand, the position estimated by the IKF follows the computed tracklets more closely and the estimated global track error becomes small. This effect is observed for both non-maneuvering and maneuvering targets.

### *5.2. Two Reporting Units*

The second part of the analysis has two reporting units, each observing all targets with their sensors. In this section, the fusion algorithms will be compared with the results of a virtual test PU that receives all contacts from the two reporting units (PU 1 and PU 2). The results of this virtual test PU are the "optimal result" as they correspond to a centralized data fusion system [4].

The compared methods are the Inverse Kalman Filter, the Inverse Information Filter, the SPF with TQ and the SPF with covariance matrix. Let us suppose that the contact fusion of two sources has better precision than the fusion of one source only, then the SPF with covariance matrix can be used as an estimate of the upper bound for the "contact fusion" positional error. This approximation makes sense since the SPF with the covariance matrix method returns the covariance matrix of the latest report. Because the global node fuses two sources of data, it is expected that the overall result of the fusion has a lower positional uncertainty than SPF with covariance matrix. Then the best methods will be those that are below SPF with covariance matrix, and closest to the centralized data fusion results.

Figure 8 shows the trace of the covariance matrix for the straight-line trajectory and figure 9 for the sinusoidal trajectory. As expected, the trace obtained from the SPF with TQ is larger, always above the SPF with covariance matrix. The Inverse Information Filter is always below the SPF with covariance matrix, but often very close. The Inverse Kalman Filter results follow very closely the Inverse Information Filter results.

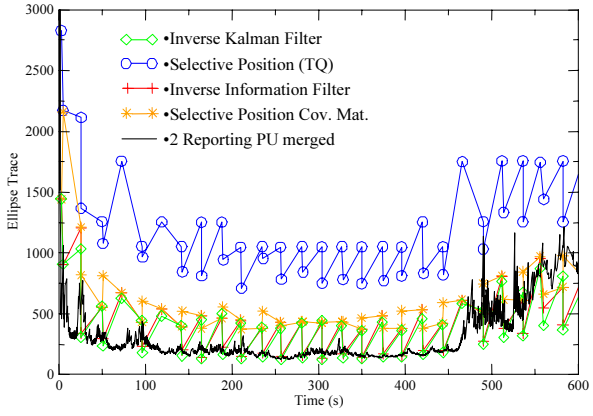



**Figure 8.** Non-maneuvering target, 2 PU reporting, ellipse trace with reporting PU.

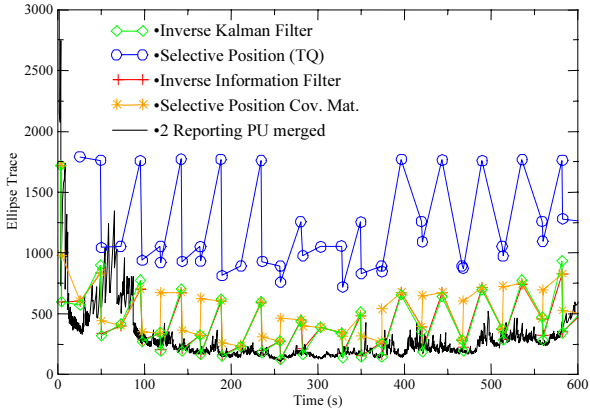



**Figure 9.** Sinusoidal maneuvering target, 2 PU reporting, ellipse trace with reporting PU.

The oscillations observed for all graphs are the consequence of the two reporting PUs sending their data at approximately 1-second intervals with a Net polling cycle time of 20 seconds. Having two track updates in a small-time frame makes the tracker more confident of the resulting position. Then the 20 second time lapse is taken into account and this enlarges the covariance matrix computed by the time update of the track. This effect is observed for both F-14 trajectories.

## 6. Conclusions

Track fusion algorithms were tested and analyzed in the LM Canada Testbed. The constraints of the Testbed force the Global MSDF of each PU to make no assumption on the configuration of the remote tracker that is reporting tracks. For example, the process noise used by the remote tracker is unknown when a reported track is received and this information cannot be used during track fusion of the receiving unit. For this reason, the process noise needed by the Inverse Information Filter is estimated from the reported track and is used by the Global MSDF of the receiving unit.

From the above analysis of track fusion algorithms on maneuvering and non-maneuvering targets, the Inverse Information Filter generally outperforms the other methods in the given scenarios. With one reporting unit, this method follows the results of the local track on the reporting unit. With two reporting units, the Inverse Information Filter results are just a little below of that of the SPF with covariance matrix and close to the centralized Reporting Units results.

However, on a real system, the full covariance matrix is not necessarily available for reported tracks. Therefore, none of the methods based on the covariance matrix may be used in this context. The only method left is the SPF with TQ. Although this method overestimates the error on the track, it does provide a consistent estimate of the position of a track. When only TQ information is available, the SPF with TQ method should be used.

Finally, there are avenues to investigate. Since Covariance Intersection tends to overestimate the fused covariance matrix, it will be interesting to compare the Covariance Intersection method with the SPF with TQ, which also overestimates the fused covariance matrix. Another topic of interest would be to study the effect of varying the

network polling cycle time on track fusion. Another study could also be performed to analyze reactions of the fusion engine when targets go out of the short range sensor, which usually provides better positional estimation. Finally, the effect of miss-association on track fusion algorithms could be studied.

## References

[1] O.E. Drummond, Track and Tracklet Fusion Filtering Using Data from Distributed Sensors, Proceedings of the Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom, pp. 167–186, May 2001.

[2] A.R. Benaskeur and J. Roy, A comparative study of data fusion algorithms for target tracking, Technical report # TR 2001-224, Defence Research & Development Canada – Valcartier, pp. 1–75, October 2002.

[3] A. Jouan., L. Baril, G. Michaud, E. Shahbazian, Distributed Data Fusion Architecture for the Evaluation of Information Sharing Strategies, SYSTEMS CONCEPTS AND INTEGRATION PANEL, the SCI-116 Symposium, Norfolk, Virginia, 21 to 23 October 2002.

[4] P. Bergeron, J. Couture, J.-R. Duquet, M. Macieszczak, and M. Mayrand, A New Knowledge-Based System for the Study of Situation and Threat Assessment in the Context of Naval Warfare in FUSION 98, Las Vegas, 6–9 July 1998, Vol. II, pp. 926–933.

[5] Hall, D.L. and Llinas, J., Handbook of Multisensor Data Fusion, CRC Press, 2001.

This page intentionally left blank

# 4. DATA FUSION FOR IMAGERY

This page intentionally left blank

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Multisensors and Contextual Information Fusion

## *Application to Object Detection and Automated Terrain Analysis*

Yannick ALLARD[a] and Alexandre JOUAN[b]
[a] *Senior Associate Member, R&D Department, Lockheed Martin Canada, 6111 Royalmount ave., Montreal, QC, H4P 1K6, Canada*
[b] *Defence Research and Development Canada – Valcartier – Space Optronics Section*

**Abstract.** Several studies have already shown that remote sensing imagery can provide valuable information for area surveillance missions and activity monitoring and that its combination with contextual information could significantly improve the performance of target detection/target recognition (TD/TR) algorithms. The next generation of spaceborne sensors is expected to consist mainly of multi-polarization SAR (PolSAR) and hyperspectral sensors. SAR measurements are sensitive to surface geometry and dielectric constants. In the context of surveillance missions, spaceborne SAR are particularly useful due to their ability to operate day and night under any sky conditions. On the other hand, hyperspectral data are more related to the biochemical state of the observed object and can provide valuable information about the target's surrounding environment and composing material. Therefore, synergistic use of both sensors should refine the image interpretation and scene description by taking advantage of the complimentary nature of the extracted information from the two types of sensors. This paper presents methodologies for change monitoring based on the fusion of contextual features extracted from multi-pass SAR imagery and for automated terrain analysis based on the fusion of features extracted from SAR and hyperspectral data that can be used to derive contextual information such as the target environment. Both proposed methods contain classifiers that are used to build hypotheses on a generic set of attributes (water, wetland, shore, vegetation, fields, man-made objects,…). The hypotheses are further combined using evidence theory.

**Keywords.** Dempster-Shafer evidential reasoning, multisensor fusion, SAR, hyperspectral imagery, object detection, contextual information, land-use mapping

### 1. Introduction

Remote sensing imagery, due to its large spatial coverage, enables the monitoring of large areas. It provides valuable information in the context of area surveillance and activity monitoring. The next generation of sensors that will likely be used for that particular task mainly consist of high-resolution polarimetric SAR (PolSAR), due to their ability to operate under any sky conditions, night or day, and hyperspectral imagery (HSI) because of its high spectral resolution. These two kinds of sensors will provide a large number of data and there is a need to develop tools and methodologies to automatically analyze them and extract meaningful information. While PolSAR measure-

ments are sensitive to surface geometry and dielectric constants, hyperspectral data are more related to the biochemical state of the observed object and can provide valuable information about the target's surrounding environment and composing material. Therefore, synergistic use of both sensors should refine the scene description by taking advantage of the complimentary nature of the extracted information from the two types of sensors.

This paper describes two applications of remote sensing imagery in the context of area surveillance and scene description. Sections 2 and 3 respectively present polarimetric SAR, and HSI data are described along with the features that can be extracted from them. A methodology for area surveillance and change monitoring using multi-temporal single-channel SAR data is presented in section 4 while section 5 presents an automatic procedure for automatic terrain analysis. Finally, conclusions are drawn in section 6.

## 2. Polarimetric SAR Data

The SAR is an active sensor that illuminates an area with an electromagnetic wave in the GHz frequency range (C-band, X-band, P-band, L-band, Ku-band) and forms an image from the radiation scattered back to the antenna. Image quality is degraded by the presence of speckle noise resulting from the interference of coherent radiation. This effect is less important in low resolutions (space-borne) than in high resolutions (airborne) because of spatial averaging. SAR images can be acquired in various modes for local high definition imaging (spotlight) or swath imaging (stripmap). SAR sensors provide a night and day, all weather imaging capability but weather conditions have a critical impact on some features (lakes, rivers and oceans will have a very different aspect under windy conditions than under calm conditions).

The availability of SAR systems that provide information via the use of multiple polarization is rapidly increasing. These systems provide an enhanced capability for terrain analysis because polarimetric data allow the differentiation of scattering mechanisms. This section recalls some of the information that can be extracted from polarimetric SAR imagery for the case of distributed targets, i.e. natural ground covers.

### *2.1. Backscatter Cross Section*

The Backscatter Cross Section (in dB) is computed from the intensity of the co-polarized and/or cross-polarized channel. It is defined as:

$$\sigma_{dB} = \begin{bmatrix} 10\log_{10}(\langle |HH|^2 \rangle) \\ 10\log_{10}(\langle |HV|^2 \rangle) \\ 10\log_{10}(\langle |VV|^2 \rangle) \end{bmatrix}$$

Working in the log domain provides a more robust clustering in the presence of speckle (the speckle noise corruption becomes additive and the power more uniform

across the image) and reduces the differences in absolute magnitude between the co-polarized and cross-polarized channels.

### 2.2. Textural Analysis

The co-occurrence texture features analyze grey-level spatial dependencies [1]. This matrix contains the relative frequencies of all pairwise combinations of backscatter values at a certain distance *d* and direction $\alpha$ within the Area of Interest (AOI).

With $N_g$ grey levels in the image, the dimension of the matrix is $N_g^2$. The (*i,j)* element $C_{ij}$ of this matrix is defined as:

$$C_{ij} = \frac{c_{ij}^{d,\alpha}}{\sum_{i,j}^{L} c_{ij}^{d,\alpha}}$$

where $c_{ij}^{d,\alpha}$ is the frequency of occurrence of grey levels *i* and *j*, separated at a distance *d* and direction $\alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$.

The summation is over the total number of pixel pairs *L*, given *d*, in the window. Many parameters can be computed from this matrix to characterize the ground cover. Some mathematical definitions can be found in [1].

### 2.3. Van Zyl's Classifier

Probably one of the earliest classification processes for polarimetric data, Van Zyl's method provides an indication of the distribution of the main scattering classes present in a scene [2]. An unsupervised classification of scattering behavior is obtained by comparing the Mueller matrix for every pixel in an image with the polarization properties of three simple classes:

1. **even** number of reflections;
2. **odd** number of reflections;
3. **diffuse** scattering.

Considering the general case of scattering by any azimuthally isotropic medium, the symmetry dictates that the ensemble average of the cross-products of the co-polarized (HH and VV) and the cross-polarized (HV) elements of the scattering matrix be zero, and the general form of the Mueller matrix be written as

$$[M] = \begin{bmatrix} \frac{1}{2}\left\langle |S_{hh}|^2 + |S_{vv}|^2 \right\rangle + \left\langle |S_{hv}|^2 \right\rangle & \frac{1}{2}\left\langle |S_{hh}|^2 - |S_{vv}|^2 \right\rangle & 0 & 0 \\ \frac{1}{2}\left\langle |S_{hh}|^2 - |S_{vv}|^2 \right\rangle & \frac{1}{2}\left\langle |S_{hh}|^2 + |S_{vv}|^2 \right\rangle - \left\langle |S_{hv}|^2 \right\rangle & 0 & 0 \\ 0 & 0 & \left\langle \Re(S_{hh}^* S_{vv}) \right\rangle + \left\langle |S_{hv}|^2 \right\rangle & \left\langle \Im(S_{hh}^* S_{vv}) \right\rangle \\ 0 & 0 & -\left\langle \Im(S_{hh}^* S_{vv}) \right\rangle & \left\langle \Re(S_{hh}^* S_{vv}) \right\rangle - \left\langle |S_{hv}|^2 \right\rangle \end{bmatrix}$$

where $\Re$ means real part of, $\Im$ means imaginary part of, the superscript * denotes complex conjugation and $\langle\ \rangle$ denotes ensemble averaging.

The classification process is based on the property that every scattering event adds an 1800 phase shift between the vertically (VV) and horizontally (HH) polarized electric fields of the scattered wave. Details about the classification process can be found in [2].

*2.4. Polarization Synthesis*

The knowledge of the scattering matrix permits the computation of the received power for any possible combination of transmitting and receiving antennas. This process is called the polarization synthesis. It is expressed as [3]:

$$P = \frac{K}{2}\begin{bmatrix} 1 \\ \cos 2\chi_{A_r}\cos 2\psi_{A_r} \\ \cos 2\chi_{A_r}\sin 2\psi_{A_r} \\ \sin 2\chi_{A_r} \end{bmatrix}^T \langle [M] \rangle \begin{bmatrix} 1 \\ \cos 2\chi_{A_i}\cos 2\psi_{A_i} \\ \cos 2\chi_{A_i}\sin 2\psi_{A_i} \\ \sin 2\chi_{A_i} \end{bmatrix}$$

where $[M]$ represents the Mueller Matrix (described earlier).

The angles $(\chi_{A_r}, \psi_{A_r}, \chi_{A_i}, \psi_{A_i})$ are the Stokes parameters representing the rotation angle ($\psi$) and ellipticity angle ($\chi$) characterizing the polarization state of the electromagnetic wave ($0 \le \psi \le \pi, \quad -\pi/4 \le \chi \le \pi/4$).

Surface identification algorithms that rely on that feature are based on the extraction of typical parameters from the polarization signature. Parameters such as the maximum and minimum values of the received power, as well as the pedestal height values, have proven to be useful classification properties [4].

*2.5. H-A-$\alpha$ Classification*

The classification process proposed by Cloude & Pottier [5], Pottier & Lee [6] is an unsupervised approach that aims at estimating the parameters characterizing the "average" dominant scattering mechanism in polarimetric data. This approach is based on the local estimates of the 3x3 coherency matrix formed using pixel averaging over a defined neighborhood as:

$$\langle [T] \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i \cdot k_i^+ = \frac{1}{N}\sum_{i=1}^{N} [T_i]$$

with

$$k = \frac{1}{\sqrt{2}}\begin{bmatrix} S_{HH} + S_{VV} & S_{HH} - S_{VV} & 2S_{HV} \end{bmatrix}^T$$

The dominant scattering matrix is determined from the eigenvector associated with the highest eigenvalue of T and has the degree of randomness or entropy defined from the eigenvalues (in the von Neumann sense) as:

$$H = \sum_{i=1}^{n} -P_i \log_n P_i$$

with

$$P_i = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_i}$$

If the entropy is low, then the observed target can be considered as weakly depolarizing. If the entropy is high, then the observed target is depolarizing. The number of distinguishable polarimetric classes will be higher if the H is low. The dominant scattering mechanism can also be characterized by another scalar descriptor called the anisotropy A, which is defined as

$$A = \frac{\lambda_2 - \lambda_3}{\lambda_2 + \lambda_3}$$

When A=0, the second and third eigenvalues are equal. This may be the case for a dominant scattering mechanism where these two eigenvalues are close to zero, or for a totally random scattering where all three eigenvalues are equal. Finally, the preferred orientation for the scatterer can be estimated by the angle $\beta$ resulting from the parametrization of the three eigenvectors of $T$ introduced as:

$$u = \left[ \cos\alpha \quad \sin\alpha\cos\beta e^{j\gamma} \quad \sin\alpha\sin\beta e^{j\gamma} \right]^T$$

The mean parameters of the dominant scattering mechanism from the coherency matrix are defined by the vector $f$

$$\underline{f} = [\overline{\alpha} \quad \overline{\beta} \quad \overline{\delta} \quad \overline{\gamma}]$$

where

$$\overline{\alpha} = \sum_{i=1}^{3} P_i \alpha_i$$

Cloude has proposed an unsupervised classification scheme based on the use of the two-dimensional H–$\overline{\alpha}$ classification plane where all random scattering mechanisms can be represented. The H–$\overline{\alpha}$ plane is subdivided into nine basic zones characteristic of classes of different behavior.

## 3. Hyperspectral Data

Airborne or satellite imaging spectrometers record reflected solar or emissive thermal electromagnetic energy in hundreds of contiguous narrow spectral bands. The spectrum associated with each pixel of the hyperspectral datacube is usually modeled as a linear combination of the pure material spectral components called endmembers. The estimation of the weight of each spectral component in the resulting spectrum is called spectral unmixing.

By making $R_i$ the reflectance of a given spectrum in the $i^{th}$ spectral band, $N$ the number of endmembers, $R_i^e$ the reflectance of the reference endmember in the $i^{th}$ spectral band, $f_e$ the unknown fraction of endmember *(e)* and $\varepsilon_i$ the error in the ith band of the fit of the N endmembers, the spectral unmixing problem can be written as:

$$R_i = \sum_{e=1}^{N} R_i^e f_e + \varepsilon_i$$

Least-squares techniques are commonly used to estimate the $N$ values of $f_e$ while minimizing the error $\varepsilon_i$. For unconstrained unmixing techniques, $f_e$ is allowed to take any value needed to minimize the error. Partially constrained techniques assume that the sum of all fractions $f_e$ within a pixel must be unity. Both unconstrained and partially constrained unmixing techniques allow positive and negative fractions. The fully constrained condition requires that each fraction have a positive value between 0 and 1. Once all endmembers are found, the data set is unmixed using a constrained linear technique.

When the processing ends, a collection of N spectral endmembers will have been extracted. Two products can actually be generated at the end of the process:

- a pixel-based representation with each pixel of the scene having a set of N fractions representing the contribution of each endmember in the resulting spectrum;
- abundance maps generated for each endmember by attributing to pixels a value corresponding to the endmember fraction.

Both representations can be used to generate pixel-based or feature-based levels of confidence associated with the presence of the extracted endmembers. These descriptions will be useful for fusion with data from another source or from data resulting from running spectral unmixing on another – and consistent – datacube.

Narrow band HSI data offer a portion of the spectra where reflectance and absorption features related to specific crop physical and chemical characteristics can be de-

tected [7]. Many indexes have been designed to detect different physical and chemical properties that can be related to stress conditions or green vegetation presence. These spectral indexes can provide valuable information for automatic scene interpretation and some mathematical definition of such indexes can be found in [8]. An example of applications using spectral indexes for terrain characterization can be found in Section 5.

## 4. Change Monitoring Using Multi-Pass SAR Data

In this example application, a set of multi-date RADARSAT-1 acquisitions with change detectors and textural classifiers is processed. Classifiers are used to build hypotheses on a set of contextual attributes that are combined using evidence theory. This combination strategy provides a powerful information management tool for the fusion of imprecise, incomplete and inaccurate data.

The study site is located around the airport of Stephenville, Newfoundland. In addition to remotely sensed imagery, information layers from a Geographical Information System (GIS) or weather information can be useful sources of information to improve the interpretation of the backscattered signal from the ground. Environmental conditions at the time acquisitions were provided by Environment Canada and the Canadian Hydrographic Service and used as *a priori* information. However, data provided in GIS files may not correspond to the situation on the ground. It may be inaccurate or outdated. For this reason, a preprocessing step based in Markov random fields has been designed in order to make a more efficient use of incomplete GIS files.

### *4.1. Land Use Mapping and Activity Monitoring Using Evidential Fusion*

Textural analysis is performed with a neural network (multi-layer perceptron) trained to classify pixels into contextual classes from a set of eight textural features computed from the GLCM. The number of textural features used as input to the neural network is optimized using genetic algorithms. The output of the neural network is a pixel-based distribution of weights associated to each contextual class. This output can be used to generate a map by giving each pixel the label associated to the larger weight. However, the pixel-based distribution of weights must be kept in memory for further combination with the output of the textural classifier executed on another acquisition. This combination is performed with evidential fusion using Dempster-Shafer theory. Rule-based reasoning is also used to interpret class variations and discriminate between the changes due to acquisition parameters or those due to the apparition of new structures.

The first set of contextual features is the textural features computed from the Grey Level Co-occurrence Matrices (GLCM). The number of gray level for each SAR image was reduced to 11 and the matrices were computed using a 9x9 pixel neighborhood window. For each GLCM, a set of height parameters is computed and used as the input of a neural network trained to classify textures in four different classes namely water, wetland/short grass, vegetation, and man-made objects. From the output of the classifier, two types of analysis can be made.

First, a ***thematic map*** can be built by labeling pixels according to the highest output of the classifier (see figure 1 for an example). Another approach is to build a ***pixel-based declaration*** (in the Dempster-Shafer sense) from the whole set of confidence values. Temporal monitoring can thus be performed through the fusion of such inde-

pendent declarations obtained by the classifier on scenes taken at different times. In order to increase or decrease the level of confidence associated to the classification as well as to improve the interpretation of changes over time, the declarations are combined by using Dempster-Shafer evidential reasoning.

### *4.1.1. Temporal Monitoring Using Dempster-Shafer Theory*

Texture measurements are used to generate a declaration characterizing the texture. Declarations are made of the four texture propositions: WATER, WETLAND, VEGETATION, MAN-MADE OBJECT with a mass associated to each of them. These masses represent the confidence level that the corresponding pixel actually belongs to its associated class. The combination of the texture declarations obtained from multitemporal data leads to the generation of compound hypotheses that may be used to interpret changes. For example, the shore class will appear as water (submerged) in some images (high tide), and as wetland in others, and the result of the fusion process classifies these changes over time as temporary submerged areas. The same strategy applies for the detection of ships resulting from the fusion of a man-made object detected over an area previously classified as water.

Once the fusion is ended, the resulting pixel-based declaration of hypotheses is stored to later be combined with the newly generated texture declaration. A thematic map can also be obtained by labeling pixels according to the proposition associated to the highest mass.

Figure 1 shows the city of Stephenville, its airport and Port Harmon harbor. The upper and lower left maps are the result of the classification that is fused in the map on the right. On the fused map, a ship is detected docked on a pier located at the east side of the harbor. In the center of the image, the airport is labeled as water, due to the fact

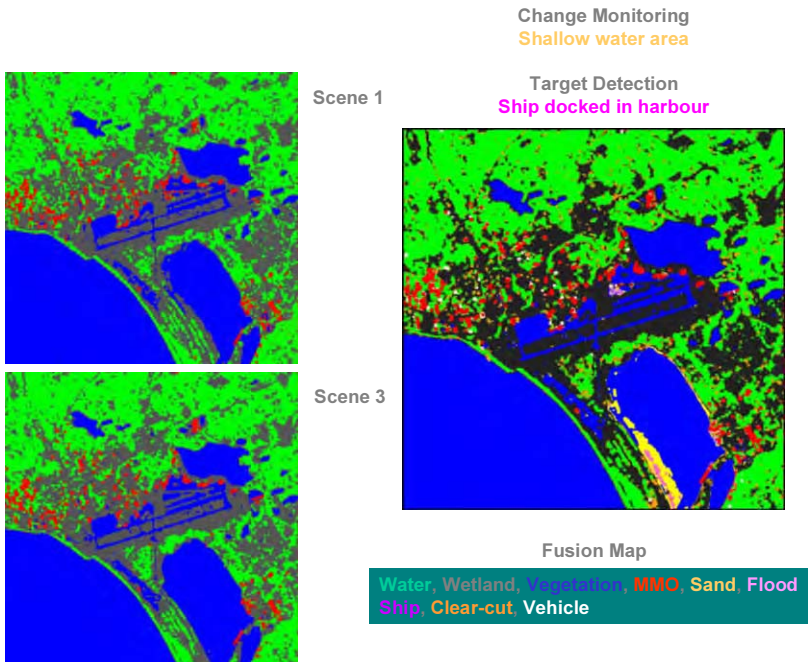



**Figure 1.** Change monitoring and target detection using evidential fusion.

that its texture is very similar to the texture of the water. Thus, the classifier detected a "ship" on the airport, which is probably a man made structure that has been temporarily placed on the airport. This misclassification can be corrected by correlating the results with *a priori* information from GIS files. We can also notice the large zone detected as sand in the west of the bay area. This is confirmed by the water level, which is 0.607 m in Scene 1, June 14th 1998, and 0.982 m in Scene 3, May 25th 1998. There is a 20 cm difference between the 2 images, which explains the submerged area in the Scene 3 image.

This section presented an application of coastal monitoring/target detection using evidential fusion. The results obtained so far are very promising, and can easily be adapted to other monitoring applications using multi-temporal imagery. In the next section, we will describe an application for automated terrain analysis using multisensor imagery.

## 5. Multisensor Fusion for Background Characterization

Lockheed Martin Canada has developed the Intelligent Data Fusion System (IDFS) based on the evidential fusion of features extracted from polarimetric SAR (PolSAR) and Hyperspectral (HSI) imagery. Polarimetric SAR provide hypotheses about the likelihood that some object of interest may be present in the scene based on textural and scattering properties of the analyzed surface. On the other hand, end-member selection techniques provide a set of pixel-based hypotheses reflecting the likelihood that some typical material may be present in the scene based on the spectral properties of the analyzed surface. Hypotheses provided by each source of information represent an incomplete, inaccurate and imprecise description of reality. A high level diagram of IDFS can be found in [9].

The data sets used for this example were acquired over Indian Head (SK, Canada) by CCRS on June/July 1999. The data set consists of C-band polarimetric SAR and Probe-1 HSI imagery. The selected portion of imagery for co-registration contains features such as water, roads and building structures. A rough co-registration procedure involving a simple rotation, scale and translation (RST) warping technique (ENVI) was used and tie points were selected manually. Co-registration accuracy was estimated to be 4.9 pixels but is most likely greater where localized deformities in the HSI data are present. Figure  shows the area of interest from the Probe-1 HSI datacube (left) and its roll-corrected and co-registered version (right).

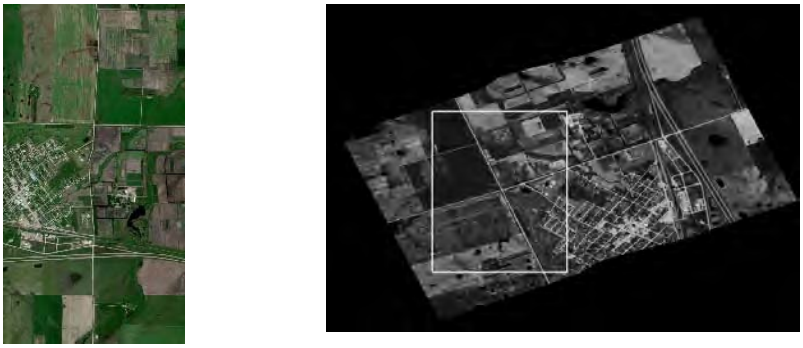


**Figure 2.** Line 2 Indian Head (SK) 2000 (left :HI, right: roll-corrected and co-registered with PolSAR).

### 5.1. Data Fusion

The fusion of the declarations provided by the SAR and HSI features is performed using the Dempster-Shafer evidence theory. This framework offers a convenient way to include in the reasoning process imprecision and uncertainties. Since the operators used in this study are not able to distinguish all the different classes in a precise way, the capability of the evidence theory to take into account compound hypotheses is essential.

#### 5.1.1. Focal Elements and Mass Functions Definition

For each operator, we have to define focal elements on which this operator will give support during the fusion step. The choice of the focal elements and the mass functions has been done in a supervised way, using the knowledge about the information provided by each operator. In order to derive mass functions, training samples have been selected. In this example, mass functions are built from the histograms of occurrence values by fitting trapezoidal functions.

In that particular case, the unsupervised classification method proposed in [5] and described earlier couldn't be used properly. On C-band data the entropy of the eigenvalues is usually higher than for P- or L-band data (where volume scattering is more important). Rough surface scattering dominates on C-band (on our test data) and classes of interest are less distinguishable and exhibit poor polarimetric variations in the entropy-alpha plane. Only bare surfaces can be distinguished with enough confidence on our test data using this decomposition.

### 5.2. Synergistic Use of SAR and HSI Data for Terrain Analysis

If the available SAR data is only single polarization, Figure 3 shows a simple tree of propositions that may be produced by the fusion of the output of textural classifiers with the output of the spectral indices. This taxonomy tree is used here only to illustrate the typical output of the developed system, but the user has the possibility to build his own interpretation flowchart.

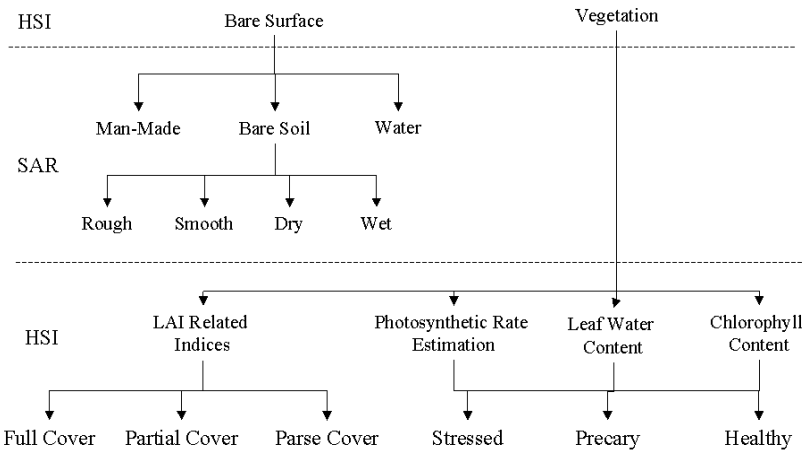



**Figure 3.** Propositions generated by the fusion of textures (SAR) and spectral indices.

The goal of fusion is to increase the belief in a given hypothesis or to lower the number of hypotheses. A rough description of the scene can be obtained using the HSI vegetation indices. They provide initial generic declarations about the type of surface (bare surface or vegetation). Then, the "bare surface" proposition can further be refined into three subtypes, namely "man-made object," "water" and "bare soils" using the SAR features. Figure 4 shows the land cover description before (left) and after (right) the fusion of the SAR backscatter information with the HSI vegetation indices.

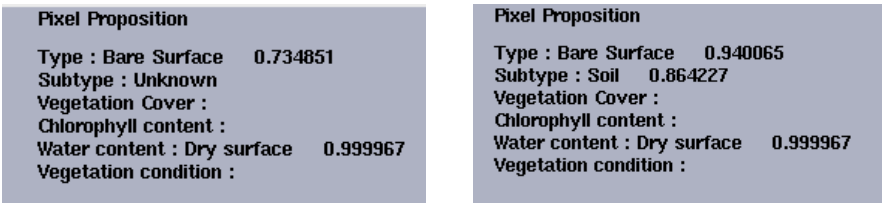



**Figure 4.** Description of the land cover using generic propositions produced by spectral vegetation indices (left); refinement after fusion with propositions produced by polarimetric backscatter (right).

The declaration provided by the vegetation indices can also be used to determine other vegetation properties. Figure 5 shows the map of vegetation density (left) with the associated belief map (right) resulting from the numerical fusion of the spectral features. Bare surfaces are in black and therefore don't have any confidence value in the belief map since this attribute is not computed for that type of surface.

**Figure 5.** Map of the vegetation's density and associated belief.

Many parameters are used to evaluate the condition of the vegetation. The chlorophyll content is computed using the ratio TCARI/OSAVI as proposed in [10], the Photochemical Reflectance Index and the leaf water content are used since they are related to the vegetation's condition. Figure 6 is an example of outputs that can be obtained for a vegetated area. Two water related spectral indices are also used to evaluate the leaf water content, the WBI and NDWI, while SAR backscatter provides information on soil humidity when bare surfaces have been previously discriminated.

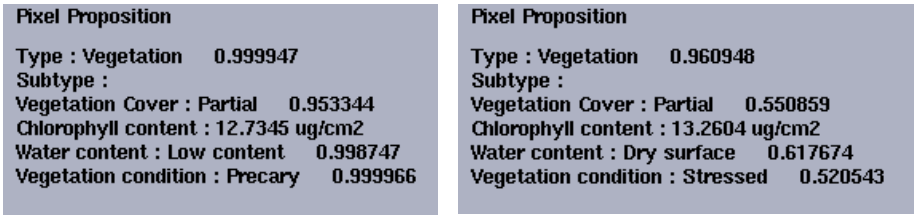



**Figure 6.** Partial cover, precary condition (left), partial cover, stressed vegetation (right).

### *5.3. Discussion*

The fusion of HSI indexes and PolSAR features provides an easy way to make a generic interpretation of the scene since both are related to different parameters of the illuminated surface. The introduction of end-members in the fusion process should improve the scene description. It must be stated that Cloude's decomposition doesn't perform well on C-band SAR data (at least for our dataset), most of the scene being dominated by rough surface scattering. The use of P- or L-band SAR data would be more efficient because of the increasing contribution of volume scattering, which should be reflected by a greater differentiability of the classes in the polarimetric space.

## 6. Conclusions

Applications of remotely sensed imagery for area monitoring and terrain analysis were presented. Examples show that this source of information is useful for those particular tasks. In the near future, spaceborne platforms will deliver a great amount of data of very different modalities. Therefore, there is a need for tools that automatically analyze the data and provide support to the image analyst. But one should keep in mind that all the needed information can't be extracted from the data and that the use of contextual information is very important in order to provide a consistent interpretation of the scene.

## References

[1] R. Haralick, K. Shanmugan, I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, p. 610, 1973.

[2] J. Van Zyl, "Unsupervised Classification of scattering behavior using radar polarimetry data," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 27, no. 1, p. 36, 1989.

[3] F.T. Ulaby, C. Elachi, "Radar Polarimetry for Geoscience Applications," Artech House, 1990.

[4] H. McNairn, J.C. Deguise, J. Secker, "Development of Remote Sensing Image Products for Use in Precision Farming," *Third European Conference on Precision Farming*, Montpellier, France, June 18–20, 2001.

[5] S.R. Cloude, E. Pottier, "An Entropy Based Classification Scheme for Land Applications of Polarimetric SAR," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 35, pp. 68–78, 1997.

[6] E. Pottier, J.S. Lee, "Application of the H-A-α Polarimetric Decomposition Theorem for Unsupervised Classification of Fully Polarimetric SAR Data Based on the Wishart Distribution," in *Proc. Committee on Earth Observing Satellites SAR Workshop*, Toulouse, France, Oct. 26–29, 1999.

[7] I. Strachan, E. Pattey, J. Boisvert, "Impact of nitrogen and environmental conditions on corn as detected by hyperspectral reflectance," *Remote Sensing of Environment*, No. 80, pp. 213–224, 2002.

[8] J.A. Gamon, H.-L. Qiu, D.A. Roberts, S.L. Ustin, D.A. Fuentesl, A. Rahman, D. Simsl, C. Stylinskic, "Water expressions from hyperspectral reflectance: implication for ecosystem flux modeling," *Proceedings of the AVIRIS workshops,* 1999.

[9] A. Jouan, Y. Allard, J. Secker, A. Beaudoin. E. Shahbazian, "Intelligent Data Fusion System (IDFS) for Airborne/Spaceborne Hyperspectral and SAR Data Analysis," *Proc. of the Workshop on Multi/ Hyperspectral Technology and Applications*, Redstone Arsenal, Huntsville, Alabama, 2002.

[10] D. Haboudane, J. Miller, N. Tremblay, P. Zarco-Tejada, "Combining hyperspectral vegetation indices for a better estimation of leaf chlorophyll content in corn canopies," *Proc. International Symposium on Spectral Sensing Research,* 2001.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Fusion of Two Imagery Classifiers

## *A Case Study*

Hugues DEMERS
*Lockheed Martin Canada*

**Abstract.** Two image classifiers are fused to improve classification performances of infrared ship images. A moment based classifier is fused to a template based classifier. Two methods for combining classifiers are studied, the product rule of combination and the Dempster rule of combination for evidences. Emphasis is placed on the methodology used and the analysis of each subprocess' impact on overall performance.

**Keywords.** Classifier combination, Demspter-Shafer theory, visual perception segmentation, deformable templates, shape, FLIR imagery

### 1. Introduction

The fusion of classifiers is by now recognized as a good way of improving classification performances. By combining all individual opinions, a consensus decision can be made which in some combination scheme consistently outperforms a single best classifier. Various combination schemes have been proposed over the years in the literature (see [1,2] for an overview). They differ in their architectures, the characteristics of the combiner, and selection of the individual classifiers.

As exemplified by Kittler [3], there are two scenarios for combining an ensemble of classifiers. In the first scenario, all classifiers use the same representation of the input pattern. This is, for example, an ensemble of neural networks trained with different initial values. In the second case, each classifier uses its own representation of the input pattern, i.e. the measurements extracted by each classifier are unique to each. In the first case, each classifier can be considered to produce an estimate of the same a posteriori class probability, but not in the second case.

The output information of various classification algorithms can be divided into three levels with increasing details for each [3].

- the abstract level in which a classifier only outputs a unique class label;
- the rank level (or cardinal) for which the classifier ranks each class label in a list with the top one being the most probable one;
- the measurement level (or ordinal) where a measure is given for each label that addresses the degree that the given input vector has that label.

The last level is the one that gives the most details about the second choices made by the classifiers. The second choices are important because accuracy improvement will generally occur when more information is taken into account than just a unique label assigned by a classifier. This is especially true in the case of a small ensemble. In

that case, the improvement coming from the statistical combination (e.g. majority voting) of the output of many abstract-level classifiers is not effective. The additional information must come from secondary choices. The treatment of the output of a small ensemble of classifier has to take this into account.

The experiment described in this paper deals with Forward Looking Infrared (FLIR) ship images and their classification. The images used are from the United States Naval Airfare Center, China Lake, and were provided by Dr. Sklansky of the University of California at Irvine. They were digitized to a resolution of 256×64 with 8 bits per pixel. Each image can be classified into one of eight types: destroyer (DT), container (CT), civilian freighter (CF), auxiliary oil replenishment (AOR), landing assault tanker (LAT), frigate (FR), cruiser (CR) and destroyer with guided missile (DGM). Two sets of images are provided, one taken at 30° and the other at 90° of relative heading. Only the latter set of images is used here.

This paper presents how two different classifiers can be combined at the measurement level to improve decision accuracy. The system was built with complete automation in mind, i.e. no human intervention. This constraint was particularly difficult to implement for the segmentation process as the signal-to-noise ratio (SNR) covers a wide range of values.

In the next section, each classifier, from segmentation to classification, are presented. Section 3 gives details about different ways of fusing the two classifiers. Performance analysis is presented in section 4. Finally, conclusions are drawn in section 5.

## 2. Presentation of Classifiers

In this section the algorithms used by the different classifiers are presented. The usual steps in a classification process are segmentation, features extraction and classification proper. The segmentation step is identical for both classifiers. The first classifier uses moments to describe the segmented ship and the second is template based. The goal is to have a second classifier that complements the first one. Where the moment based classifier has poor performances, the template based one should excel.

### *2.1. Segmentation*

The segmentation process, which is identical for both classifiers, uses a biologically motivated algorithm [4]. The visual perception segmentation (VPS) process is based on human brightness perception and foveal adaptation. It offers a simplified description of the main features of the human visual system: transformation of illuminance into a psychological stimulus, brightness adaptation, and the existence of a minimal perceptible contrast. To simulate the biological accommodation, an artificial fovea centralis is moved through the image in order to calculate a contrast ($C$) and a minimum perceptible contrast ($C_{A,min}$) once for every pixel. If $C$ is greater than $C_{A,min}$, a pixel belongs to the object. This segmentation method is very fast and in our case very effective for a wide range of image SNR. For further details about this segmentation process the reader is referred to [4].

### 2.2. Moment Based Classifier

This classifier uses moments presented in [5]. It makes use of the Dempster-Shafer theory of evidence to fuse expert proposition sets built from the frequency distribution of moments. Fourteen moments are used to describe the segmented image. Half of these moments are structural and the other half are intensity-based. The idea behind the use of intensity-based moments is to use all the information provided by an infrared image and not just shape information.

#### 2.2.1. Feature Extraction

To compute the moments, the segmented ship is partitioned into seven equal sections along the *x*-axis and into two sections along the *y*-axis delimited by a centroid point defined below. The structural moments are computed on the part of the segmented ship that is above the centroid, i.e. on the discriminant part which is above the hull. The cendroid used here takes into account the intensity of each pixel and is defined as:

$$\overline{x} = \frac{1}{\sum_k (I_k - c)} \sum_l x_l (I_l - c)$$

$$\overline{y} = \frac{1}{\sum_k (I_k - c)} \sum_l y_l (I_l - c)$$

where $I_l$ is the gray-level intensity of pixel $l$ and $c$ is a constant. The following moment

$$\mu^i = \frac{1}{LN_i^+} \sum_{l \in S_i^+} (y_l - \overline{y})$$

is computed for each of the seven parts of the segmented ship. Here $i$ is the ship section number, $L$ is the ship length, $N$ is the number of pixels in $S_i^+$, the $i$th section above the centroid point. These are the structural moments. The intensity-based moments are given by

$$v^i = \frac{1}{\sum_l (I_l - c)} \sum_{l \in S_i^+} (I_l - c) .$$

For each section $i$, the intensity of each pixels is summed up and divided by the ship's intensity.

#### 2.2.2. Classification

By computing moments over a test sample of segmented ships a knowledge database is built, from which frequency distributions are extracted. These frequency distributions are used to compute a probability of occurrence for each combination of moments and classes of ship of an unknown feature vector. From these probabilities of occurrence,

expert opinions in the form of proposition sets are computed where a particular proposition corresponds to the probability that a given moment represents a given class. Given an unknown feature vector, 14 moments combined with 8 classes gives 14 proposition sets of 8 propositions each. The Dempster-Shafer theory of evidence is used to combine the 14 expert opinions. The output is a belief score for each ship's class.

### *2.3. Template Based Classifier*

A very different classifier based on templates is chosen to complement the preceding one. It uses a shape matching algorithm to compute a distance from each template and thus classify an unknown ship. To achieve this, a contour is extracted from the segmented image and shape descriptors are calculated on this contour. Using those shape descriptors, points of the unknown ship are assigned to corresponding points of a template and a mapping transformation is computed from this assignment. This transformation is used to achieve translation, rotation and scale invariance. A distance measure between the unknown input and each available template is calculated and used as a classification basis. The shape descriptor and matching algorithm used are detailed in [6] and briefly described below.

#### *2.3.1. Shape Descriptors*

Features extraction here consist of computing shape descriptors that will be used for template matching. These descriptors are called shape contexts. In this approach, a shape is essentially captured by a subset of points of the external and/or internal contours. The subset is chosen as uniform spaced points of the contour, i.e. a downsampled contour. A shape context is computed for each point of the subset. The ensemble of shape context represents the reduced object information needed for matching.

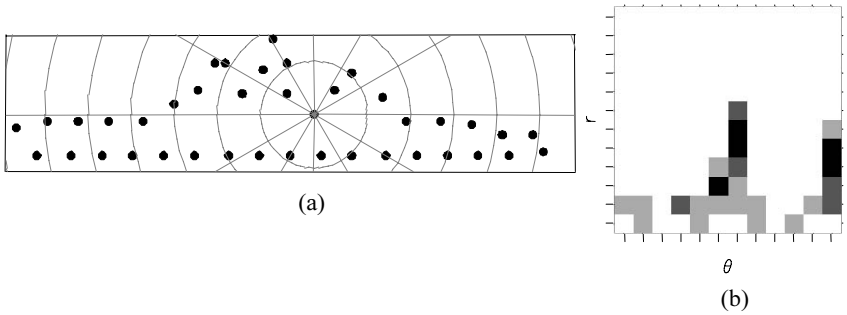
A shape context corresponding to a point $p_i$ is the spatial distribution of all other points $q_i$ about point $p_i$. In other words, for a point $p_i$ on the shape, a coarse histogram $h_i$ of the relative coordinates of the remaining $n$–1 points is computed,

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \text{bin}(k)\}.$$

An example is shown in Fig. 1. The cost of matching two points, $p_i$ from the first shape and $q_i$ from the second shape, is denoted $C_{ij} = C(p_i,q_j)$. The $\chi^2$ test statistic is used

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{K}\frac{\left[h_i(k) - h_j(k)\right]^2}{h_i(k) + h_j(k)}$$

where $h_i(k)$ and $h_j(k)$ denote the $K$-bin normalized histogram at $p_i$ and $q_j$, respectively.




(a) (b)

**Figure 1.** Shape context computation. (a) Overlaid histogram bins used in computing the shape context of the center point. (b) Graphical representation of the shape context associated with the center point of (a).

The assignment problem is solved by minimizing the total cost of matching, given a set of costs $C_{ij}$ between all pairs of points $p_i$ on the first shape and $q_j$ on the second shape

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)}),$$

subject to the constraint that the matching be one-to-one, that is $\pi(i)$ is a permutation. This is a linear assignment problem solved by the Jonker-Volgenant-Castanon algorithm [7]. The input is a square cost matrix with entries $C_{ij}$ and the result is a permutation assigning one point $p_i$ of the first shape to one and only one point $q_i$ of the second shape.

*2.3.2. Classification*

Having a one-to-one correspondence between two shapes, a plane transformation is estimated to map arbitrary points from one shape to the other. By mapping all points of a template onto the unknown ship, translation, rotation and scaling invariance are achieved. A function known as a thin plate spline is used to estimate the coordinate transformation. This is the 2D generalization of the cubic spline. See [8] for further details. With the template transformed onto the unknown ship, it is possible to estimate a distance measure between the two. The measure used is the symmetric sum of the minimum Euclidian distance over a subset of best matching point of two shapes $P$ and $Q$, i.e.,

$$D_{sc}(P,Q) = \frac{1}{n}\sum_{p\in P} \arg\min_{q\in Q} \|p - T(q)\| + \frac{1}{m}\sum_{q\in Q} \arg\min_{p\in P} \|p - T(q)\|$$

where T($q$) denotes the estimated thin plate spline shape transformation of point $q$, $n$ and $m$ are the number of points of shape $P$ and $Q$ respectively. Classification is achieved by calculating such a distance for each template to the unknown ship. The smallest distance indicates that the corresponding template is the most likely class candidate for the unknown input.

### 2.4. Results

The confusion matrix of the moment based classifier is given in Table 1. The figures represent the accuracy defined as the number of ships correctly classified over the total number of ships of that category. For some ship categories the performances are very poor. The True Acceptance Rate (TAR) is given as the number of ships correctly classified in a given category over the total number of ships misclassified into that category. It represents the confidence that the output really is what the classifier claims it is.

**Table 1.** Confusion matrix for the moment based classifier. Figures are percentages. Overall accuracy is 75.5%

| | DT | CT | CF | AOR | LAT | FR | CR | DGM |
|---|---|---|---|---|---|---|---|---|
| DT | 78.9 | 3.6 | 2.1 | 0.4 | 1.5 | 6.0 | 5.2 | 2.4 |
| CT | 4.9 | 79.2 | 11.5 | 2.2 | 0.9 | 0.9 | 0.4 | 0.0 |
| CF | 1.6 | 12.0 | 85.3 | 0.5 | 0.5 | 0.0 | 0.0 | 0.0 |
| AOR | 1.7 | 2.2 | 11.7 | 84.4 | 0.0 | 0.0 | 0.0 | 0.0 |
| LAT | 14.4 | 5.3 | 9.0 | 1.1 | 67.0 | 2.7 | 0.0 | 0.5 |
| FR | 27.2 | 9.8 | 4.9 | 0.0 | 0.0 | 50.0 | 7.1 | 1.1 |
| CR | 12.7 | 0.6 | 0.3 | 0.6 | 0.3 | 8.9 | 76.3 | 0.3 |
| DGM | 24.2 | 1.7 | 2.1 | 0.8 | 1.3 | 5.1 | 1.7 | 63.1 |
| TAR | 75.3 | 66.1 | 56.1 | 95.9 | 87.5 | 50.0 | 80.9 | 87.1 |

The overall accuracy of the template based classifier is lower than the moment based one, 73.1% compared to 75.5%. Although the overall appearance of the confusion matrix resembles the previous one, closer inspection shows that in many cases, complementarity is achieved. Analysis of per category performances indicates that both classifiers exhibit the same behavior with increasing ship distance, i.e., diminishing number of pixels in the segmented image. This means that no classifier is more robust to the disappearance of features with distance.

**Table 2.** Confusion matrix for the template based classifier. Figures are percentages. Overall accuracy is 73.1%

| | DT | CT | CF | AOR | LAT | FR | CR | DGM |
|---|---|---|---|---|---|---|---|---|
| DT | 60.3 | 2.1 | 0.7 | 4.9 | 2.9 | 7.4 | 9.4 | 12.2 |
| CT | 0.0 | 95.6 | 0.9 | 0.4 | 3.1 | 0.0 | 0.0 | 0.0 |
| CF | 1.1 | 31.7 | 51.9 | 12.0 | 3.3 | 0.0 | 0.0 | 0.0 |
| AOR | 0.0 | 0.0 | 0.2 | 99.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| LAT | 5.3 | 2.7 | 0.5 | 13.8 | 72.9 | 2.1 | 1.1 | 1.6 |
| FR | 5.4 | 3.3 | 0.0 | 3.3 | 6.5 | 61.4 | 9.8 | 10.3 |
| CR | 7.3 | 0.3 | 0.0 | 1.3 | 7.3 | 12.1 | 69.5 | 2.2 |
| DGM | 12.3 | 0.0 | 0.0 | 3.0 | 6.4 | 1.3 | 1.7 | 75.4 |
| TAR | 86.0 | 71.5 | 91.3 | 80.2 | 61.7 | 52.8 | 69.7 | 59.5 |

## 3. Fuser Description

T.K. Ho [2] pointed out two strategies for designing multiple classifier systems, namely decision optimization methods and coverage optimization methods. In the former case, the classifiers are given and unchangeable, they are considered as already optimized for the task, even thought they might not be. The goal is then to optimize the combination function in the hope of making the best out of what the classifiers offer. In the case of coverage optimization methods, the combination function is fixed and unchangeable in form. The strategy is to create a set of classifiers that will complement each others and will yield the best final decision under the chosen combination function.

Ho [2] summarized the best known decision combination methods under joint consideration of two factors: the possibility of training the classifiers and the level of information provided by the classifiers, i.e. unique, ranked or measurement level. Another consideration exemplified by the experiment described in this paper would be the number of classifiers forming the ensemble. As this number diminishes, the optimization focus must shift from the combination function to the coverage, because relying on a statistical combination of poorly performing classifiers is not possible in that case. With a small number of classifiers the impact of one being wrong is strongly felt on the final decision. So, performances of each individual classifier must be optimal, and uniform coverage of the input patterns must be achieved. Furthermore, to have accuracy improvement with a small number of classifiers the secondary choices of each classifier have to be considered. In our experiment, when both classifiers are wrong, but the secondary choices are the right ones, the combination scheme must allow to have the correct final decision. This become possible when the output of each classifier is a confidence or a distance measure of some sort and the combination function takes all outputs into consideration.

Two combination schemes were investigated: the product rule and the Demspter-Shafer theory of evidence. The next section presents the classifiers' output processing necessary for combination and the fuser schemes are presented in the following sections.

### *3.1.1. Classifier Output Processing*

The output of the template based classifier is a distance measure which must be transformed into a pseudo confidence level. The following function was used to map the output of a distance function to the range [0,1]:

$$f(x) = e^{-x^2}$$

Following the work of Xu et al., [9], information from the confusion matrix of the test sample was used to weight the output of each classifier. The confidence measure assigned to class label $C_i$ was weighted by the corresponding true acceptance rate.

### *3.1.2. Product Combination Rule*

This rule of combination quantifies the likelihood of the input pattern being assigned a class label, $C_i$, by combining the *a posteriori* probabilities generated by each individual classifier for that particular class label.

Consider a pattern recognition problem where $R$ classifiers are to give their opinion on the possible membership to $M$ classes of an unknown input pattern $x$. Under the condition of statistical independence, the Bayesian decision rule can be modified into the product decision rule [3]. This rule states that $x$ is assigned to class $C_i$ provided that the following is maximum:

$$\max_{i=1}^{M}\left[P^{-(R-1)}(C_i)\prod_{k=1}^{R}P_k(x\in C_i)\right]$$

where $P(C_i)$ is the *a priori* probability of occurrence and the $P_k$ are the *a posteriori* probabilities. The condition of independence here is a good approximation considering that each classifier uses its own internal representation of the input pattern. The weighted confidence levels of each classifier are taken as *a posteriori* probabilities.

### *3.1.3. Dempster Rule of Combination*

The Dempster-Shafer theory of evidence has been used previously to combine the output of multiple classifiers [9]. The technique is more robust than Bayesian approaches because the uncertainty of each classifier is taken into account, although a test sample is necessary to determine it.

The masses or basic probability assignments are taken from the weighted measures of each classifier. The ignorance is taken as the accuracy of each classifier for the particular ship category under consideration. This differs from Xu's approach, where the classifiers' output is of the abstract type. Here, all the information provided by the classifiers is used.

## *3.2. Results*

The confusion matrices are presented in Tables 3 and 4. The overall accuracy of each fuser is product 80.8% and D-S 80.5%. An improvement of 5 percent points over the best classifier is observed.

**Table 3.** Confusion matrix for the product fuser. Overall accuracy is 80.8%

| | DT | CT | CF | AOR | LAT | FR | CR | DGM |
|---|---|---|---|---|---|---|---|---|
| DT | 80.0 | 3.2 | 0.8 | 2.8 | 0.5 | 1.9 | 7.6 | 3.3 |
| CT | 0.0 | 91.2 | 4.9 | 3.1 | 0.9 | 0.0 | 0.0 | 0.0 |
| CF | 0.5 | 18.6 | 76.0 | 4.9 | 0.0 | 0.0 | 0.0 | 0.0 |
| AOR | 0.0 | 1.0 | 4.5 | 94.5 | 0.0 | 0.0 | 0.0 | 0.0 |
| LAT | 12.8 | 2.1 | 6.4 | 4.8 | 72.9 | 0.0 | 0.0 | 1.1 |
| FR | 23.9 | 4.3 | 1.6 | 2.7 | 0.5 | 56.5 | 9.2 | 1.1 |
| CR | 7.0 | 0.3 | 0.0 | 1.0 | 1.0 | 6.0 | 83.8 | 1.0 |
| DGM | 22.0 | 0.8 | 0.4 | 0.8 | 0.4 | 0.8 | 0.4 | 74.2 |
| TAR | 80.8 | 72.8 | 72.8 | 87.6 | 92.6 | 74.8 | 77.9 | 84.5 |

**Table 4.** Confusion matrix of the Dempster-Shafer fuser. Overall accuracy is 80.5%

| | DT | CT | CF | AOR | LAT | FR | CR | DGM |
|---|---|---|---|---|---|---|---|---|
| DT | 80.0 | 2.5 | 1.1 | 3.4 | 0.4 | 1.9 | 8.0 | 2.8 |
| CT | 0.9 | 89.4 | 4.4 | 3.5 | 1.8 | 0.0 | 0.0 | 0.0 |
| CF | 0.5 | 19.1 | 75.4 | 4.9 | 0.0 | 0.0 | 0.0 | 0.0 |
| AOR | 0.0 | 0.5 | 2.2 | 97.5 | 0.0 | 0.0 | 0.0 | 0.0 |
| LAT | 12.8 | 1.6 | 7.4 | 5.9 | 70.2 | 0.5 | 0.0 | 1.6 |
| FR | 27.2 | 4.9 | 2.7 | 2.7 | 0.5 | 50.0 | 11.4 | 0.5 |
| CR | 7.3 | 0.3 | 0.0 | 2.2 | 0.3 | 3.5 | 85.1 | 1.3 |
| DGM | 20.8 | 1.3 | 0.4 | 0.8 | 0.8 | 1.3 | 0.8 | 73.7 |
| TAR | 80.2 | 72.7 | 74.6 | 85.7 | 91.3 | 76.0 | 76.4 | 85.7 |

## 4. Performance Evaluation

Performances are evaluated in the context of a completely automated process. This constraint is a very stringent one particularly for the segmentation part. However, it makes possible the objective study of a complete reconnaissance system and the interaction of its internal parts.

The segmentation algorithm has to deal with images varying over a large range of SNR. The parameters of the algorithm have been adjusted for a typical looking image. No automatic adjustment of the segmentation based on general characteristics of the image have been implemented and no pre-processing of the images is done. As ship images get smaller (i.e. the distance increases between the observer and the ship) the SNR diminishes and discriminant features disappear. The latter effect cannot be overcome. However, the lower SNR can be compensated by an appropriate segmentation and pre-processing. This would lead to an important performance improvement.

The moment based classifier uses features based on the brightness of the pixel. The analysis of the probability of occurrence shows that the discriminative power of these attributes is not significant. However, the low resolution images of the database preclude their utility. High-resolution images should be tested. The use of Zernike's moments, Hu's moments and other geometric moments should be investigated. Along with new moments, the implementation of a feature selection algorithm would also be beneficial. This not only reduces the burden of the recognition process by reducing the number of features, but in some cases it can also provide a better classification accuracy [10].

Analysis of the accuracy per number of pixels of the segmented ship shows that the template based classifier generally performs better for ships near the observer, i.e. with a large number of pixels. This is caused mainly by the choice of templates. They were built from high-definition images with many details. A multi-resolution set of templates could improve performances particularly if the distance to the ship was known. This would generally be the case if the ship is already tracked by a radar.

Regarding the template matching process, a circular preserving order algorithm was implemented but deemed too CPU-intensive for the added benefits. This algorithm preserves the circular order of points on both templates, which helps in dealing with outliers.

This experiment was conducted on images of ships seen at 90° at sea level. In a real-life scenario any angle would be possible. Both classifiers would have to be modified to take into account a change in angle lookdown (in the case of an airborne

imager) and relative heading. When no prior information is available, from a tracking radar for example, the search space would be very large and performances would suffer. However, a real-life scenario would involve other sensors from which the distance and heading would be known. This would reduce the problem of an all-aspect ship recognition to the one presented in this paper, namely a simple aspect classification.

Even thought the product rule of combination performs slightly better, the Dempster rule of combination is the preferred method here. It is more robust in the sense that the state of unknown information is represented by the ignorance.

It is obvious from what was presented that the combining function is not the primary target of optimization. The ensemble of classifiers has to be improved first. It is a case of coverage optimization instead of combining function optimization.

## 5. Conclusions

A two classifier automatic recognition system has been presented which classifies into eight categories FLIR ship images seen at 90°. A moment based and a template based classifier provide confidence measures to a fuser. Two combination functions were studied: the product rule of combination and the Dempster combination rule. An overall classification accuracy of 80.5% is achieved by the Dempster fuser, which is better than the best classifier.

This experiment exemplifies the fact that with a small number of classifiers, each one of them must output secondary choices with an associated confidence level. The combining function must use all of this information in order to improve results. Accuracy improvement comes primarily from the optimization of the ensemble of classifiers and not from the combining function.

## References

[1] K. Jain, R.P.W. Duin, J. Mao, "Statistical pattern recognition: a review," IEEE PAMI, vol. 2, no. 1, pp. 4–37, Jan. 2000.
[2] T.K. Ho, "Multiple Classifier Combination: Lessons and Next Steps," in Hybrid Methods in Pattern Recognition, eds. A. Kandel, H. Bunke, 2002.
[3] J. Kittler, R.P.W. Duin & J. Matas, "On Combining Classifiers," IEEE PAMI, vol. 20, no. 3, pp. 226–239, Mar 1998.
[4] L. Heucke, M. Knaak & R. Orglmeister, "A New Image Segmentation Method Based on Human Brightness Perception and Foveal Adaptation," IEEE Signal Processing Letters, vol. 7, no. 6, pp. 129–131, Jun 2000.
[5] S. Allen, "Signal Based Features with Applications to Ship Recognition in FLIR Imagery," in Proceedings of 4th Annual Conference On Information Fusion, vol. 2, p. FrC2–3, 2001.
[6] S. Belongie, J. Malik & J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," IEEE PAMI, vol. 24, no. 24, pp. 509–522, Apr 2002.
[7] O.E. Drummond, D.A. Castanon & M.S. Bellovin. "Comparison of 2-D Assignment for Sparse, Rectangular, Floating Point, Cost Matrix," Journal of the SDI Panels on Tracking, Institute for Defense Analyses, no. 4, pp. 4–81 to 4–97, 1990.
[8] F.L. Bookstein, "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations," IEEE PAMI, vol. 11, no. 6, pp. 567–585, Jun 1989.
[9] L. Xu, A. Krzyzak and C.Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition," IEEE SMC, vol. 22, no. 3, pp. 418–435, May/Jun 1992.
[10] S.J. Raudys and A.K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," IEEE PAMI, vol. 13, no. 3, pp. 252–264, Mar 1991.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Application of Multi-Dimensional Discrete Transforms on Lie Groups for Image Processing

Ashot AKHPERJANIAN[a], Armen ATOYAN[b],
Jiri PATERA[b] and Vardan SAHAKIAN[a]
[a] *Yerevan Physics Institute, Armenia*
[b] *CRM, Université de Montréal, Canada*

**Abstract.** We are developing methods of discrete Fourier transforms (hereafter DFT) of discrete functions produced from the sampling of continuous functions on the nodes of multi-dimensional grids in the fundamental regions F of compact semisimple Lie groups. Depending on the rank and the symmetry intrinsic to the chosen group, discrete Fourier transforms of functions given on the multi-dimensional grids of different symmetries are possible. Implementation of this method (abbreviated as DGT for Discrete Group Transform) in the one-dimensional case corresponding to the group SU(2), formally results in the transform known as Discrete Cosine Transform, or DCT. Here we discuss the properties of the continuous extension of the DCT, abbreviated CEDCT, such as convergence, validity of the localization principle, and its "differentiability." These properties of CEDCT, and of CEDGT generally, are similar to the properties of the canonical continuous Fourier transform, but they do not hold for the continuous extension of the standard DFT. We show that the continuous extension of the 2-dimensional DCT, formally corresponding to the DGT on the SU(2)×SU(2) group, can be effectively applied for various purposes in image processing. These include: (a) interpolation of the data between points of the grid; (b) noise suppression and visualization of targets; (c) image compression. We also present examples of similar applications of DGT of another rank-2 group, the Lie group SU(3), which allows Fourier transforms of discrete functions defined on the grids of triangular or hexagonal symmetries.

**Keywords.** Discrete transform, Fourier, lie group, interpolation, noise, image, compression, zooming

## 1. Introduction

Numerical computation of various transforms is nowadays widely implemented in practical science and industry, and also attracts significant theoretical research. Here we outline our results on the new methods for discrete Fourier transforms of functions of many variables in the context of the group theoretical approach developed earlier in [1–3], and discuss possible applications of these transforms for imaging and data processing. These methods of discrete transforms, which we abbreviate as Discrete Group transforms, or DGT, are based on the decomposition of functions defined on the fundamental region of a compact Lie group into a finite series of orbit functions of the group. As we have recently shown in [4], application of this method in the simplest possible case, which is the group SU(2) when the fundamental region $\boldsymbol{F}$ is reduced to a

1-dimensional segment, results in a type of discrete Fourier transform which is known since 1974 as *Discrete Cosine Transform*, or DCT [5,6].

Although DCT is often considered mostly as a specific case of the standard Discrete Fourier Transform (hereafter DFT), it has proven to be very effective for a number of practical applications, with general performance characteristics exceeding the ones of the DFT (see [6] for a review). In fact, these two discrete transforms become dramatically different if the properties of the *continuous extensions* (hereafter CE) of their inverse transforms onto all points of the segment $\boldsymbol{F}$, i.e. between the grid points, are considered. The CEDCT repeats many good analytic properties of the canonical *continuous Fourier transform* polynomials of appropriate finite order, such as *convergence*, *locality*, and *differentiability* of the sequence, whereas the continuous extension of the standard DFT sequence even does not converge to the originating continuous function. These good convergence properties seem to be common for continuous extensions of all other DGTs, as we demonstrate in this paper on the example of CEDGT of SU(3) groups. Generally, these Fourier transforms make "full use" of the symmetry properties intrinsic to the chosen Lie group. This provides a qualitative explanation for the good properties of such transforms.

In Section 2 of this paper we introduce the reader to the basic approaches and the results of the DGTs, the detailed derivation of which is found in [4]. In Section 3 we demonstrate the potential of the CEDGT approach for some practical applications, such as image processing and data compression, using 2-dimensional DCT for discrete functions on rectangular grids. In Section 4 we show, on the example of the application of the DGT on SU(3) groups for images collected by photo cameras of detectors with hexagonal symmetry used in the Very High Energy (VHE) gamma-ray astronomy telescopes, that these good properties seem common for other multi-dimensional CEDGTs as well.

## 2. Fourier Transforms on Lie Groups

### *2.1. Discrete Lie Group Transforms*

There are two ingredients and one basic property on which our method of DGT is based. These are *elements of finite order* (EFO's) [1,2,7] and *orbit functions* [2,8] of a compact semisimple Lie group $G$, generally of a rank $n$, and the property of orthogonality of a subset of orbit functions [2] on a discrete grid of points in $R^n$.

Without going into details, let us remind here that any element $X$ of group $G$ is conjugate to some element of its maximal torus, and that such an element is unambiguously represented by a point in the fundamental region $\boldsymbol{F} \subset R^n$ of the group $G$. There is a one-to-one correspondence between elements of $\boldsymbol{F}$ and conjugacy classes of elements of $G$. An EFO is an element $X$ of the group, such that $X^N=I$ for some natural number $N$, and that can be described by a vector in the fundamental region $\boldsymbol{F}$. Action of the Weyl group $W$ of the group $G$ transforms each vector $\lambda \in \boldsymbol{F}$ into a finite set $W(\lambda) \subset R^n$ of vectors equidistant from the origin. This set makes the Weyl group orbit of $\lambda$. An orbit function $\Phi_\lambda(r)$ is defined [2,3] as a finite sum of exponentials on this orbit:

$$\Phi_\lambda(r) = \sum_{\mu \in W(\lambda)} e^{2\pi i(\mu|r)}, \qquad r \in R^n \tag{1}$$

Here $(\mu \mid r)$ is the scalar product in $R^n$. Although $\Phi_\lambda(r)$ is defined in the entire space $R^n$, we are particularly interested in its values when $r \in \boldsymbol{F}$, and furthermore, when the coordinates of the vector $r$ are rational in the *basis of fundamental weights* $\{\omega_t \mid t = 1,\ldots,n\}$ that also define the geometry of $\boldsymbol{F}$. These are the EFOs, which can generally be expressed as $\{r_j = (\Sigma_i k_{ji}\omega_t)/N\}$ with integer $k_{ji}$ and N.

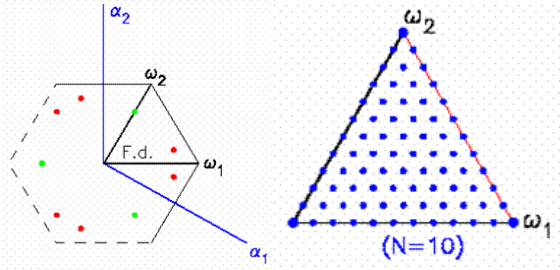



**Figure 1.** (a) *on the left*: fundamental domain $\boldsymbol{F}$ (F.d.), fundamental weights ($\omega_1$,$\omega_2$), and simple roots ($\alpha_1$,$\alpha_2$) of SU(3); (b) *on the right*: elements of finite adjoint order $N$=10 in F.

For example, in Figure 1a (on the left) we show the fundamental weights ($\omega_1$,$\omega_2$) of the SU(3) group, the fundamental domain $\boldsymbol{F}$ in the form of an equilateral triangle between these vectors, and the points corresponding to the orbit of the Weyl group for 2 elements in $\boldsymbol{F}$. The Weyl group orbit of an element $\lambda = k\omega_1 + m\omega_2 \equiv (k,m)$ in SU(3) is produced by reflections of $\lambda$ in the lines of vectors $\omega_1$ and $\omega_2$. It generally consists of 6 vectors, $W(\lambda) = \{k\omega_1+m\omega_2;\ m\omega_1-(k+m)\omega_2;\ -(k+m)\omega_1+k\omega_2;\ -k\omega_1+(k+m)\omega_2;\ (k+m)\omega_1-m\omega_2;\ -m\omega_1+k\omega_2\}$, and it is reduced to 3 vectors (or 1) if $k$=0 or (and) $m$=0. Action of the Weyl group on the fundamental domain $\boldsymbol{F}$ expands it to an hexagonal region representing the maximal torus of SU(3) (e.g. see [2]). Vectors $\alpha_1$ and $\alpha_2$ represent the roots of the group and are defined by $(\alpha_i \mid \omega_j)=\delta_{ij}$, $(\alpha_1 \mid \alpha_2)= -1$ and $(\alpha_j \mid \alpha_j) = 2$. Note also that the absolute values $|\omega_{1,2}| = \sqrt{2/3}$.

In Figure 1b (*on the right*) we show dots representing the set of EFOs in the fundamental region for $N$=10. The property that allows decomposition of discrete functions $\{g_j \equiv g(r_j)\}$ given on a grid of elements (vectors) $\{r_j\}_N$ of any final (*adjoint*) order $N$ in $\boldsymbol{F}$, is the *discrete orthogonality* of orbit functions (1) on the set $\{r_j\}_N \equiv \boldsymbol{F}_N$ [2]. For SU(3), the orthogonality of $\{\Phi_\lambda\}$ reads that any pair of orbit functions with integer $\lambda_1=(k,m)$ and $\lambda_2=(p,q)$ satisfy the relation

$$\langle \Phi_{k,m}, \Phi_{p,q} \rangle_N = \sum_{r_j \in F_N} P_j \, \Phi_{k,m}(r_j) \overline{\Phi_{p,q}(r_j)} \propto \delta_{kp}\delta_{mq} \equiv \delta_{\lambda_1 \lambda_2} \qquad (2)$$

where $P_j$ is a multiplicity factor equal to the number of elements in the Weyl group orbit of $r_j$, and the Kroneker symbol $\delta_{\lambda\mu} = 0$ if $\lambda$ and $\mu$ cannot be connected to each other by integer translations along the vectors $(\alpha_1,\alpha_2)$, and $\delta_{\lambda\mu} = 1$ otherwise [9]. Using (2), the coefficients $A_\lambda$ for the set of Fourier transform equations $f_j = \Sigma\, A_\lambda \Phi_\lambda(r_j)$ can be found.

The case of the SU(2) group represents the simplest example of a DGT on compact Lie groups. It allows, however, a most revealing comparison of properties of DGT and

DFT. For this rank 1 group the fundamental region $\boldsymbol{F}$ is reduced to a segment $[0, \frac{1}{2}]$, and the Weyl group orbit of any $\lambda \in R^1$ consists generally only of 2 elements, $W(\lambda) = \{\lambda, -\lambda\}$. The orbit functions (1) are then reduced to the sum of 2 exponential functions (to compare with 6 for the SU(3) group), resulting in $\Phi_\lambda(\theta) = 2\cos(2\pi\lambda\theta)$ for $\lambda > 0$, and $\Phi_0(\theta) = 1$ for $\lambda = 0$, where $\theta \in [0,0.5]$. For SU(2) the elements of finite adjoint order $N$ in the fundamental region correspond to the set of points $\{\theta_j = j/2N \mid j = 0,1,\ldots,N\}$. In the case of integer $\lambda = k$ the orbit functions are reduced to $\psi_k(\theta_j) = \Phi_k(\theta_j)/2 = \cos(\pi kj/N)$. This is exactly the basis for the well known *discrete cosine transform* (DCT), or more exactly, the DCT-1 (see [6]), with the orthogonality property

$$\sum_{j=0}^{N} C_{N,j} \cos\frac{\pi jk}{N} \cos\frac{\pi jm}{N} = \frac{2N}{C_{N,k}}\delta_{km}, \quad 0 \le k, m \le N, \tag{3}$$

where $C_{N,j}=1$ for $j=0,N$ and $C_{N,j}=2$ otherwise. DCT of any discrete function $\{g_k = g(t_k)\}$, produced by sampling of a continuous function $g(t)$ at points $\{t_k = k/N \mid k = 0,1,\ldots,N\}$ of the (*normalized*) interval [0,1], means inversing the system of equations $g_k = \Sigma_m a_m \cos(\pi km/N)$. With the use of (3), the transform coefficients are easily found:

$$a_m = \sum_{k=0}^{N} \frac{C_{N,m}C_{N,k}}{2N} \cos\frac{\pi mk}{N} g_k, \quad (m = 0,1,\ldots,N) \tag{4}$$

In the current literature, the DCT is often being considered as a particular case of the standard *discrete Fourier transform*, which is commonly abbreviated as DFT, and is given by the pair of equations for the direct and inverse transforms (e.g. [6,10]):

$$u_m = \frac{1}{N}\sum_{k=0}^{N-1} g_k e^{-i\frac{2\pi}{N}km}, \tag{5}$$

for the direct transform, and

$$g_k = \frac{1}{N}\sum_{m=0}^{N-1} u_m e^{i\frac{2\pi}{N}km} \tag{6}$$

for the inverse DFT. This is because formally the DCT coefficients can be calculated through the $2N$-point DFT as $a_k^{(N)} = C_{N,k}\, u_k^{(2N)}$ (see [4,11]). There is, however, a very essential difference between the DFT and the DCT. While the DCT is based on the cosine functions of both integer and half-integer harmonic order $k \le N/2$, and thus does not exceed the Nyquist sampling rate [12], the DFT uses harmonic functions $\exp(2\pi ikt)$ only of integer order up to $k=(N-1)$, which exceeds twice the Nyquist rate. This circumstance plays a crucial role resulting in very different properties of the *continuous extensions* of these two transforms [4].

### 2.2. Continuous Extension of the Discrete Transform

Both DCT and DFT correspond to "lossless" discrete transforms, i.e. their inverse transforms return the initial set $\{g_k\}$ at the grid points $\{t_k\}$. It is then natural to ask how effectively it is possible to recover the original continuous function $g(t)$ by a Fourier series on the entire continuous segment [0,1]. Note that if the original function is defined on some arbitrary segment $t \in [A,B]$, its variable can always be renormalized to the segment [0,1]. Then the ratio $k/N$ in the inverse transform is simply equal to $t_k$, which immediately suggests a natural way for a continuous Fourier extension of any discrete transform. The CEDCT of an $N$-interval discrete function $\{g_k\}$, which generally could be complex valued, results in the cosine polynomials of integer and half-integer harmonic orders $n \le N/2$,

$$f_N(t) = \sum_{n=0}^{N} a_n \cos(\pi n t), \qquad t \in [0,1] \tag{7}$$

with the DCT coefficients $a_n$ given by (4). The CEDFT is similarly found by extending $t_k \to t$ in the inverse DFT given by equation (6).

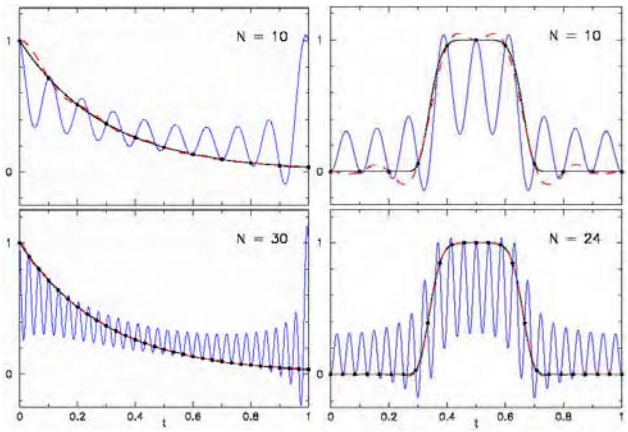



**Figure 2.** Approximation of analytic functions (*heavy solid curves*) provided by continuous extensions of DFT (*thin solid curves*) and of DCT (*dashed curves*). Large dots show the values of $g_k$ at the points $t_k$ of grids with different interval numbers $N$.

Figure 2 shows examples of CEDCT and CEDFT for discrete functions produced by the sampling of 2 continuous functions on the equidistant grids with different $N$. The CEDFT functions return the values of $\{g_k\}$ at exactly the grid points $t=t_k$, as they should, but they fail to provide any reasonable approximation between the grid points even for very large $N$. Meanwhile, the CEDCT function rapidly converges to the original $g(t)$ with increasing $N$. Note also that the DFT itself is not applicable at the last (N+1) point t=1 of the N-interval grid if the function is not periodical, i.e. if $g(0) \neq g(1)$, whereas DCT and CEDCT do not depend on that condition.

These very different behaviors of CEDCT and CEDFT can be understood in view of the Shannon-Nyquist sampling theorem, which states [11]: *every continuous band-limited analog signal $X(t)$ can be reconstructed fully from its discrete samples*

$X_k = X(k\Delta t)$, *provided the sampling rate* $\Delta t$ *exceeds twice the maximum frequency* $\nu_{max}$ *contained in* $X(t)$. The sampling rate $\Delta t = 2\nu_{max}$ corresponds to the sampling at the rate of 2 points per period $T = 1/\nu_{max}$, which is the Nyquist rate.

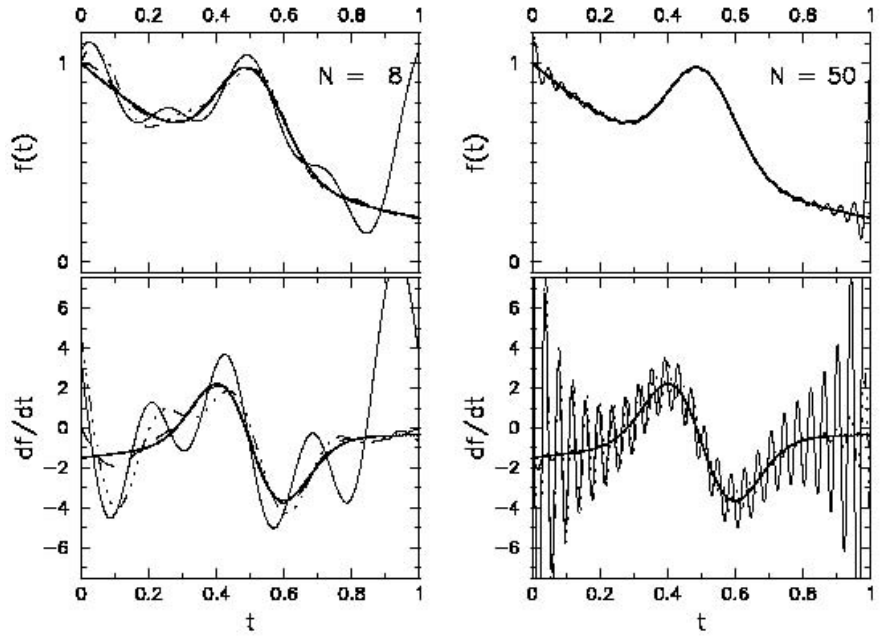



**Figure 3.** Approximation of a continuous function $g(t)$ and its derivative $dg/dt$ (*heavy solid curves*) with CEDCT function $f_N(t)$ and its derivative, respectively, for $N = 8$ and 50 (*dashed curves*). Note that the dashed curves can be distinguished from the heavy solid curves only on the left panels at small $t$. The thin solid curves correspond to the interpolation provided by series using DFT coefficients but truncated to the harmonic orders $k \le N/2$, and *3-dot-dashed curves* show approximations provided by the Shannon interpolation (see text).

Thus, for any N one can view CEDCT as a continuous band-limited function with a maximum frequency equal to $N/2$. This function is formally sampled at the Nyquist rate. However, for large N the DCT coefficients $a_k$ for large order harmonics rapidly drop [6]. It leaves the CEDCT with an *effective* bandwidth of the original function $g(t)$. Therefore at large $N$ the set $\{g_k \mid k = 0,1,...,N\}$ samples CEDCT with a rate much higher than the Nyquist rate. Therefore one could also say that effectively "g(t) provides a good reconstruction to the CEDCT." Meanwhile, for any $N$ the CEDFT function is sampled by $\{g_k\}$ *always* at the rate equal to its effective maximum frequency, as far as $\tilde{u}_{N-k} = u_k$, so the high-frequency DFT coefficients recover the amplitude of the low-frequency ones.

Besides the property of fast convergence, CEDCT also has *localization* and *differentiability* properties [4], which are very similar to the properties of canonical *continuous* Fourier transform polynomials. The localization property of $f_N(t)$ means that for large $N$ an interpolation provided by $f_N(t)$ at any chosen $t_0$ between the discrete points of the grid depends mainly on the values of $g_k$ in the near vicinity of $t_0$. This property ensures that possible errors of the grid function at some distant segment would not have a significant impact on the result of calculations at the point $t_0$.

The property of *differentiability* of CEDCT is that $f_N'(t) \rightarrow g'(t)$, if $g(t)$ is a smooth function with a bounded second derivative on the interval [0,1] (see [4]). This property is illustrated in Figure 3, where we have chosen g(t) in the form of a superposition of a Gaussian and an exponential function. For comparison, the dashed curves show interpolations provided by the continuous extension of the DFT series, but from which the high order harmonics are excluded (see [4] for details), and the 3-dot dashed curves are for the Shannon interpolation formula [11] in case of a limited number of sampling points $t_k = k/N$, corresponding to the sampling interval $\Delta t = 1/N$:

$$X_N(t) = \sum_{k=0}^{N} g_k \frac{\sin(\pi(t - t_k)N)}{\pi(t - t_k)N} \tag{8}$$

Figure 3 shows that although both of the latter 2 interpolations do converge to the original $g(t)$, neither of these series is "differentiable." Meanwhile, the CEDCT is, since its derivative rapidly converges to $g'(t)$.

## 3. Applications of CEDCT

It is important for practical applications of DCT that the coefficients of this transform allow fast methods of calculations using the fast Fourier transform algorithms [6]. It is also important that the generalization of this transform to any multi-dimensional space is straightforward, as far as multi-dimensional DCT is a separable transform. The latter implies that all the good analytic properties of CEDCT also hold for continuous extensions of DCT for functions defined on multi-dimensional rectangular grids. Note that such DCTs correspond to discrete transforms on Lie groups SU(2) ×…× SU(2). Here we consider applications of 2-D CEDCT, which has a simple form [4]

$$f_{K,N}(x, y) = \sum_{k=0}^{K} \sum_{n=0}^{N} A_{kn} \cos(\pi k x) \cos(\pi n y) \tag{9}$$

### *3.1. Image Interpolation and Zooming*

The high potential of the application of the 2-dimensional CEDCT for the interpolation of real images of high quality is demonstrated in Figure 4, where we show the effect of interpolation of a strongly zoomed 81×81 pixel fragment of the standard test image "Lena." The panel on the right shows the CEDCT interpolated image of the fragment on the left panel. The good quality provided by such interpolation is apparent.

The actual surface density of points used on the right panel in Figure 4 is 5×5=25 times larger than on the original image. Meanwhile, the real size of the 2 files are in fact exactly equal because the CEDCT file uses exactly the same number of DCT transform coefficients $\{A_{nm} \mid n=0,1,\ldots,N-1;\ m=0,1,\ldots,M-1\}$ (in the general case of a rectangular image of $N \times M$ pixel size) as the original image $\{G_{nm}\}$. It is therefore important that in principle, using the same file $\{A_{nm}\}$, the image could be stored and "continuously" zoomed whenever needed to any spatial size without increasing the actual size of the data file.

**Figure 4.** A fragment of the 256×256 pixel test image "Lena" strongly zoomed to reveal the pixel sizes (*on the left*), and its interpolation with CEDCT (*on the right).*

Effectively, this also implies new possibilities for data compression based on a CEDCT approach similar to the ones implemented in the JPEG standard, where (mostly) the 8×8 block DCT is implemented [6], but for blocks of much larger sizes the CEDCT is applied. Discussion of these possibilities, however, is out of the scope of this paper, and we refer to [12] for details.

### *3.2. Noise Suppression by CEDCT*

The fact that CEDCT, and more generally the continuous extensions of DGTs of other Lie groups like SU(3) [9], represent trigonometric polynomials with good analytic properties similar to the ones of canonical continuous Fourier transforms, suggests that the multidimensional CEDCT series truncated to harmonic modes lower than the maximal nominal modes $n_i=N_i/2$ (for blocks of size $N_i$ in the $i$-th dimension) used in the exact CEDCTs, should also be converging to the original continuous functions. This implies some important applications of CEDGT functions, such as fast and simple methods for suppression of additive noise in the images/data.

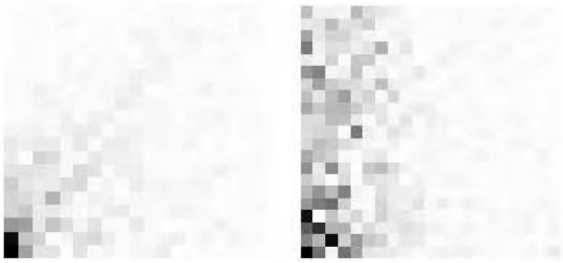


**Figure 5.** DCT "image" $\{A_{kn}\}$ (see text) of a block from the test image "Lena" (*left panel*), and of a similar size block from an image taken by FLIR detector. $A_{00}$ is on the bottom-left corner.

This possibility is basically connected with the characteristic properties of DCT that (a) for smooth ("*pure*") images the amplitudes of the high-order modes are rapidly declining [6], and (b) the Fourier transform pattern of a Gaussian noise is again a Gaus-

sian noise. The "transform image," in terms of absolute values of the 2-dimensional DCT coefficients $\{A_{kn}\}$, for a 18×18 size block from the test image "Lena" is shown on the left panel of Figure 5, where the dark intensity is scaled linearly with the absolute values $|A_{kn}|$ of the DCT coefficients. The low-order modes are on the bottom-left corner. For comparison, the panel on the right shows the transform pattern of an image taken from the Forward-Looking Infra-Red (FLIR) detector. As one can see, the amplitudes of high-frequency modes on the right panel do not decline in vertical direction. This reflects a "horizontal" noise intrinsic to FLIR detectors that operate in the line scanning mode. The relative dominance of high frequencies in the DCT of Gaussian noise, however, is a general feature, which is qualitatively explained by uncorrelated change of the amplitude of the noise signals from one pixel to the next.

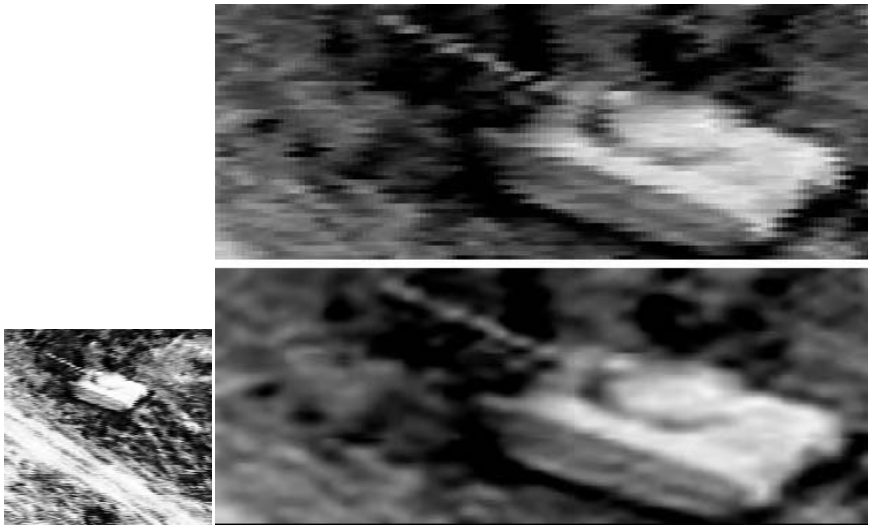


**Figure 6.** The FLIR image of a ground object (from http://www.cis.edu/data.sets/nawc_flir, NAWCWPNS, China Lake, CA; image approved for public release), which is strongly zoomed and represented by truncated CEDCT with $\alpha_{cut} = 0.5$ (see text).

Because for an image with additive noise the transform coefficients are presented as $A_{kn} = A^{(0)}_{kn} + S_{kn}$, where coefficients $A^{(0)}_{kn}$ of the "pure" image decline rapidly but $S_{kn}$ for noise does not, the latter is potentially most destructive in the domain of high frequencies. And if the CEDCT of the original image $\{A^{(0)}_{kn}\}$ is indeed similar to the continuous Fourier transform polynomials, one could expect to significantly "get rid of" this noise by simply discarding the modes in the high-frequency domain, $N_{cut} < k,n \le N$, and using the thus truncated CEDCT function to reconstruct the image.

In Figure 6 we demonstrate the effect of such simple "noise suppression" procedures. The image of a tank taken by FLIR detector is zoomed, so that noise in the image becomes apparent. Then we use the truncated CEDCT series where only the terms with $\{A_{kn}\}$ in the range $0 < k,n \le N_{cut} = N/2$ are used. Note that such simple low-pass filtering effectively also performs compression of the image by a factor $(1-\alpha_{cut})^{-2} = 4$, where $\alpha_{cut} = 1 - N_{cut}/N$. Although this procedure also removes all the information at highest frequencies that could be present in the image, and therefore formally reduces the image resolution, the positive effect of the noise suppression is obvious in Figure 6.

Note also that the CEDCT in Figure 6 is calculated for (20×20) square blocks, the result of which can be noticed in Figure 6. It demonstrates, however, that the "block-effect" connected with the "lossy" character of the truncated DCT, is rather weak for CEDCT functions. It is important for applications that this feature also holds at much higher levels of image compression [12].

## 4. Applications of CEDGT on SU(3)

The discrete transform on the Lie group SU(3) provides a possibility for Fourier analysis of discrete functions defined on triangular or hexagonal grids. Continuous extension of DGT on this group shows the same analytic properties of convergence, localization and differentiability, as CEDCT. In this section we will demonstrate some examples of these good properties of CEDGT on SU(3), referring to [9] for more a detailed description of results.

For this group, the discrete grid points which correspond to EFOs of some given adjoint order $N$ as discussed in Section 2, can be presented in the basis of weights $(\omega_1,\omega_2)$ shown in Figure 1 as $r_{kn}=(k\ \omega_1 + n\ \omega_2)/N$, where both $0 \le k,n \le N$ and $0 \le k+n \le N$. Correspondingly, the CEDGT can be written in the form

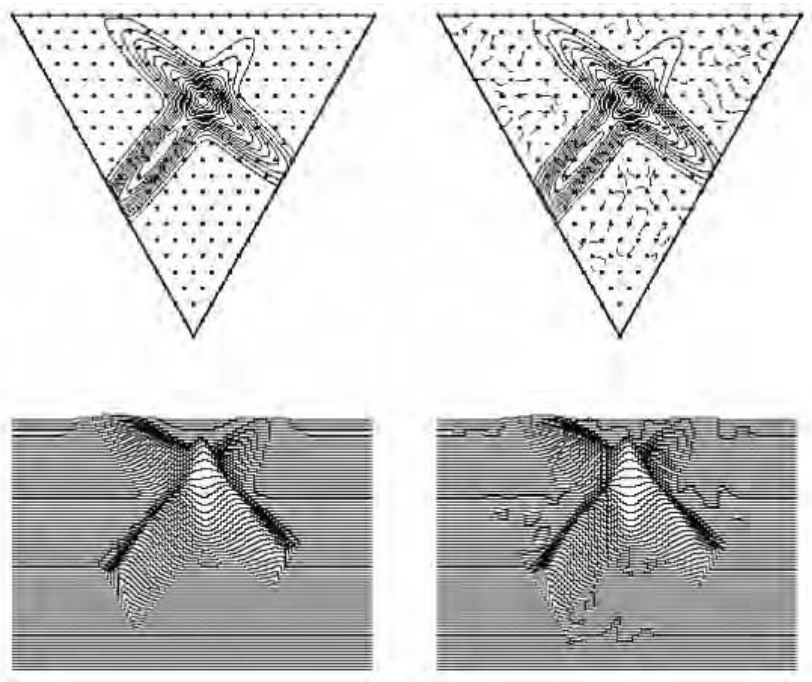


**Figure 7.** Contour plot and the 3D-view of an analytic function $G(x,y)$ composed of the superposition of 2 Gaussian ellipsoids perpendicular to each other (*panels on the left*), and its CEDGT reconstruction from the discrete function $\{G_{kn}\}$ on a triangular grid (panels on the right). The dashed contours correspond to the zero level.

In Figure 7 we show the image, in terms of contour plots and 3D-images, of an analytic function $G(x,y)$ representing the sum of two 2-dimensional Gaussian functions

with large axes perpendicular to each other, and widths (dispersions) $\sigma_{\perp} = 2/3N$, where $N$=20 is the number of intervals along the sides of the fundamental triangle (top panels). Although the dispersion in this case is even smaller than the distance $1/N$ between the points of the grid, one can see some difference between CEDGT and the analytic images only in the form of wiggles in the lowest contour levels.

In practice, detectors based on grids of hexagonal symmetry are rather specific. An example of such detectors is found in VHE gamma-ray astronomy telescopes, which detect the Cherenkov light from individual extensive air showers (EAS) initiated either by relativistic cosmic ray particles (dominated by protons) or VHE gamma-rays entering the atmosphere of Earth. The optical photons emitted by a large multiplicity of relativistic particles created in the shower are collected by the set of PMTs ("photomultiplier tubes") which are typically assembled in a hexagonal grid in the photocamera of the telescope. In order to be able to detect a gamma-ray source, one also has to be able to separate the EAS initiated by gamma-rays from those triggered by the protons. This may be possible because of existence of qualitative differences in the respective images of these 2 types of EAS, connected with differences in the processes of interactions of the protons and gamma-rays with the atmosphere. The images of gamma-ray showers should be more compact, and be oriented towards the central PMT of the photocamera of the telescope directed at the source (e.g. see [13]).

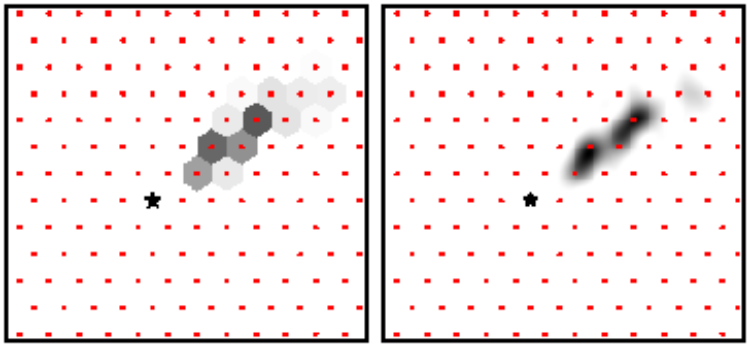


**Figure 8.** A discrete image of a Monte-Carlo simulated gamma-ray event (left panel), and its CEDGT representation (right panel). The star shows the central pixel of the photocamera.

In Figure 8 we show the result of the application of CEDGT of SU(3) to a Monte-Carlo simulated image of EAS induced by a VHE gamma-ray. The Monte-Carlo simulations are done on the basis of MOCCA code [13] supplemented by a full ray-tracing code for the telescope [14]. The continuous image is obviously more regular, and provides opportunities for the implementation of more accurate methods for discrimination between gamma-ray and proton EAS.

In order to demonstrate possibilities of CEDGT for the suppression of high-frequency noise, similar to CEDCT considered in Section 3.3, in Figure 9 we show 2 images of gamma-ray EAS and 2 images of proton EAS reconstructed by the application of CEDGT. The Monte-Carlo simulation code does not include any possible source of noise, so the irregularities seen in the gamma-ray EAS images that would be intrinsic mostly to the proton events, are explained by statistical fluctuations in the distribution of photons, the total number of which in the image on average is only of order 100. The images on the second row show the CEDGT images of the same events after

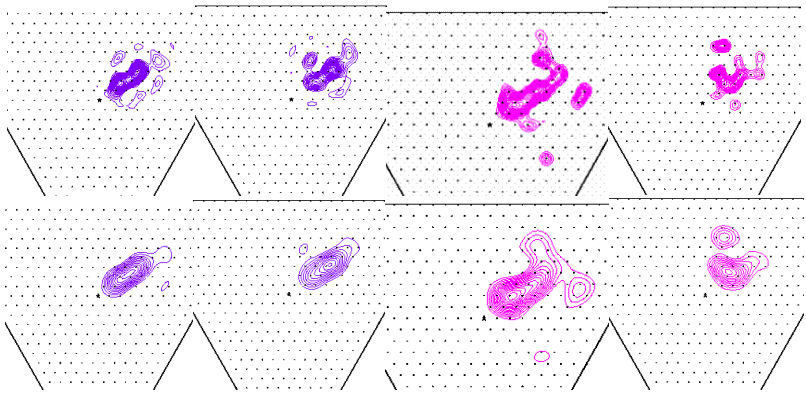


**Figure 9.** CEDGT images of EAS events initiated by VHE gamma-rays (1st and 2nd panels in the rows) and protons (3d and 4th panels). The panels in the second row show the same images after application of low-pass filter with parameter $\alpha_{cut} = 0.3$.

the application of a simple low-pass filter, as described in Section 3.3, with the cut parameter $\alpha_{cut} = 0.3$. The "gamma-ray" images have been unified, and are now more exactly oriented in the direction of the central PMT (shown by a star), while the proton images still remain significantly less ordered and poorly unified. Our study shows (paper in preparation) that the application of the CEDGT approach and filtering does increase the efficiency of the suppression of proton background and signal/gamma-ray extraction from the data.

## Acknowledgements

We are grateful to Dr. Pierre Valin and Dr. Malur Sundareshan for helpful discussions. The work has been partially supported by the NATO Collaborative Linkage Grant, Ref. PST.GLS.979437, by Lockheed Martin Canada, and by the Discovery Grant of the National Science and Engineering Research Council of Canada.

## References

[1] R.V. Moody, J. Patera. "Elements of finite order in Lie groups and their applications," XIII Int. Colloq. on Group Theoretical Methods in Physics, ed. W. Zachary, World Scientific Publishers, Singapore (1984), 308–318.

[2] R.V. Moody, J. Patera. "Characters of elements of finite order in simple Lie groups," SIAM J. on Algebraic and Discrete Methods 5 (1984), 359–383.

[3] V. Moody, J. Patera. "Computation of character decompositions of class functions on compact semisimple Lie groups," Mathematics of Computation 48 (1987), 799–827.

[4] A. Atoyan, J. Patera. "Properties of continuous Fourier extension of the discrete cosine transform and its multidimensional generalization," JMP (2004), in press; (math-ph/0309039).

[5] N. Ahmed, T. Natarajan, K.R. Rao. "Discrete cosine transform," IEEE Trans. Comput. C-23 (1974), 90–93.

[6] K.R. Rao, P. Yip. "Discrete cosine transform Algorithms, Advantages, Applications," Academic Press (1990).

[7] V.G. Kac. "Automorphisms of finite order of semisimple Lie Algebras," J. Funct. Anal. Appl. 3 (1969), 252–254.

[8] J. Patera. “Algebraic solutions of the Neumann boundary value problems on fundamental region of a compact semisimple Lie group,” Proc. of the Workshop on Group Theory and Numerical Methods, (Montreal 26–31 May, 2003).
[9] A. Atoyan, and J. Patera. “Discrete Fourier Transforms on SU(3) group,” in preparation.
[10] H.J. Nussbaumer, “Fast Fourier transform and convolution algorithms,” Springer-Verlag, Berlin Heidelberg N-Y (1982).
[11] C.E. Shannon. Proc. Institute of Radio Engineers 37 (1949), 10–21.
[12] A. Atoyan, J. Patera. “Continuous extension of the discrete cosine transform, and its applications to data processing,” Proc. of the “Workshop on Group Theory and Numerical Methods” (Montreal, 26–31 May 2003).
[13] A.M. Hillas. “The sensitivity of Cherenkov radiation pulses to the longitudinal development of cosmic-ray showers,” J. Phys. G: Nuc. Phys. 8 (1982), 1475–1492.
[14] A. Akhperjanian, R. Kankanian, V. Sahakian, A. Heusler, C.-A. Wiedner, H. Wirth. “The Optical Layout of the HEGRA Cherenkov Telescopes,” Exp. Astron. 8 (1998), 135–152.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Application of Continuous Extension of DCT to FLIR Images

Armen ATOYAN and Jiri PATERA
*CRM, Universite de Montreal, Canada*

**Abstract.** The discrete cosine transform, or DCT, represents the most simple, 1-dimensional case of discrete Fourier transforms on compact Lie groups, abbreviated as Discrete Group Transforms, or DGT. Continuous extension of the DCT has analytic properties, such as convergence and differentiability, similar to those of canonical continuous Fourier transform polynomials. Here we show that continuous extensions of 2-dimensional DCT can be effectively used for the fast suppression of high-frequency noise in the images taken by Forward Looking Infrared (FLIR) detectors simply by "cutting-off" high frequency harmonics and using the truncated CEDCT for a reconstruction of image intensity at any point within the image frame, including the ones between the grid points. We also show that interpolation provided by truncated CEDCT not only compresses the initial image file, but also allows zooming and revealing of details that are not apparent in the initial FLIR image.

**Keywords.** Discrete cosine transform, Fourier transform, image, interpolation, noise, data

## 1. Introduction

We are developing methods for the practical calculation of discrete transforms on compact Lie groups, abbreviated as DGT for *Discrete Group transform*. Generally, these Fourier transforms are based on the concept of discrete elements of finite order [1,2] which make "full use" of symmetry properties intrinsic to the problem. This provides a qualitative explanation for the good analytic properties of the *continuous extensions* of the DGTs [3], or else CEDGTs, which interpolate the image defined on the given discrete grid to any point in between. These properties include *convergence*, *localization*, and *differentiability* of CEDGT sequences, which may not necessarily hold for continuous extensions of some other discrete transforms, such as of the standard discrete Fourier transform (see [3] for details).

In the simplest implementation of this method, corresponding to the group SU(2), the DGT results in a type of transform which is known as *discrete cosine transform*, or DCT [4]. Generalization of one-dimensional DCT to any multidimensional rectangular grid is straightforward, because this is a separable transform. It also implies that the analytic properties of CEDCT hold for multidimensional CEDCT functions as well. In our previous studies [5,6] we have shown that these properties could be effectively used for various practical applications of CEDCT, such as imaging and signal processing. These properties also apply to CEDGTs generally in case of discrete functions defined on the grids of symmetries different than the rectangular.

Here we continue our discussion of possible applications of the continuous extension of 2-dimensional DCT for imagery, using images taken by Forward Looking Infrared Detectors. A characteristic feature of these images is the presence of significant high-frequency noise connected with the line-scanning mode of operation of FLIR detectors. In this paper we demonstrate the high potential of the interpolation method based on CEDCT for purposes of fast and effective image processing, including "denoising" and compression of images. These features become particularly important when and where the maximum speed and minimum data size capacities are priorities, rather than achieving the best possible quality of the image by applying various optimization algorithms based on detailed (and time consuming) statistical analysis of images.

## 2. 2D CEDCT and Noise Suppression

Let us here only remind the definition of a continuous extension of DCT. For further details refer to [4]. Consider a 2-dimensional discrete image, or generally a discrete function, $\{G_{ij}\}$ which results from the sampling of a continuous function $G(x,y)$ with $\{0 \le x \le X,\ 0 \le y \le Y\}$ on a rectangular grid of points $(x_i, y_j)$, where $\{x_i = iX/N \mid i=0,1,\ldots,N\}$, $\{y_j = jY/M \mid j=0,1,\ldots,M\}$. The trigonometric series

$$F_{NM}(x,y) = \sum_{n=0}^{N}\sum_{m=0}^{M} A_{nm} \cos\frac{\pi n x}{N} \cos\frac{\pi m y}{M} \quad , \tag{1}$$

where

$$A_{nm} = \sum_{i=0}^{N}\sum_{j=0}^{M} \frac{C_{N,i} C_{N,n} C_{M,j} C_{M,m}}{4NM} \cos\frac{\pi n i}{N} \cos\frac{\pi m j}{M} G_{ij} \quad , \tag{2}$$

with $C_{K,k} = 1$ for $k=0$ or $k=K$ and $C_{N,k} = 2$ for $0<k<K$ , represents *continuous extension* of the (*inverse*) *DCT* of the given discrete function $\{G_{ij}\}$ on all points (x,y) of the rectangle [0,X][0,Y]. At the grid points the CEDCT function returns exactly the corresponding value of the grid function, i.e. $F(x_i, y_j) = G_{ij}$ .

The set $\{A_{nm}\}$ represents the DCT of type 1. The characteristic feature of the DCT of a good quality image with a low level of noise is that the transform coefficients rapidly decrease in absolute values with the increase of either of the indices n or m [3]. This is demonstrated on the left panel in Figure 1, which shows the DCT "image" $|A_{nm}|$ of a square block, $N=M$, from the test image "*Lena.*"

By contrast, the values $|A_{nm}|$ in the next 2 panels in Figure 1, which correspond to the DCT of 2 fragments of a FLIR image of a ship shown on the upper-left panel in Figure 2, decrease with the increase of $n$ and $m$ only initially, until $n$ and $m$ are significantly smaller than N. Later on, however, the decline of $|A_{nm}|$ along the axis of $m$, corresponding to the DCT in the vertical direction of the FLIR image, stops, "saturating" at the level of noise present in the image. This is generally a characteristic feature of FLIR images, which operate in the horizontal line scanning modes and result in significant apparent discontinuity in the signal amplitudes between the nearest neighbour pixels on the same vertical direction but on different horizontal lines. Effectively, it creates

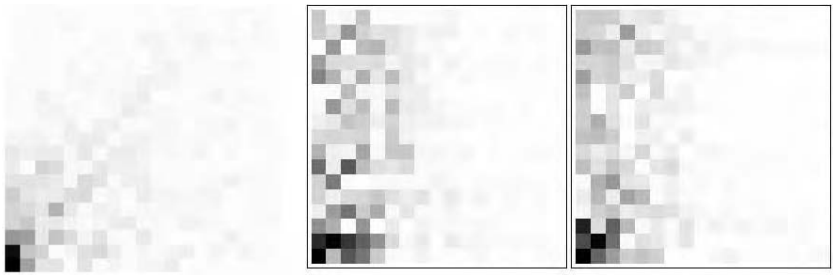


**Figure 1.** Absolute values of DCT coefficents of a fragment from the test image "Lena" (on the left) and of two fragments of a FLIR image of a ship (panels in the middle and on the right).

a random high frequency noise in the image. DCT of such noise is again a random noise which does not decline at high harmonic orders.

While for images of high quality the CEDCT function provides a good interpolation between points of the grid [5,6], direct application of CEDCT to the original FLIR images results mostly in clearing the image noise. This is demonstrated in Figure 2, where the 2 panels in the middle represent the CEDCT extensions of the original FLIR images of the ships on the 2 upper panels, respectively. Note that higher contrast of the CEDCT image of the ship on the right only makes the noise more apparent.

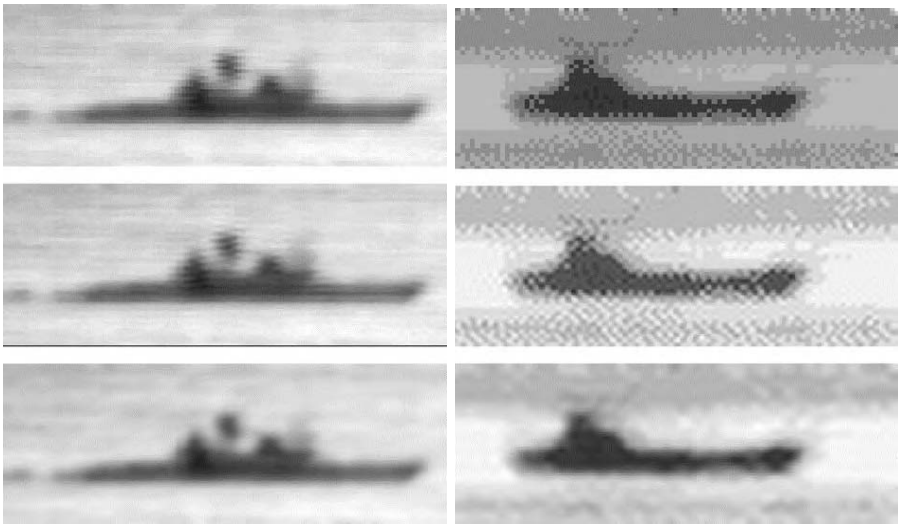


**Figure 2.** FLIR images of ships (2 top panels), and their continuous interpolations with CEDCT (panels in 2nd row). The bottom panels show the CEDCT images after cutting the high frequencies with the parameter $\alpha_{cut} = 0.5$. The original images are provided by US NAWC through Dr. Sklansky (UC Irvine, USA) and Dr. Valin (LMC, Canada).

Since noise dominates the FLIR images at high frequencies, as apparent in Figure 1, the simplest possibility for removing such noise would be to discard the high-frequency modes in the CEDCT function (1) by letting $A_{nm} \to 0$ for either $N_{\rm cut} \le n \le N$ or a $M_{\rm cut} \le m \le M$. If the properties of CEDCT are indeed close to the properties of canonical continuous Fourier transform polynomials of analytic functions [4], then one would expect that the thus truncated CEDCT series could also be used for approxima-

tion of the original image, but in such a new image the random noise fluctuations between neighbour pixels would be essentially smoothed out.

The 2 bottom panels in Figure 2 show the results of the application of such simple low-pass filters to the respective original images of the ships shown in the 2 top panels. The improvement of the images is apparent in both cases. Note that the size of the images is 64×160 pixels, and calculations are done by fragmenting the image into $N\times N$-pixel square blocks with $N$=32. In Figure 2 the value $N_{\rm cut}=N/2$ for both indices $n$ and $m$.

It is important that such procedures to "cut off" the noise effectively results in compression of the image by a factor of 4. Generally, introducing parameters $\alpha_x = 1-N_{\rm cut}/N$ and $\alpha_y = 1-M_{\rm cut}/M$, the image is compressed by a factor $C\cong [(1-\alpha_x)\times(1-\alpha_y)]^{-1}$. Our calculations show that in fact the quality of the FLIR images reconstructed by truncated CEDCT functions is practically the same if only the high frequency modes along the vertical axis $y$ are suppressed, i.e. if $\alpha_x=0$ and $\alpha_y=0.5$. In principle, such semi-truncated images would contain more high-frequency (although noisy) information that perhaps could be useful. This would, however, correspond to a twice smaller compression factor $C$, which can be crucial in various environments, such as in operations where fast on-board pre-processing and compression of data prior to their transmission are priorities.

## 3. Zooming of Image Details

Because the CEDCT is a continuous function by definition, which is able to provide a reasonable interpolation between the grid points, one can also try to use it for zooming some small details in the scene. An example of such zooming is demonstrated in Figure 3, where the upper panel shows the original scene detected by a FLIR detector. The next two panels show a zoomed fragment from that scene containing the image of a tank, and the result of the application of the CEDCT for "de-noising" that fragment. The spatial density of points used in the second panel is by a factor 9 larger than the density of pixels in the original FLIR image, but the real size of the data file for the truncated CEDCT is 4 times smaller since filtering with $\alpha_x=\alpha_y=0.5$ is used. Besides an obvious improvement of the image quality because of the removal of horizontal stripes, the "cleaned" image reveals some details which cannot be clearly distinguished in the zoomed raw image. In particular, dark stripes in the frontal part of the tank are now combined in a more united pattern which might resemble a person popping up from the tank. Truncated CEDCT can be used for further zooming of that pattern, as shown in the 2 panels at the bottom of Figure 3.

Another example of such zooming in the details of FLIR images is presented in Figure 4. Here we show a landscape containing some possible targets, in particular the one that could be seen near the crossroad at the bottom of the original image on the top-left panel. The bottom-left panel shows the same scene after its reprocessing by truncated CEDCT using the low-pass filter with $\alpha_x=\alpha_y=0.5$. Note that the spatial density of points chosen for that plot is the same as the density of pixels in the original image. Actually, it is equivalent to using the truncated DCT since there is no interpolation between the grid points. The high-frequency line-scanning noise in the FLIR image is obviously also suppressed in that case, although one can distinguish the image granularity in such a case. The two panels on the right of Figure 4 show strongly zoomed views of the crossroad target. Zooming of the original fragment on the top panel does not help, and makes the visual perception of the target even worse. Cutting off the high

frequency noise and using CEDCT with a sufficiently high density of points reveals that the target might be a vehicle, perhaps a jeep.

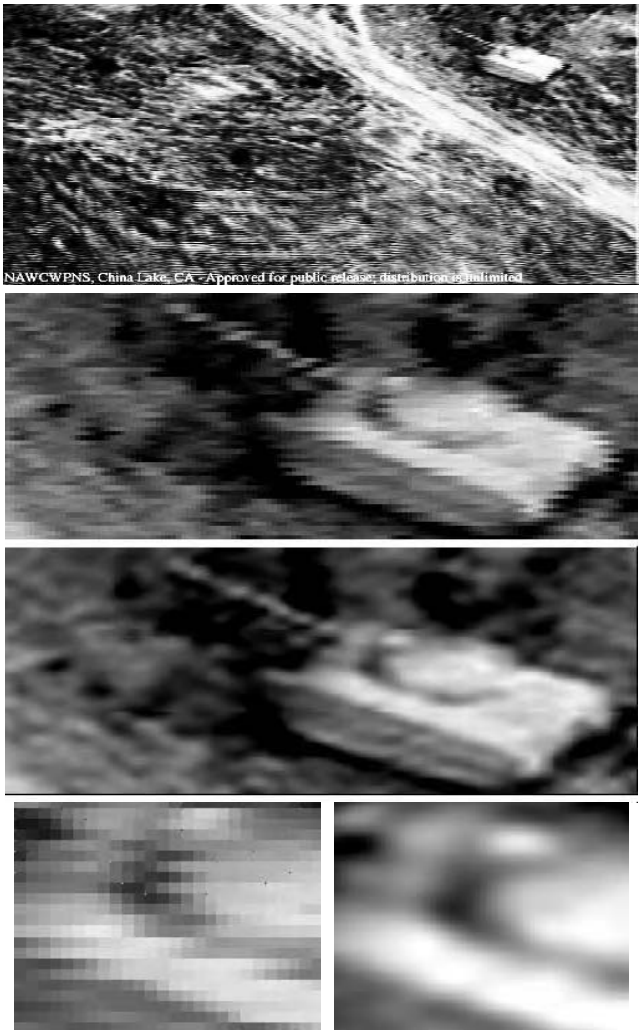



**Figure 3.** *Upper panel*: the scene detected by FLIR detector (NAWCWPNS, China Lake, CA; image approved for public release, http://www.cis.edu/data.sets/nawc_flir). *Second panel*: zoomed fragment of a tank from the original scene. *Third panel*: the image of the tank shown in the second panel represented by the truncated CEDCT with parameters $\alpha_x=\alpha_y=0.5$ and interpolated with the spatial density of points 3×3=9 times larger than the original. *Bottom panels*: further zooming of a smaller original fragment from the tank (*on the left*) and its CEDCT representation (*on the right*).

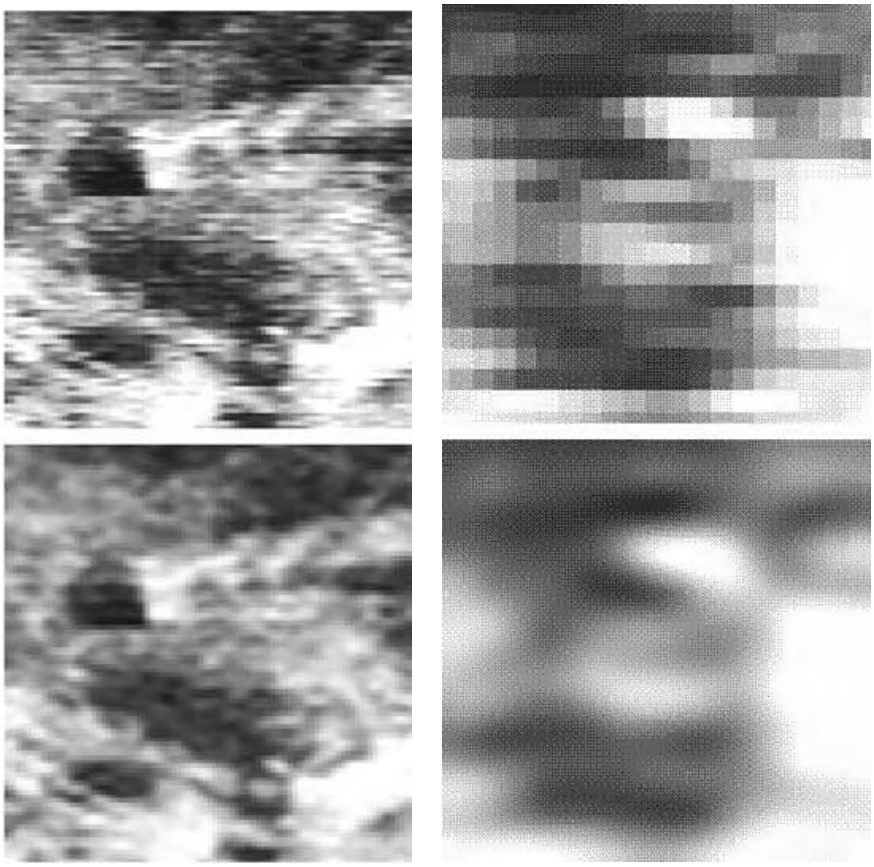


**Figure 4.** FLIR image of ground targets (*top-right panel*; NAWC, China Lake, CA, approved for public release, http://www.cis.edu/data.sets/nawc_flir), and its representation by truncated CEDCT/DCT with parameters $\alpha_x=\alpha_y$ =0.5 and spatial density of points equal to that of the original image (*bottom-left panel*). *Panels on the right* show a strongly zoomed image of a target from the original scene (*top-right panel*), and its CEDCT view with the same $\alpha_x$ and $\alpha_y$ as in the bottom-left panel, but plotted with the density of points 25 times higher.

## 4. Block Effects

A potential problem connected with the use of truncated CEDCT polynomials is that truncation of the DCT results in the loss of exactness of the transform. The CEDCT interpolation of a large image with size $N_0 \times M_0$ can be much faster if done in square blocks of size $N \times N$ with $N << N_0$, $M_0$. The lossy character of the truncated DCT then generally results in visibility of the block edges. This is a known problem of DCT transform which is implemented in the JPEG standard of image compression. Block edges become particularly apparent in the case of high compression ratios (see [3,5]).

Note that all truncated CEDCT images shown in Figures 2, 3 and 4 above are calculated in square blocks with size $N$ of about 20 (somewhat varying for different figures). Although a careful examination of these CEDCT images would reveal the block edges, the effect is not strong for the moderately low values of the cut-off parameters ($\alpha_x$ ,$\alpha_y$) used.

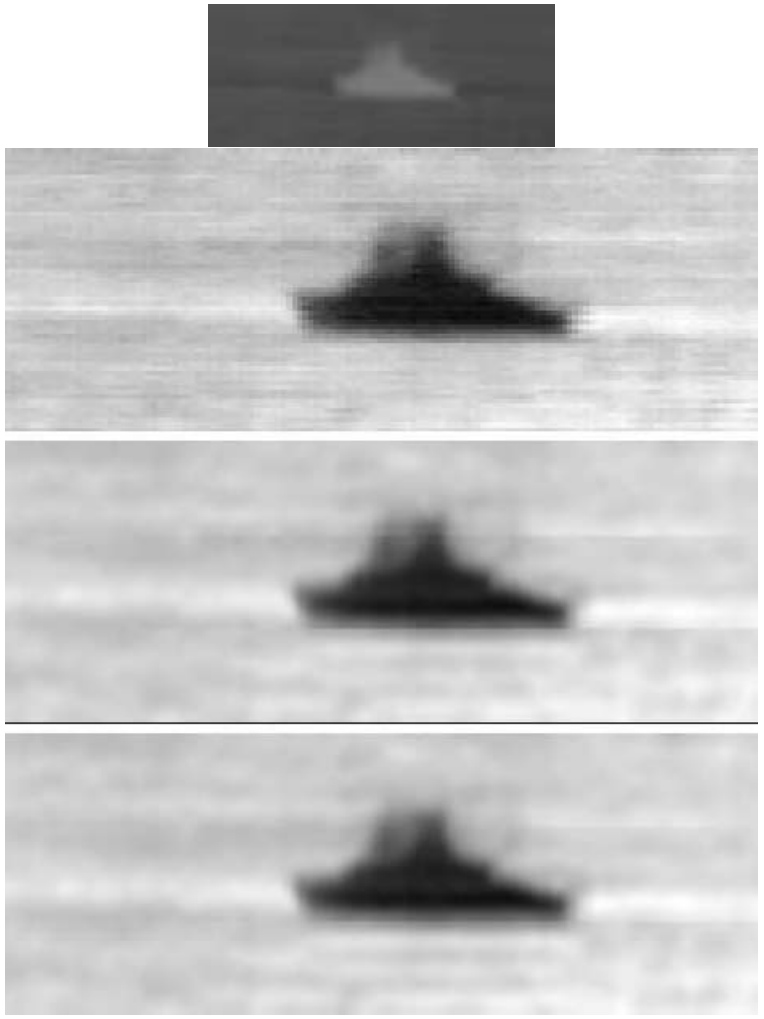


**Figure 5.** A 60×160 -pixel original FLIR image of a ship (*top panel*), and its zoomed and inverted representation on *the 2nd panel*. The *3d panel* shows the truncated CEDCT image calculated in square blocks of size 20×20 using parameters $\alpha_x$=0.5 and $\alpha_y$=0.7 and the spatial density of points 3×3=9 times larger than the density of pixels in the original image. The *last panel* shows the same image calculated using CEDCT with the same parameters as above, but using the approach of the differentiation scheme (see Section 4). The original image is provided by US NAWC through Dr. Sklansky (UC Irvine, USA) and Dr. Valin (LMC, Canada).

In Figure 5 we show a strongly zoomed FLIR image of a ship (second panel), the contrast of which has been increased and the brightness inverted to a black-on-white representation compared with the original data (top panel) for a better visual perception. The third panel shows the CEDCT image calculated in square blocks with N=20 using cut-off parameters $\alpha_x$=0.5 and $\alpha_y$=0.7. The use of a higher value for $\alpha_y$ (i.e. for the vertical direction), which corresponds to keeping only the first 7 rows from 21 in the block matrices $\{A_{nm}\}$, results in a stronger smoothing of random structures in vertical than in horizontal directions within the blocks, and also leads to appearance of 2 horizontal lines across the image indicating corresponding edges of blocks.

The last panel in Figure 5 shows the image calculated again by CEDCT, but where the suppression of high frequency noise has been done using the following scheme. The image $\{G_{ij}\}$ has been first subdivided into quartets composed of 4 elements at the apexes of the smallest square cells of the grid. Denoting the elements of such a cell as $G_{ii} \rightarrow G_1$, $G_{i+1,i} \rightarrow G_2$, $G_{i+1,i+1} \rightarrow G_3$, $G_{i,i+1} \rightarrow G_4$, the following sums and differences of these elements are constructed:

$$\begin{aligned} S &= G_1 + G_2 + G_3 + G_4 \\ d_1 &= G_1 + G_2 - G_3 - G_4 \\ d_2 &= G_1 + G_4 - G_2 - G_3 \\ d_{12} &= G_1 + G_3 - G_2 - G_4 \end{aligned} \tag{3}$$

This procedure leads to subimages $\{S_{ij}\}$, $\{d_{1.ij}\}$, $\{d_{2.ij}\}$, $\{d_{12.ij}\}$, with the total size of each 4 times smaller than $\{G_{ij}\}$. Further processing of each of these files is done by applying DCT for blocks of size $N_1 \times N_1$ (with $N_1=N/2$), and then discarding the transform coefficients in the frequency space with the same parameters $\alpha_x$ and $\alpha_y$. Using then the thus truncated inverse DCT, the new values for the subgroups $\{S_{ij}\}$, $\{d_{1.ij}\}$, $\{d_{2.ij}\}$ and $\{d_{12.ij}\}$ are found. The last step is to find the new values of the grid function $\{G_{ij}\}$ by inversing the system of equations (3) with respect to $\{G_1,G_2,G_3,G_4\}$. This function is then used for the CEDCT interpolation of the image. Note that because all transformations involved in such scheme are linear, the CEDCT will not contain the high frequency modes.

Comparing of the images in panels 3 and 4 shows that the block effects in panel 4 have been significantly reduced in such a scheme, which we call a "differentiation scheme." Note that the one quarter of the sum S in the system of equations (3) represents the best estimate for the expected value of the image function G(x,y) in the centre of the given square cell. Meanwhile all other variables introduced in this system actually are (proportional to) the derivatives of G(x,y) at that point in different directions in the (x,y) plane. Summing (with positive or negative signs) of the 4 grid function values in equations (3) is a known method for suppressing an additive Gaussian noise in the image by a factor of 2. The good performance of the considered above scheme, which actually corresponds to a differentiation scheme, can be qualitatively understood as a consequence of the convergence property of the first derivatives of the CEDCT function proved in [4].

## 5. Conclusions

The application of truncated DCT/CEDCT to images detected with FLIR detectors offers a simple method for fast and effective suppression of high-frequency noise intrinsic to these type of images. Fast Fourier transform algorithms are applicable to DCT. It is also important that the actual size of the image file is compressed very significantly, typically by factor of 4 or even higher, such as by a factor of 6 in Figure 5.

Not excluding a possibility for more sophisticated methods of noise suppression based on different image optimization procedures, we note that the method described here can be particularly useful in environments where fast image processing and/or transmission are crucial.

## Acknowledgements

We are grateful to Dr. Elisa Shahbazian and Dr. Pierre Valin for helpful discussions regarding different aspects of this work. The work has been partially supported by Lockheed Martin Canada, by the NATO Collaborative Linkage Grant, Ref. PST.GLS.979437, and by the Discovery Grant of the National Science and Engineering Research Council of Canada.

## References

[1] R.V. Moody, J. Patera. "Elements of finite order in Lie groups and their applications," XIII Int. Colloq. on Group Theoretical Methods in Physics, ed. W. Zachary, World Scientific Publishers, Singapore (1984), 308–318.

[2] R.V. Moody, J. Patera. "Characters of elements of finite order in simple Lie groups," SIAM J. on Algebraic and Discrete Methods 5 (1984), 359–383.

[3] K.R. Rao, P. Yip. "Discrete cosine transform Algorithms, Advantages, Applications," Academic Press (1990).

[4] A. Atoyan, J. Patera. "Properties of continuous Fourier extension of the discrete cosine transform and its multidimensional generalization," J. Math. Phys. (2004), to appear in May issue; (math-ph/0309039).

[5] A. Atoyan, J. Patera. "Continuous extension of the discrete cosine transform, and its applications to data processing," Proc. of the "Workshop on Group Theory and Numerical Methods" (Montreal, 26–31 May 2003).

[6] A.A. Akhperjanian, A. Atoyan, J. Patera, V. Sahakian. "Application of multi-dimensional discrete transforms on Lie groups for image processing," these Proceedings.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Neural Network-Based Fusion of Image and Non-Image Data for Surveillance and Tracking Applications

## *Neural Network Fusion of Image and Non-Image Data*

Malur K. SUNDARESHAN and Yee Chin WONG
*Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ*

**Abstract.** While it is intuitive that addition of information captured by an imaging sensor should improve the surveillance and tracking performance delivered by a system configured to process the data measured by a primary sensor, typically of the non-image type (from a range radar, for instance), such addition in practice is fraught with complexities arising not only from the dissimilar nature of the data captured by the two sensors but also due to the significantly enhanced computational complexities posed by the implementation of a mechanism to process image data. This paper addresses the question – How to employ image-based target information for improving the performance of a surveillance and tracking system without significantly enhancing the computational burden? By selecting an illustrative scenario of tracking a maneuvering target, we answer this question by proposing a feature-level fusion of image and non-image data implemented by a trained neural network that provides an estimate of the target maneuver and updates a Kalman filter in order to continue to provide target state estimate in the face of arbitrary and complex maneuvers executed by the target. The architectural details and the maneuver tracking performance of such a scheme that is capable of fusing data from two dissimilar sensors, an imaging sensor and a non-imaging sensor (primary sensor used for tracking the target), will be described. The integration of the Kalman filter with a trained multilayer neural net in essence provides an intelligent way of implementing an overall nonlinear tracking filter (without the attendant computational difficulties), and the use of a neural network that integrates information measured by the diverse sensors in order to provide a reliable estimate of the target maneuver is a novel idea underlying this approach. Results from a number of performance evaluation studies are included to demonstrate the strong points of the fusion scheme as well as of the overall tracking system

**Keywords.** Image fusion, neural networks, target tracking, maneuvering targets

## 1. Introduction

A variety of sensing devices ranging from radar systems to lasers and optical imaging systems are presently being developed for surveillance and tracking operations. The limitations of using a single sensor in these operations, such as limited accuracy and lack of robustness, have motivated the trend towards designing surveillance and track-

ing systems with multiple sensors either deployed on the same platform (an airborne or spaceborne reconnaissance platform or a tactical missile seeker, for instance) or on distinct platforms (such as in multiple earth orbiting remote sensing satellites) which can provide large amounts of useful data to detect, track and identify targets of interest. However, current surveillance and tracking algorithms usually use information from only one sensor (such as a Track-While-Scan (TWS) radar) or attempt to combine information from different sensors in an ad hoc manner. While it is intuitive that using additional data available can result in improved detection, classification and track maintenance performance, attempting to include this data efficiently to perform the ultimate task will require novel methods for fusion of information measured, which need to be carefully tailored due to the disparate forms of data collected. Development of such signal processing methods aimed at enhancing the surveillance and tracking performance is the primary focus in this article. In order to develop precise architectures and algorithms for data fusion, and to demonstrate quantitative performance improvements resulting from fusion of measurements from multiple sensors, we will consider a specific illustrative scenario of target tracking where a typically non-cooperative target is executing complex evasive maneuvers. Corresponding architectures and algorithms in other simpler surveillance scenarios (such as in air traffic control scenarios where the targets are generally cooperative and do not execute fast maneuvers) can be developed in a similar manner.

A major limitation that often precludes the integration of additional data that may be available, perhaps in a form that is different from the main data form being used, is the resulting complexity of the needed processing. For the particular case of target tracking, it is rather well known that, while simple linear processing algorithms employing a Kalman filter for target state estimation can be synthesized for processing radar data, inclusion of a different form of data (image or image-format data, for instance) will require a nonlinear processing method (such as an Extended Kalman filtering algorithm) [1]. The enormous processing complexity this may introduce could render the implementation impractical due to the real-time processing requirements underlying the tracking function and the need to keep up with the rapid target motions during the maneuvers. Consequently, an intelligent architecture that facilitates successful fusion of the diverse data forms to result in reliable surveillance and tracking performance in the face of complex target maneuvers is a highly challenging task.

Due to reasons that will be emphasized later, feature-level integration of information measured by different sensors will be the paradigm of choice for sensor fusion in this work. Extraction of appropriate features from the signals to be fused thus forms the critical step in the design of fusion architectures and algorithms. A major advantage of a feature-level fusion scheme (as distinct from data-level fusion or decision-level fusion [2]), such as the one shown in Fig. 1, is that it permits signals acquired from sensors with dissimilar characteristics to be combined more easily (such as integration of radar data and image-format data, for instance). Furthermore, once the sensor outputs are abstracted into a set of features, a number of sophisticated procedures, both mathematical (probabilistic or statistic-based combining, for instance) and knowledge-based (such as rule-based combining or neural network methods), can be utilized to design specific algorithms for integration of information.

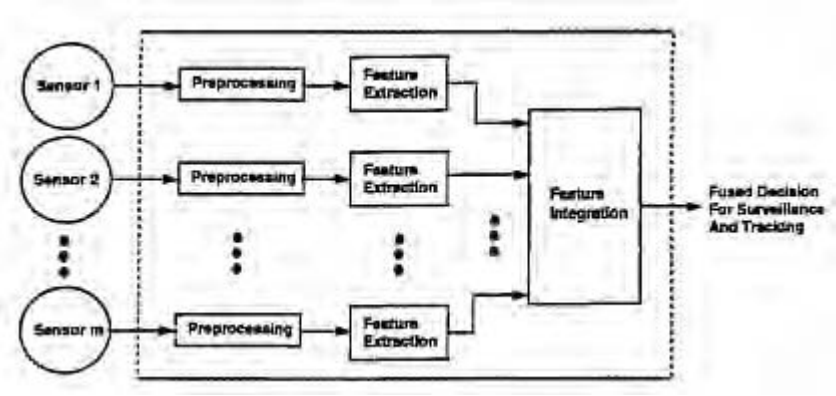



**Figure 1.** A general feature-level fusion scheme for combining multi-sensor data.

The major question addressed in this article is the following: How to employ image-based target information for improving the performance of a tracking filter without significantly enhancing the computational burden? To answer this question, we propose a feature-level fusion scheme implemented by a trained neural network that provides an estimate of the target maneuver and updates a Kalman filter in order to continue to provide target state estimate in the face of arbitrary and complex maneuvers executed by the target. The basic building blocks of a specific maneuver tracking scheme that enables a simple updating of an existing tracking filter (a Kalman filter, for instance) designed to process one type of data (measurements from a range radar, for instance) by accepting data from additional sensors, perhaps dissimilar to the primary sensor, and which employs a neural network as a fusion device, is shown in Fig. 2. The neural network accepts as inputs a set of features extracted from the data in each sensor channel and is trained to output estimates of a set of maneuver parameters characterizing the target maneuver that is represented in the feature set. Since features abstracted from the measurements obtained from dissimilar sensors are used as inputs to the neural network, the processing of data by the network implements a feature integration process and thus performs sensor fusion. The neural network outputs are used to update the target state estimates formed by a Kalman filter, which implements a recursive state estimation algorithm based on a linear model of the target dynamics. The architectural details and the maneuver tracking performance of such a scheme that is capable of fusing data from two dissimilar sensors, an imaging sensor (sensor to be added) and a non-imaging sensor (primary sensor used for tracking the target), will be described in this article. The integration of the Kalman filter with a trained multilayer neural net in essence provides an intelligent way of implementing an overall nonlinear tracking filter (without the attendant computational difficulties), and the use of a neural network that integrates information measured by the diverse sensors in order to provide a reliable estimate of the target maneuver is a novel idea underlying this approach.

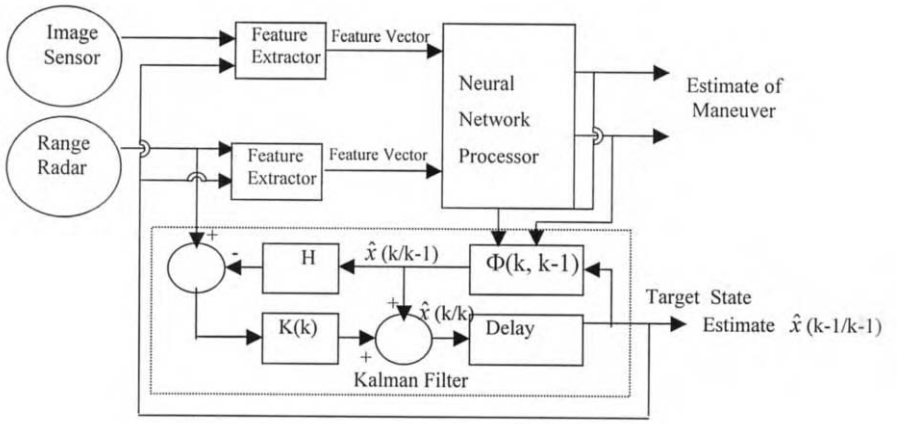



**Figure 2.** Neural network-based fusion of data from dissimilar sensors for tracking target maneuvers.

## 2. Fusion of Image and Non-Image Data for Surveillance and Tracking

### *2.1. Use of Imagery Data in Tracking of Target Maneuvers*

Amongst the surveillance and tracking problems that have received attention in the literature, sensor-based tracking of target maneuvers is noted for the complexities associated with the problem and the challenges posed to the designer. It is well recognized that the basic problem posed by a maneuvering target is the mismatch between the modeled target dynamics and the actual dynamics [4,5]. Traditionally, the exogenous target dynamics, i.e. the dynamics resulting from operator-induced or pilot-induced maneuvers, are modeled as a continuous random process in the design of a Kalman filter (for example an autoregressive process driven by white random noise). Provided this target model is correct, the Kalman filter provides reliable estimates of the target's position and velocity. However, if the target initiates and maintains a sudden pilot-induced maneuver, the target model will not be correct since pilot-induced maneuvers are often not well modeled as continuous random variables. Unless this discontinuity is accounted for, the Kalman filter will accumulate errors, diverge, and possibly lose track. Note that if the maneuvers are properly and accurately compensated, the Kalman filter can continue to provide reliable target state estimates. It may however be noted that target trajectories, especially in fast and evasive maneuver scenarios (such as combat and dog-fight environment), are highly complex and hence there will always exist a mismatch between the modeled dynamics and the true dynamics. Although the two aforementioned problems, viz. the mismatch between the true and modeled target dynamics and the requirement for a precise evaluation of the needed compensation for the maneuvers executed, are considered in the development of target tracking algorithms [4–7], fundamental limitations set by the discrete nature of the tracking system limits its accuracy.

Single Target Tracking (STT) systems are generally employed for tracking a single target and they usually implement a pointing and tracking scheme. As the name implies, a pointing and tracking system is a system whose radar is "instructed" to point to a specific location to look for the target. The location is extrapolated in time from the

most recent state, which is the target's current position and velocity. It is with this type of scheme that the track of a target executing complex maneuvers is easily lost. To circumvent the problem of losing target tracks, a STT system may increase its sampling rate, and, in the event that the target is lost, the system may execute a search for the target. However, both actions are undesirable for the following reasons. First, as mentioned above, searching for targets costs energy and second, increasing the sampling rate leads to increased data load, which in turn requires the system to have a high processing capability. These may render the real-time implementation of the tracking system impractical. Clearly, in order for a STT system to avoid losing target tracks, and at the same time maintain reasonable energy consumption, reliable estimates of the target maneuvers are necessary. One way of achieving more reliable estimates of the target maneuvers is to capture maneuver-related information during the course of their execution or prior to it if possible. Herein lies the advantage of imaging sensors. The early work of Kendrick et al. [8] points out the significant coupling that exists between acceleration and orientation, and the importance of using imaging sensors for target surveillance and tracking. It must be emphasized that to use imaging sensors effectively, they must be pointed towards the vicinity of the targets and hence require a form of pointing and tracking system. Hence this necessitates evaluating the performance of the tracking system in a point and track environment.

Generally, different types of information can be extracted from a sequence of image frames and the methods to be used in the extraction process depend on the type of application. For example, in some applications perhaps only the spectral composition of the image is of interest, in which case Fourier transform methods may be more appropriate in the extraction process. For target tracking applications, the two principal types of information that can be extracted from an image are the centroid and the orientation of the target of interest, which in turn yield two distinct approaches to image-based tracking: viz., Tracking using target centroid [9,10] and Tracking using target orientation [11,12]. The orientation of a target is the "placement" of a target with respect to some references such as a reference plane, line-of-sight, etc. For example, the roll, pitch, and yaw of a aircraft relative to a sensor's line-of-sight can be used to represent its orientation. The centroid of a target, on the other hand, can be considered as a point-mass state estimate of the target. It can be calculated using existing algorithms such as those developed in [13,14]. Both approaches have their strong and weak points. In general, the orientation angles provide more information about the target motion than the centroid. However, evaluation of the orientation angles from an image sequence is highly computation-intensive and does not lend itself to real-time implementations of algorithms capable of tracking of high speed target motions. Since our interest in this article is in the tracking of complex maneuvers executed by a target utilizing features extracted from the image for training a neural network, we will limit ourselves to using the target centroid computed from the imagery data. Use of orientation data (instead of the centroid, or in addition to the centroid) can be considered in exactly the same manner if available computation resources permit this in the specific tracking scenarios of interest. It should be noted however that irrespective of whether the target centroid or the target orientation is contemplated for use, the acquired image data may need to be preprocessed before computing these features. Typical preprocessing operations [15] that improve the value of the features extracted from the image data include enhancement of the contrast or the resolution in the image, filtering for noise and clutter removal, segmentation, region-of-interest extraction, etc.

### 2.2. Tracking with Target Centroid Computation

In this section we shall give a brief discussion of tracking with target centroid, since this is the information that will be abstracted from the image data in order to compute the features used in our fusion experiments. Fig. 3 shows a typical flow chart of the operations in centroid-based tracking using image data. For a brief description of the various blocks, the output of the imaging sensor after appropriate preprocessing operations is input to the Transformer. Assuming that the pixel intensity is discretized into a certain number of gray levels (say, 256), the Transformer divides the image into several layers of gray level intensities, with each layer characterized by an upper and a lower limit. It is assumed that a sufficient number of target pixel intensities are within the limits of a certain layer. Using the upper and lower limits of the "target layer" as threshold limits, the original image is converted into a binary image (which is the output of the Transformer). The Clusterer then groups the pixels in the binary image into clusters (i.e. the output of the Clusterer block for an image consisting of multiple targets is an image of clusters). The Centroid Calculator then computes the centroids of the clusters of interest. The calculation of the centroid reduces the clusters to point-mass state estimates of the various targets. These point-mass state estimates are then forwarded to the tracking filter, such as a Kalman filter or a Probabilistic Data Association Filter (PDAF) [6,7], whose output is the estimated state vector of each target.

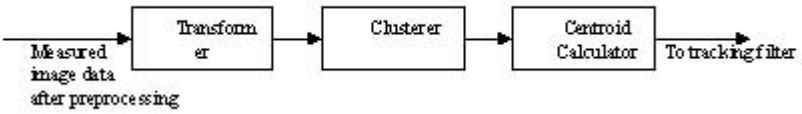



**Figure 3.** Schematic of image-based tracking system using centroid computation.

Some details on the signal processing operations performed within the individual blocks are useful. The function of the Transformer is to convert the original image into a binary image. The input image is divided into 3 layers of intensity and a new binary image, with new intensity $\beta_i$, is obtained by a hard limiter according to

$$\beta_i = \begin{cases} 1 & I_L \le I_i \le I_H \\ 0 & otherwise \end{cases} \tag{1}$$

where $I_L$ and $I_H$ are two threshold limits selected by the user, and $I_i$ denotes the intensity of pixel i in input image. The computation performed downstream within the Clusterer block involves computing the probability $p_i$ of the pixel intensity attaining value 1, called pixel detection probability. This quantity is defined by

$$P(\beta_i = 1) = p_i \tag{2}$$

Evidently, $P(\beta_i = 0) = 1 - p_i$. Assuming a Gaussian distribution for $I_i$ with mean $\mu$ and variance $\sigma^2$ (i.e. $I_i$ is $N(\mu, \sigma^2)$), $p_i$ can be computed as

$$p_i = \frac{1}{\sigma\sqrt{2\pi}} \int_{I_L}^{I_H} e^{-(x-\mu)^2/2\sigma^2} dx \tag{3}$$

The mean and the variance of the binary intensity of a single pixel in this case can be shown to be

$$\mu(\beta_i) = p_i; \quad \text{var}(\beta_i) = p_i(1-p_i) \tag{4}$$

Figs. 4a and 4b respectively show a simulated image of a target in a noisy background that is input to the transformer and the corresponding output from the transformer.

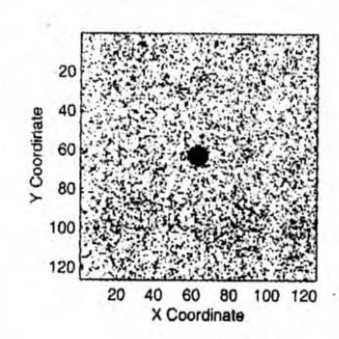



**Figure 4a.** A typical input image containing a target.

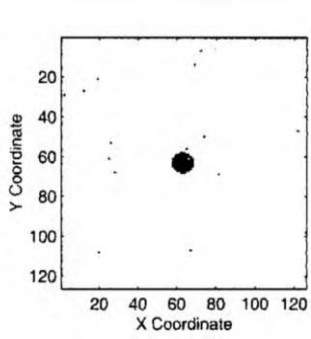



**Figure 4b.** Resulting image after original image is processed by the transformer.

The function of the Clusterer is to group the binary image into clusters. To perform this operation any standard clustering method such as the nearest neighbor technique, K-means clustering, neural network-based methods, etc. [13,14,16] can be used. For the present application of computation of target centroid from an image frame, many of the popular clustering methods ensure satisfactory levels of classification efficiency; however, some of these may demand significant computational resources that may preclude real time implementation. Hence the method to be used needs careful selection keeping this tradeoff in mind. A powerful clustering procedure is the recently developed K-means hierarchical clustering algorithm [16]. Simple clustering schemes that employ the nearest neighbor technique assign a pixel to a cluster if it is linked to at least one other pixel within that cluster by a distance which is less than a certain proximity distance. The proximity distance, denoted by $d_p$, affects the size, shape, and number of the clusters obtained by clustering. For example, if $d_p$ is larger than the size of the image, then the whole image may show up as a single cluster. On the other hand, if $d_p$ is less than one pixel, then every single pixel becomes a separate cluster. A typical selection of $d_p$ using the pixel detection probabilities in the target region and in the noise region, suggested by Bhanu [17] can be made as

$$d_p = \frac{d_t + d_v}{2} \tag{5}$$

where $d_t \cong \sqrt{1/p_t}$ and $d_v \cong \sqrt{1/p_v}$, $p_t$ denoting the pixel detection probability of the target while $p_v$ denotes the pixel detection probability of the noise. $p_t$ and $p_v$ are calculated from an image of a target with a surrounding ring of background noise. More specifically, assume that a target, of size (area) $N_t$ pixels with pixel detection probabilities $p_t$, is surrounded by a "basic" ring of $N_v$ pixels, due to noise, with pixel detection probabilities $p_v$, and that the intensity of the pixels for such image is given by

$$I_i = \begin{cases} s_i & 1 \le i \le N_t \\ v_i & elsewhere \end{cases}$$

if the pixel intensity of the target and noise are assumed to be Gaussian with specified mean and variance, then $p_t$ and $p_v$ can be evaluated from Eq. (3) for the chosen intensity limits $I_L$ and $I_H$.

The Centroid Calculator, as its name implies, calculates the centroids of the clusters generated by the Clusterer. For calculating the centroid of a cluster, consider a cluster of N points (pixels) in a Cartesian coordinate system where each point is denoted by a single index i, i =1, …, N. The cluster centroid is defined by

$$x_{nc} = \frac{\sum_{i=1}^{N} x_{ni} I_i}{\sum_{i=1}^{N} I_i} \tag{6}$$

where $x_{ni}$ is the $n^{th}$ coordinate of point i and $I_i$ denotes its intensity. It may be noted that not all clusters need to have their centroids calculated. A cluster whose size is less than $N_{min}$, a user-defined parameter, can be ignored. The value of $N_{min}$ can be appropriately chosen from knowledge of the expected target size. Once the centroid of a cluster corresponding to a target of interest is calculated, it is then a simple matter to relate it to the target's spatial position, i.e. the x-, y-, and z-coordinates, via simple geometry and trigonometry. That is, given a series of centroids extracted for a target from successive image frames, a series of target position measurements can be obtained. This data can then be forwarded to the tracking filter, which will then process the image-derived target position information in the same manner as it would process measurements from the radar.

### *2.3. Alternate Fusion Schemes for Combining Image and Non-Image Data*

For fusing measurements formed by an imaging sensor with those obtained from a primary tracking sensor, such as a radar that provides target range measurements, the design goals outlined above will be served well by employing a scheme that extracts target position information from the image data. The information obtained from the two sensor channels will then be of a similar nature, which consequently simplifies the fusion requirements. Since the computation of the target centroid from each image frame provides such data (as described in the previous section), one can implement a scheme

for extraction of features that are of a comparable form to the features extracted from the radar measurements. The task for the neural network will then be to appropriately combine the two sets of features in order to produce an estimate of the target maneuver parameters, and in turn update the Kalman filter estimate of the target position and velocity. This will realize a fused system that permits adding an imaging sensor to an existing tracking system served by a primary tracking sensor (such as the radar) and implements a multi-sensor target tracking architecture of the type shown in Fig. 2. It should be emphasized that training of the neural network in order to produce an estimate of the target maneuver by no means essentially requires that the features extracted from the two data streams to be fused are of a like nature. In fact one of the strengths of employing a neural network for this function is that signals that are highly dissimilar can be simultaneously used as inputs to the network. Thus, one can use the same architecture (as in Fig. 2) for integrating orientation angle information extracted from the successive image frames with the features extracted from radar data. However, the use of target centroid extracted from image data, due to its similarity with data from the other sensor, permits a number of alternate methods to be employed for combining the information from the two parallel sensor channels, which will be described in the next section. In addition, as mentioned before, computation of centroid is less computation-intensive and may hence be preferred in the tracking of fast target maneuvers where time plays a critical role. The specific features that will be extracted from each sensor channel for training the neural network will be described in a later section.

Several alternatives exist for combining the data from the two sensor channels to implement the target tracking scheme shown in Fig. 2, with one of the sensors being an imaging type while the other producing non-image data. In particular, three distinct architectures, shown in Figs. 5a–c, can be employed to implement the fused two-sensor target tracking system. For discussion in the following, let us denote the three architectures shown in Figs. 5a, 5b, and 5c as Arch 1, Arch 2, and Arch 3 respectively. In Arch 1, a set of features is extracted from each of the two sensor measurements and these are processed within the Combiner block to form a set of combined features. Very simple combination rules can be selected if the two sets of features are of similar nature. The combined features are then used for training the neural net to estimate the maneuvers. If the two sets of features are similar in nature, pairs of like features can be combined using any appropriate combination rule that reflects the user desire to emphasize one set of information over the other (for instance, if one sensor is known to be faster or more reliable than the other for the specific tracking scenario and the conditions of operation). For the tracking results that will be demonstrated in a later section, we have used a linear combination rule, i.e. weighted sums of the two features in each pair, with the features derived from the image sensor data being weighted 75% while the features derived from the range radar data being weighted 25%, i.e., the combined features were developed according to the rule

$$O(k) = 0.75 ID_{in}(k) + 0.25 RD_{in}(k) \tag{7}$$

where $O(k)$ is the output of the Combiner block shown in Fig. 5a, $ID_{in}(k)$ is the output of the Feature Extractor block that processes the imaging sensor output and $RD_{in}(k)$ is the output of the Feature Extractor block that processes the range radar output, $k$ denoting the discrete time instant. It should be emphasized that this rule for combining the features is only illustrative and is chosen for quantitatively evaluating the tracking per-

formance in specific scenarios (as will be noted later, motivation for selecting the weights comes from the observation that the imaging sensor provides measurements about three times as fast as the radar in our simulation experiments). More sophisticated nonlinear combinations of the features may be considered if desired in specific implementations (for a few other types of methods for combining the features and their effects on the overall tracking performance, one may refer to [19].

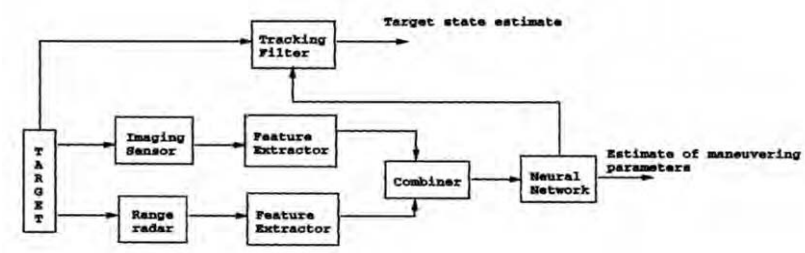



**Figure 5a.** Arch 1 two-sensor fusion architecture.

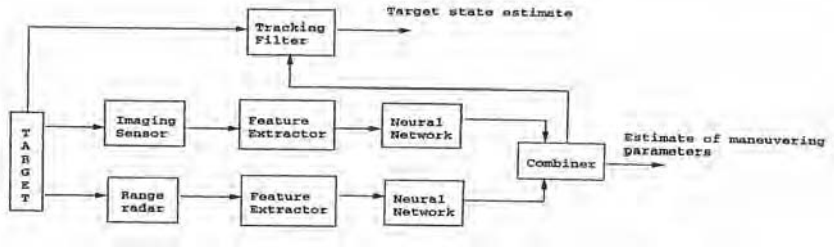



**Figure 5b.** Arch 2 two-sensor fusion architecture.

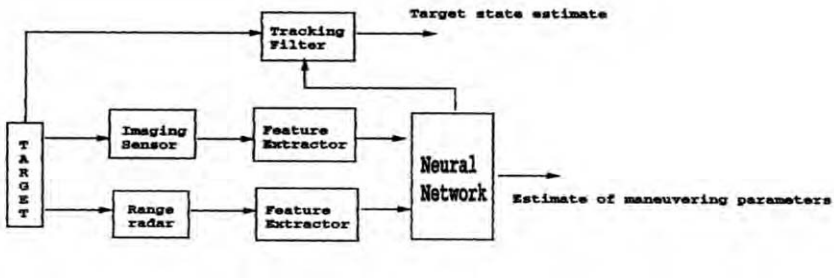



**Figure 5c.** Arch 3 two-sensor fusion architecture.

The architecture Arch 2, is slightly different from architecture Arch 1 in that the features extracted from each of the sensors are not combined prior to feeding to the neural network. Instead, the features are kept distinct and are fed into two distinct neural nets. The outputs of the neural nets (the estimates formed of the target maneuver parameters) are then combined to form an overall output. Again, because the forms of the outputs are similar, they can be linearly combined employing a rule similar to that

in Eq. (7) and using the typical weight values discussed earlier. The features from each of the sensors in Arch 3 are kept distinct. However, instead of feeding them into two distinct neural nets, the features are fed into a single neural net, which is trained with all features. The output of the neural net gives the target maneuver estimate. In a later section we shall provide a comparative performance evaluation of the three tracking architectures.

## 3. Design of the Neural Network-Assisted Tracking Filter

### *3.1. Dynamics of the Tracking Filter*

For reliably estimating the state of a target that may perform a complex maneuver in a clutter and noise environment, the present tracking scheme employs recursive state estimation performed by a Kalman filter which is assisted by a trained neural network. The design of the Kalman filter is based on the Equivalent Velocity Tracking Model (EVTM) for target motion [19,20] during a maneuver, which for a 2-dimen-sional tracking scenario has the form

$$X(k+1) = FX(k) + G\,\delta(k) + \gamma(k) \tag{8}$$

where $X^T(k) = [\,x(k) \;\; \dot{x}(k) \;\; y(k) \;\; \dot{y}(k)\,]$ is the state vector, $\gamma(k)$ is the process noise and $\delta^T(k) = [\,\delta_x(k) \;\; \delta_y(k)\,]$ is the "velocity deviation vector" where $\delta_x(k) = v_{x_{eq}}(k) - \dot{x}(k)$ and $\delta_y(k) = v_{y_{eq}}(k) - \dot{y}(k)$, and $v_{x_{eq}}(k)$ and $v_{y_{eq}}(k)$ are "equivalent velocity variables" [19, 20] that relate two successive positions along the target motion path {x(k), y(k)} and {x(k+1),y(k+1)} by

$$\begin{aligned} x(k+1) &= x(k) + Tv_{x_{eq}}(k) \\ y(k+1) &= y(k) + Tv_{y_{eq}}(k) \end{aligned} \tag{9}$$

T denoting the sampling period used. The matrices F and G in Eq. (8) are given by

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \quad G = \begin{bmatrix} T & 0 \\ 2 & 0 \\ 0 & T \\ 0 & 2 \end{bmatrix}$$

It is important to note that during a target maneuver, the excitation variables $\delta_x(k)$ and $\delta_y(k)$ in Eq. (8) are unknown, and hence one needs estimates for these,

$\hat{\delta}_x(k)$ and $\hat{\delta}_y(k)$, in order to implement the filter. These estimates are obtained as the outputs of a multilayer neural network realizing the architecture depicted in Fig. 2.

The observation sequence

$$z(k) = HX(k) + \rho(k) \tag{10}$$

where $z(k)$ is the measurement vector, $\rho(k)$ is the measurement noise and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

is processed by the Kalman filter which implements a sequence of steps [21] to generate the minimum mean square error (MMSE) estimate, $\hat{X}(k)$ of the target state. These steps can be briefly outlined as follows:

1. One-step Prediction

$$\hat{X}(k \mid k-1) = F\,\hat{X}(k-1 \mid k-1) + G\,\hat{\delta}(k-1) \tag{11}$$

2. Filtering

$$\hat{X}(k \mid k) = \hat{X}(k \mid k-1) + K(k)\left[z(k) - H\,\hat{X}(k \mid k-1)\right] \tag{12}$$

3. Gain Computation

$$K(k) = P(k|k-1)H^T\left[HP(k|k-1)H^T + R\right]^{-1} \tag{13}$$

4. Covariance Updating

$$P(k \mid k-1) = FP(k-1 \mid k-1)F^T + GQG^T \tag{14}$$

$$P(k|k) = \left[I - K(k)H\right]P(k|k-1) \tag{15}$$

It may be noted from step (ii) above that the filter processes the innovation sequence $\{\tilde{z}(1),\tilde{z}(2),....,\tilde{z}(k),....\}$ where

$$\tilde{z}(k) = z(k) - H\hat{X}(k / k-1), \qquad \text{k=1,2,....} \tag{16}$$

Thus, with a primary tracking sensor (such as a range radar) providing the required data, the Kalman filter will satisfactorily filter out the noise and permits a reliable tracking of the target.

When the target is not maneuvering (i.e., when $\hat{\delta}(k)=0$), the mean of the innovation sequence is zero. However, when the target begins to maneuver (i.e., when $\hat{\delta}(k)\neq 0$), the mean of $\tilde{z}(k)$ is no longer zero and can be utilized to detect the maneuver. Appropriate features extracted from sensor data can hence be processed to obtain estimates of the maneuver in order to facilitate the Kalman filter to continue to track the target reliably even when it is maneuvering. In the architecture depicted in Fig. 2, this function is implemented by a multilayer feedforward neural network.

### *3.2. Extraction of Features for Neural Network Training*

The efficiency with which target maneuvers can be tracked by the present neural network-assisted tracking scheme depends on the features used for training the network. Selection of appropriate features is guided by the observation that generally there are several entities (abrupt position changes in the x- and y-directions, intensity of acceleration, abrupt change in target heading, etc.) that help in getting a good estimate of target maneuver.

Training of the neural network for providing maneuver estimates is implemented in our scheme with a set of six feature vectors extracted from the target position measurements available from each sensor channel. These six features $\tilde{\nu}_1-\tilde{\nu}_6$ are listed briefly below:

$$\tilde{\nu}_1(k)=\frac{\tilde{z}_x^2(k)}{S_{xx}(k)},\quad \tilde{\nu}_2(k)=\frac{\tilde{z}_y^2(k)}{S_{yy}(k)} \tag{17}$$

where $\tilde{z}_x(k)$ and $\tilde{z}_y(k)$ are the components of the innovation vector $\tilde{z}(k)=z(k)-H\hat{X}(k/k)$, and $S_{xx}(k)$ and $S_{yy}(k)$ are the diagonal elements of the covariance matrix

$$\begin{aligned} S(k) &= HP(k/k-1)H^T+R \\ \tilde{\nu}_3(k) &= \alpha_{LT}(k)-\alpha_{LT}(k-1) \end{aligned} \tag{18}$$

where $\alpha_{LT}(k)$ and $\alpha_{LT}(k-1)$ are heading estimates at sampling instants $k$ and $k-1$ computed by the method of least triangles [22] from using three past data points,

$$\tilde{\nu}_4(k)=H_{rel}(k)=\frac{\alpha_{LT}(k)}{\theta(k)} \tag{19}$$

where $\theta(k) = \tan^{-1}\left(y/x\right)$ is the bearing angle,

$$\tilde{v}_5(k) = \Delta x(k) = x(k) - x(k-1)$$
$$\tilde{v}_6(k) = \Delta y(k) = y(k) - y(k-1). \tag{20}$$

It may be noted that $\Delta x(k)$ and $\Delta y(k)$ respectively denote the position changes in the $x$- and $y$-directions during successive sampling instants, while $\alpha_{LT}(k) - \alpha_{LT}(k-1)$ denotes the heading change during successive sampling instants. The relevance of the six features listed above for training a multilayer neural network which produces the estimates $\hat{\delta}_x(k)$ and $\hat{\delta}_y(k)$ as outputs, which in turn are used to update the Kalman filter estimate of the target state, is discussed in several places in the past [13,19,20] and hence will not be repeated here.

For a brief explanation motivating the selection of these signals, one may note that the first two features quantify the changes occurring in the innovation sequence data $\{\tilde{z}(k)\}, k$ = 1,2 ..... due to the maneuver. The use of this data to obtain inferences on target acceleration levels has been the most popularly used maneuver tracking approach since the work of Magill [24] which proposed using a bank of N parallel filters to match the changes in innovation sequence to acceleration levels. This relation, however, being a nonlinear one, our use of this data for neural network training can be interpreted as learning this nonlinear relation. The additional four features also serve to provide measures for other types of target maneuvers involving not only acceleration changes but also turns, sharp dives, etc.

Extraction of the six features described above from radar data is straightforward. The computation of the target centroid from each image frame provides data of a similar form on this sensor channel from which the above six features can be readily computed. For the sake of further discussion, let us denote by $\tilde{v}_{rj}$, j = 1,2,…,6 the six features extracted from the radar data stream, and by $\tilde{v}_{ij}$, j = 1,2,…,6, the six features extracted from the image sequence.

### *3.3. Neural Network Architecture and Training*

The neural network selected for implementation will be trained off-line by using the two sets of features $\tilde{v}_{rj}$ and $\tilde{v}_{ij}$, j = 1,2,…,6, as inputs and estimates of the maneuver-induced velocity deviations $\hat{\delta}_x(k)$ and $\hat{\delta}_y(k)$ as outputs.

Since the desired function to be implemented by the neural network is to approximate the mapping relation that exists between these input and output variables, a simple multi-layer perceptron network [23] will suffice. The input layer of the network will have six nodes for implementing the fusion architectures Arch 1 and Arch 2 (or twelve nodes for implementing Arch 3) and an output layer with two nodes. It must be

observed that the relations between the components of the velocity deviation vector $\delta^T(k) = [\delta_x(k) \quad \delta_y(k)]$ and those of the feature vectors $\tilde{v}_r^T = [\tilde{v}_{r1} \; \tilde{v}_{r2} \; \tilde{v}_{r3} \; \tilde{v}_{r4} \; \tilde{v}_{r5} \; \tilde{v}_{r6}]$ and $\tilde{v}_i^T = [\tilde{v}_{i1} \; \tilde{v}_{i2} \; \tilde{v}_{i3} \; \tilde{v}_{i4} \; \tilde{v}_{i5} \; \tilde{v}_{i6}]$ are in general complex nonlinear functions that are learned by the neural network from the training examples. The benefits of the model independent framework provided by the network are clearly evident.

The maneuver estimation performance of a neural network with one hidden layer comprising of 14 nodes with nonlinear activation functions $f(z) = (1 + e^{-z})^{-1}$ will be described in the next section. To simplify the processing, the two output nodes of the neural net were selected as linear nodes. The network was trained by processing 1120 training vectors generated with 20 levels of acceleration covering the range 0–20 m/sec$^2$, 14 levels of initial velocities covering the range 200–900 m/sec, and 4 levels of heading changes. It may be noted that since innovation data is used for the training, neural network learning of maneuvers takes place when combined with the Kalman filter as in the schematic shown in Fig. 2. The required sensitivity for detection of maneuvers can be built into the neural network by selecting an appropriate threshold for the declaration of maneuvers (for example, for a specified standard deviation of the measurements, say $\sigma_R$ = 100 m, the threshold can be set at an acceleration value of 1 m/sec$^2$).

A number of alternate procedures exist for training a neural network with the available data and different training algorithms usually yield different sets of interconnection weights. The training algorithm that is popularly used employs the error backpropagation approach [23]. While this approach is perhaps the most popular approach for training multilayer neural networks, it has a few shortcomings as well. The backpropagation approach, being a gradient-based search algorithm, is sensitive to the initial starting point (i.e. preliminary selection of weights to start the algorithm). Also because of the gradient-based search property, it is normally trapped by the first optimal point reached and has a tendency to converge to a local minimum. This is generally undesirable since it implies that the knowledge acquired by the network is not optimal. To counter this problem, modified backpropagation algorithms have been developed which include a momentum term that can kick the parameters out of sub-optimal solutions. However, with these algorithms one has to fiddle around with the momentum term and hope that, with the selected starting point, a globally optimal solution can be reached. In general, there is no guarantee of achieving a global optimum.

In our quest to improve the efficiency of the neural network learning, which we believe is critical in equipping the network with the knowledge required for reliably recognizing complex target maneuvers, training of the neural network is conducted by using the Linear Least Squares Simplex (LLSSIM) algorithm developed by Hsu et al. [25,19]. This algorithm employs concepts from simplex optimization and is conducted by splitting the 3-layer neural network into two portions – a linear portion and a nonlinear portion. The connections between the input layer and the hidden layer form the nonlinear portion, while the connections between the hidden layer and the output layer constitute the linear portion. The simplex optimization method is used to find the optimal weights in the nonlinear portion, while a linear least squares minimization is used to determine the optimal weights in the linear portion of the network [19,25]. For implementation in the present context, the algorithm can be designed with two distinct

stopping criteria. The search for the weights in a specified network structure can be terminated either when the size of the simplex is smaller than a pre-specified threshold or the number of iterations performed exceeds a preset threshold. More details on the implementation of neural network training can be found in [19].

## 4. Maneuver Tracking Performance of the Fused Two-Sensor System

Results from two simulation experiments will be described in this section. The first experiment was conducted to determine which of the three fusion architectures shown in Figs. 5a–c yields the most accurate tracking results and has the best potential for use in a fused tracking system. The second experiment compares the tracking performance of the 2-sensor fused system with that obtainable from another tracking system that processes range radar measurements only. The goal of this comparison is to determine the performance improvements resulting from the addition of the imaging sensor.

Due to the difficulty in obtaining real field data with both imaging and non-imaging sensors deployed in the same tracking system and also the complexity in generating realistic synthetic image measurements (for example, accounting for the change in the target's size, flap angles, orientation angles, altitude, etc., as a result of the maneuvers), the following conditions are assumed in the simulations:

1. image measurements arrive at a rate of 1 Hz;
2. radar measurements arrive at a rate of 0.333 Hz;
3. arrivals of image data and radar data are synchronized;
4. size of each image frame is 128x128 pixels;
5. target is flying at a constant altitude;
6. noise in image has the distribution $N(0,50^2)$;
7. target in image has the distribution $N(10,20^2)$;
8. background of image has the distribution $N(180,20^2)$;
9. the pixel intensities in the image vary from a minimum of 0 to a maximum of 255.

Furthermore, all experiments are designed with the same target initial conditions of 700 m/sec initial velocity at an initial position of (0 m, $2.0\text{x}10^3$ m) in Cartesian coordinates, and a flight path at an angle of $0^o$ with respect to the x-axis. It must be emphasized that the chosen initial conditions serve no other purposes than simplifying the experiment scenarios and setups, and hence do not limit the versatility of the tracking system. It may also be noted that the initial position and the initial velocity of the target are typically obtained from the track initialization phase and hence the initial states of the target are known fairly accurately. One may also note that the chosen sampling rates for the two sensors are different, which adds to the generality of the simulation experiments conducted. Hence, there are time instants when no range radar measurements are available whereas image sensor measurements become available, which in turn implies that at these time instants no updated features can be extracted out of the range radar measurements while the image sensor channel will have the features updated.

### *4.1. Experiment 1*

In this experiment the target executes a loop maneuver, i.e. a $360^o$ turn, at a turn rate of 15 deg/sec. With the target initially located at (0 m, $2.0\text{x}10^3$m) and moving at a con-

stant velocity of 700 m/sec while maintaining a $0^o$ heading, it executes a 15 deg/sec coordinated turn at the onset of the $13^{th}$ scan of the radar. This maneuver is maintained at the same intensity for the next 24 consecutive radar scans, after which it resumes its constant velocity motion. One may note that upon completing 24 consecutive turn maneuvers at 15 deg/sec, the target will have completed the loop maneuver.

The tracking performance delivered by the three fusion architectures is shown in Figs. 6a–c with both the true target track and the estimated track shown on the same figure.

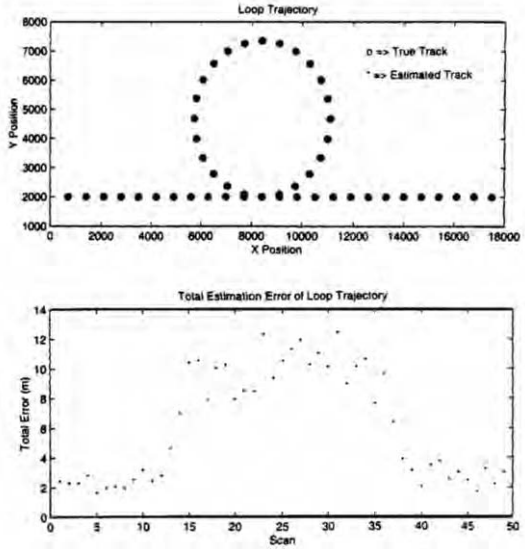



**Figure 6a.** True and estimated trajectories and plot of total position error for fusion scheme Arch 3.

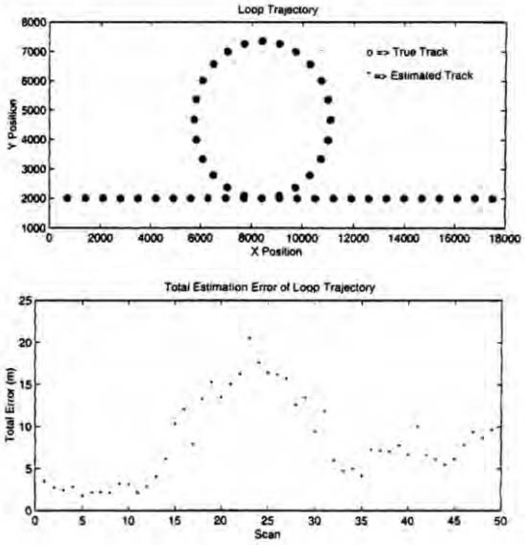



**Figure 6b.** True and estimated trajectories and plot of total position error for fusion scheme Arch 2.

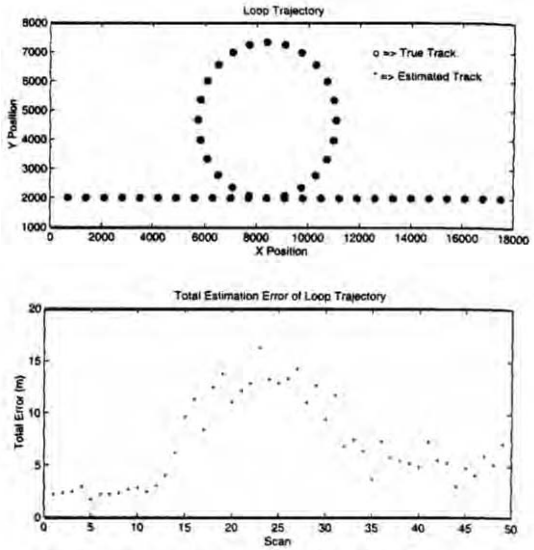



**Figure 6c.** True and estimated trajectories and plot of total position error for fusion scheme Arch 1.

Fig. 6a shows the result from employing the Arch 3 architecture that uses a combined set of 12 features for training the neural network. Comparing this result with that resulting from the other two fusion architectures Arch 2 (see Fig. 6b) and Arch 1 (see Fig. 6c), it is clear that Arch 3 gives the best results in terms of maximum peak error (it has a maximum total peak error of approximately 13 m). This outcome agrees with intuition. One may note that with the Arch 3 architecture, all information contained in the two sets of features is fed to the neural network for maneuver estimation. This is a lossless process since no form of combining of information is performed. In contrast, architecture Arch 1 requires the features to be linearly combined, which may be a lossy process (the loss arising from the averaging effect due to the linear weighting in the Combiner block), while architecture Arch 2 requires the outputs of the two neural networks to be linearly combined according to Eq. (7), which may also be a lossy process.

A comparison of the results from Arch 1 and Arch 2 are counterintuitive. Intuitively, one would expect that Arch 2 delivers better results than Arch 1. With Arch 2, the decisions reached by the two distinct neural nets are linearly combined according to Eq. (7). These decisions however are "optimal" in the sense that they are obtained with the full information available from the features derived from each sensor data. Hence, although the decisions from the individual neural nets are combined, the combination may be expected to be a more accurate estimate of the target maneuver than that estimated by Arch 1 which combines the extracted features prior to handing them over to the neural network.

An important statistic that should be recorded in a pointing and tracking system is the number of tracks lost. A track loss is declared by a tracking system when it loses the target. For all the tracking experiments performed in this work, a track loss is declared by the neural network-based tracking system if it fails to find the target during two consecutive scans. Note that when a target is lost for the first time, at the $k^{th}$ scan for example, its previous state estimate, i.e. $k-1^{th}$ state estimate, will be used to predict

the k+1 $^{th}$ state of the target. This in essence is to assume that the target did not execute any maneuver during this period. With this definition, architecture Arch 3 achieves 11% (with a peak error of approximately 13 m) of track loss while the architectures Arch 2 and Arch 1 achieve 14% (with a peak error of approximately 22 m) and 12% (with a peak error of approximately 16 m) track loss, respectively. Clearly Arch 3 is the superior architecture and hence will be the architecture of choice for implementing the fused two-sensor tracking system in the other experiments reported in this section.

### *4.2. Experiment 2*

In this experiment, the ability of the tracking scheme in following a concatenation of turn maneuvers executed in quick succession is tested. The target motion during this maneuver follows a S-shaped trajectory obtained from executing a 90$^o$ counterclockwise turn followed immediately by a 90$^o$ clockwise turn. The test scenario consists of a target initially detected at (0 m, 2.0x10$^3$m) with an initial velocity of 700 m/sec traveling at an angle of zero degree with respect to the horizon until the 13$^{th}$ scan of the radar at which a coordinated turn of 9 deg/sec was made for the next 10 scans (resulting in a 90$^o$ counterclockwise turn), followed immediately by another coordinated turn of –9 deg/sec for 10 scans (resulting in a 90$^o$ clockwise turn). At the end of the two turns, the target will have completed an S-shaped turn, after which it resumes its straight path motion.

The tracking performance delivered by the fused two-sensor tracking system during a period lasting 45 scans is shown in Fig. 7a.

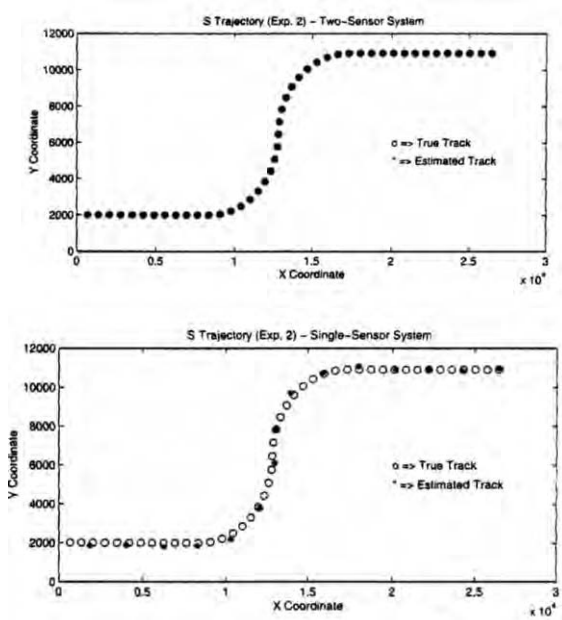



**Figure 7a.** True and estimated trajectories for the two-sensor and single-sensor systems in Experiment 2.

For comparison purposes, the tracking performance during the same period delivered by a single sensor system (using range radar only) is also shown in Fig. 7a. From these results it can be seen that the two-sensor system performs significantly better than the single sensor system. To further compare the performance, plots of the total positional error are shown in Fig. 7b. It can be seen that the single sensor system results in a maximum total error of approximately 300 m while the two-sensor system has an approximate maximum total error of only 13 m. The track loss for the two-sensor system is 9% while that for the single sensor system is 34%. It is clear that the fused 2-sensor system with the added image sensor measurements performs significantly better.

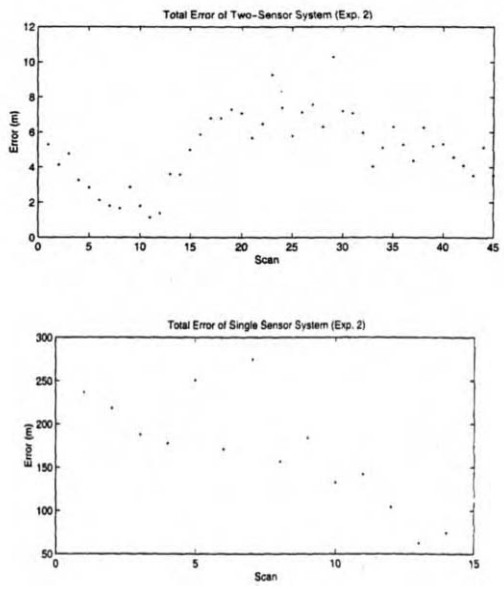



**Figure 7b.** Plots of total position errors for the single-sensor and two-sensor systems in Experiment 2.

## 5. Conclusions

The major contributions of this article are the development of a scheme for fusing image data with non-image data and the design of a tracking system architecture that efficiently integrates the pattern classification abilities of a trained neural network with the noise filtering and dynamical state estimation properties of a Kalman filter. The present approach to tracking of target maneuvers, which involves an innovative utilization of a trained neural network, offers several benefits each of which can be attributed to an intrinsic property of the neural network employed. In particular, some notable advantages resulting from the present approach to tracking are the following: (i) the pattern classification performed by the neural network provides a convenient way for a rapid detection and estimation of the maneuver at its onset, (ii) the nonlinear function approximation provided by the neural network facilitates converting the filtering and state estimation performed by a simple Kalman filter into a performance equivalent to that realizable from a more complex nonlinear filter without any attendant increases in im-

plementational complexity, and (iii) the fusion of input data streams performed by the neural network provides a natural and convenient way for implementing a feature-level fusion of measurements obtained from multiple dissimilar sensors.

## Acknowledgements

Research reported in this article was partially supported by the Office of Naval Research under grant #N00014-97-1-0965.

## References

[1] M.K. Sundareshan and F. Amoozegar, "Neural network fusion capabilities for efficient implementation of tracking algorithms," Optical Engineering, Vol. 36, pp. 692–707, March 1997.
[2] B.V. Dasarathy, Decision Fusion, IEEE Computer Society Press, Los Alamitos, CA, 1994.
[3] M.K. Sundareshan, "Neural network schemes for data fusion and tracking of maneuvering targets," Final Report to ONR (Grant # N00014-95-1-1224), April 1999.
[4] R.J. McAulay and E.J. Delinger, "A decision-directed adaptive tracker," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-9, No. 2, pp. 229–236, March 1973.
[5] P.L. Bogler, "Tracking a maneuvering target using input estimation," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-23, No. 3, pp. 298–310, March 1987.
[6] Y. Bar-Shalom and T.E. Fortmann, Tracking and Data Association, Academic Press: New York, 1988.
[7] Y. Bar-Shalom and X.R. Li, Estimation and Tracking: Principle, Techniques, and Software, Artech House: Norwood, MA, 1993.
[8] J.D. Kendrick, P.S. Maybeck, and J.G. Reid, "Estimation of aircraft target motion using orientation measurements," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-17, No. 2, pp. 254–260, March 1981.
[9] E. Oron, A. Kumar, and Y. Bar-Shalom, "Precision tracking with segmentation for imaging sensor," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-29, No. 3, pp. 977–987, July 1993.
[10] A.T. Alouani and S. Shetty, "A multi-sensor tracking system with image-based maneuver detector," Proc. IEEE, Vol. 83, pp. 1232–1236, 1992.
[11] D.D. Sworder, "Image-enhanced tracking," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-25, No. 3, pp. 701–710, 1989.
[12] D.D. Sworder and R.G. Hutchins, "Maneuver estimation using measurements of orientations," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-26, No. 4, pp. 625–637, July 1990.
[13] M.R. Anderberg, Cluster Analysis for Applications, Academic Press: New York, 1973.
[14] A.K. Jain and R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall: Englewood Cliffs, NJ, 1988.
[15] M.K. Sundareshan, S. Bhattacharjee, R. Inampudi, and H.Y. Pang, "Image preprocessing for improving computational efficiency in implementation of restoration and superresolution algorithms," Applied Optics (IP), Vol. 41, pp. 7464–7474, 2002.
[16] S.S. Shahpurkar and M.K. Sundareshan, "A novel K-means hierarchical clustering algorithm for efficient information extraction from large databases," Proc. of 2003 Int. Conf. On Information and Knowledge Engineering, Las Vegas, NV, June 2003.
[17] B. Bhanu, "Automatic target recognition: A state of the art survey," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-22, No. 2, pp. 364–379, July 1986.
[18] M.K. Sundareshan and T.J. Peterson, "Data fusion for imagery – Part 3: Design of fusion architectures for surveillance and tracking using information value mappings," Proc. of NATO ASI on Data Fusion, Armenia, August 2003.
[19] Y.C. Wong, Target tracking in a Multi-sensor Environment Using Neural Networks, Ph. D. Dissertation, University of Arizona, Tucson, AZ, 2000.
[20] Y.C. Wong and M.K. Sundareshan, "Equivalent velocity tracking model for estimation of target maneuvers and design of neural network-based tracking algorithms," Proc. of SPIE Conf. on Signal and Data Processing of Small Targets, Aerosense'98, Orlando, FL, April 1998.
[21] M.S. Grewal and A.P. Andrews, Kalman Filtering: Theory and Practice, Prentice-Hall: Englewood Cliffs, NJ, 1993.
[22] R.J. Evans, F. Barker, and Y.C. Soh, "Maximum likelihood estimation of constant heading trajectories," Proc. of IEEE Int. Radar Conf., London, Ontario, Canada, pp. 489–493, 1987.

[23] S. Haykin, Neural Networks; A Comprehensive Foundation, Macmillan: New York, 1994.
[24] D.T. Magill, "Optimal adaptive estimation of sampled stochastic processes," IEEE Trans. Automatic Control, Vol. AC-10, pp. 434–439, 1965.
[25] K.L. Hsu, H.V. Gupta, and S. Sorooshian, "Artificial neural network modeling of rainfall-runoff models," Water Resources Research, Vol. 31, pp. 2517–2530, 1995.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Super-Resolution of Tactical Surveillance and Tracking Data for Fusion of Images

## *Super-Resolution for Image Fusion*

Malur K. SUNDARESHAN and Supratik BHATACHARJEE
*Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ*

**Abstract.** Surveillance and monitoring operations directed to situation monitoring, incident detection, and security management often demand the capability for all-weather day-and-night sensing and data collection. Various missions such as reconnaissance, threat detection, landing and take-off of aircraft, covert deployment of special operations teams, detection and tracking of tactical mobile and extended area high-value targets, etc., are typical in their need for the ability to execute the mission in any set of contingencies. Development of sensors operating at different ranges of the electromagnetic spectrum has been the primary technological approach to provide this capability. Fusion of imagery collected from sensors operating at different ranges of the electromagnetic spectrum and integration of this data with information stored in databases facilitates improved surveillance and security management. A challenge that needs to be overcome however is that deployment within the same surveillance scenario of imaging sensors that may collect data at different resolution levels can pose major difficulties in designing appropriate fusion architectures and fusion logic for integrating these data forms. Consequently, a pre-processing of the images aimed at resolution enhancement prior to performing fusion is desirable. In this lecture, we shall describe a class of nonlinear image processing schemes, called super-resolution algorithms, that not only restore the spectral components within the passband (by reversing the effects of convolution with the point spread function of sensor) but also re-create through spectral extrapolation the frequencies lost due to the imposition of sensor diffraction limits. Since the loss of high frequencies during the sensing operation is the primary cause for the poor resolution in the acquired imagery, restoration of these frequencies holds the key to a satisfactory enhancement of these images and hence achieves resolution levels that permit fusion of this data with other available information. This talk will focus on three powerful and proven directions for a systematic design of image super-resolution algorithms. One of these employs a Bayesian estimation framework starting with a statistical modeling of the sensing process and constructs a maximum likelihood (ML) restoration. The second direction involves the use of set theory-based estimation methods and implements an iterative Projection-Onto-Convex-Sets (POCS) algorithm. The third direction involves attempting to combine the strong points of the two previous approaches resulting in possible hybrid implementations that facilitate design of optimal processing architectures and algorithms. Specific results of super-resolving imagery acquired from PMMW, IR, and SAR imaging sensors will be presented.

**Keywords.** Image fusion, resolution enhancement, image restoration, super-resolution

## 1. Introduction

No single sensing modality provides satisfactory imaging capabilities under all practical conditions that may be encountered in surveillance and tracking applications and there are a number of factors that prevent pristine images to be obtained for further image exploitation tasks (detection, classification, tracking, fusion, etc.). For instance, sensors such as radar, video camera, and laser radar (LADAR) are susceptible to adverse weather (such as fog) and environmental conditions (smoke and dust screens, for instance) resulting in poor quality images. Other sensors that form imagery by capturing thermal emissions from the target (such as infrared imagers), produce drastically degraded signals when a clear line-of-sight cannot be established to the target (such as when a cloud cover intervenes between the target and the imager). Yet other types of novel sensors, specifically those operating in the millimeter-wave ranges, can penetrate through cloud, smoke, dust, etc. (thus providing all-weather surveillance and tracking capability), but cannot provide good spatial resolution in the images formed. Additionally, imaging sensors used in tactical military applications will have also several limitations arising from the conditions of deployment such as platform vibrations, turbulence in the media, sampling and integration time constraints, etc. all of which pose hurdles in obtaining clear images of the scenes with the best resolution the imagers are capable of achieving.

In addition to all of the above, a major limitation also comes from the physical constraints on the size of the lens or the antenna, which in turn contribute directly to poor resolution in the acquired imagery. The diffraction-limited angular resolution, $\theta$, of an incoherent imaging system is given by

$$\theta = 1.22\frac{\lambda}{D} \tag{1}$$

where $\lambda$ is the effective wavelength of imaging and D is the diameter of the limiting aperture of the antenna or lens [1]. As $\lambda$ increases, the achievable angular resolution decreases, i.e. $\theta$, the size of the angle between two resolvable points, increases. It may be noted that the wavelength of a Synthetic Aperture Radar (SAR) sensor operating at 1 GHz is about 1 inch long and one needs an antenna as big as 40 ft wide in order to achieve a resolution requirement of being able to distinguish points in a scene separated by about 1 meter at a distance of 1 Km. For a typical Passive Millimeter-Wave (PMMW) sensor with a 1 ft diameter antenna and operating at 95 GHz, the angular resolution is only about 10 mrad, which translates into a spatial resolution of about 10 meters at a distance of 1 Km. Some recent studies have also established that for ensuring reasonably adequate angular resolution (typically of the order of 4 mrad), a 95 GHz PMMW imaging system with a sensor depression angle of $60^{o}$–$80^{o}$ needs to be confined to very low operational altitudes (of the order of 75–100 meters), which puts inordinate demands on the surveillance and guidance schemes to facilitate such requirements. Similar resolution limitations and the consequent requirements on operational conditions (some of which may be clearly impossible to satisfy for tactical missions with reliability and survivability constraints) exist for other types of sensing modalities as well.

A strategy that is fast becoming popular for overcoming the limitations of individual sensors is to deploy various types of imagers on the same observation platform and attempt to obtain good quality images under all conditions from fusing the different images captured. Due to technological advances (miniaturization, reduction in integration times, etc.) and reduced costs of sensors, it is now becoming possible to have a multitude of sensors on even small platforms (such as a small missile homing in towards a target). Unfortunately, fusion of images captured at different frequencies poses considerable practical difficulties mainly stemming from the differences in the image quality, most notably due to disparities in resolution levels. Consequently, prior to the execution of fusion operations, every image captured will necessarily undergo some preprocessing mainly directed to obtaining improved resolution in the processed images as shown in Fig. 1.

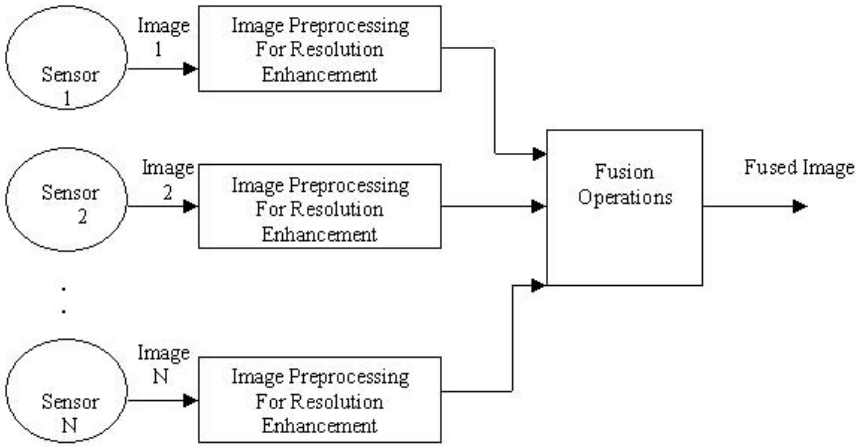



**Figure 1.** A schematic for image fusion.

The fundamental problem underlying the sensing operation is the "low-pass" filtering effect due to the finite size of the antenna or lens that makes up the imaging system and the consequent imposition of the underlying diffraction limits. Hence the image recorded at the output of the imaging system is a low-pass filtered version of the original scene. The portions of the scene that are lost by the imaging system are the fine details (high frequency spectral components) that accurately describe the objects in the scene, which are also critical for reliable detection and classification of targets of interest in the scene. Hence some form of image processing to restore the details and improve the resolution of the image will invariably be needed. Traditional image restoration procedures (based on deconvolution and inverse filtering approaches) attempt mainly at reconstruction of the passband and possibly elimination of effects of additive noise components [2]. These hence have only limited resolution enhancement capabilities. Greater resolution improvements can only be achieved through a class of more sophisticated algorithms, called super-resolution algorithms, which provide not only passband reconstruction but also some degree of spectral extrapolation, thus enabling the restoration of high frequency spatial amplitude variations relating to the spatial resolution of the sensor and lost through the filtering effects of the seeker antenna pat-

tern. A tactful utilization of the imaging instrument's characteristics and any a priori knowledge of the features of the target together with an appropriately crafted nonlinear processing scheme is what is what gives these algorithms a capability for super-resolving the input image by extrapolating beyond the passband range and thus extending the image bandwidth beyond the diffraction limit of the imaging sensor.

The principal focus in this article is to evaluate a few image processing algorithms that have demonstrated significant potential for resolution enhancement and super-resolution in the context of target surveillance and tracking applications. To provide an appropriate framework for the discussion of results, and for an explicit demonstration of resolution enhancements that can be achieved, we shall confine ourselves to imagery acquired in practice by tactical sensors. It must be emphasized that the challenges posed by the processing needs of tactical imagery data often are several times more significant than in other applications (such as in traffic management, computer vision for robot control, sensors deployed in surgical environments and other medical applications, etc.). Consequently, the needs for image processing in these scenarios demand greater capabilities than what the existing algorithms can offer, and typically require more enhanced procedures in order to achieve satisfactory restoration and super-resolution within the permitted time constraints. In this article, we shall outline three such enhancements, viz. parallel projection implementation with adaptive relaxation, use of scene-derived information for constraint set design, and a hybrid statistical and set-theoretic estimation procedure. The restoration and super-resolution performance of an iterative algorithm that incorporates these enhancements is illustrated by application to tactical imagery data (images acquired from state-of-the-art Synthetic Aperture Radar (SAR) and Passive Millimeter-Wave (PMMW) sensors), to demonstrate the preparation of these images for further fusion operations.

One item that deserves particular mention is that image fusion and image super-resolution constitute two processing tasks that have significant interrelations. On the one hand, fusing different images has been suggested [35] as a procedure for obtaining a super-resolved image; more correctly, a super-sampled image can be constructed from integrating two or more microscanned images (i.e. images of the same scene obtained by a slight translation or dithering of the sensor). Due to page limitations, we shall not at present explore this relation further. The relation in the other direction, viz. use of super-resolution techniques to better facilitate fusion of images, is the one that will be explored in this article.

## 2. Processing of Digital Images for Resolution Enhancement

### *2.1. Some Basics on Designing Iterative Algorithms for Restoration and Super-Resolution*

Image restoration and super-resolution have attracted the attention of researchers for a number of years not only due to the intellectual challenges posed in solving this "inverse problem" but also due to the tremendously large set of practical applications ranging from radioastronomy to medical imaging and industrial and military applications that can benefit from this work. Among the different approaches that have been followed for developing iterative algorithms for the restoration and super-resolution of practically acquired imagery data, two popular avenues have been statistical optimization methods and set-theoretic estimation methods.

Due to the ill-posed nature of the inverse filtering problem underlying image restoration and super-resolution objectives, it is necessary to have some a priori information about the ideal solution, i.e. the object f being restored from its image g. In algorithm development, this information is used in defining appropriate constraints on the solution and/or in defining a criterion for the "goodness" of the solution. The specific a priori knowledge that can be used may include the fundamental knowledge that the reflectivity of any point on the ground cannot be negative. In addition to the non-negativity constraint, a space constraint resulting from the known space-domain limits on the object of interest could be used. Other typically available constraints include level constraints (which impose upper and lower bounds on the intensity estimates $\hat{f}_j$), smoothness constraints (which force neighboring pixels in the restored image to have similar intensity values) and edge-preserving constraints. Varying by the extent to which a priori knowledge can be incorporated in algorithm development, there have been introduced into the literature a large number of image restoration approaches and algorithms too vast to describe or reference here. It is important to recognize that only a small subset of the approaches that are developed for image restoration have received some interest for their super-resolution capabilities, i.e. possible spectrum extrapolation performance. One may note that not all image restoration methods provide the capability for super-resolving. In fact, a majority of existing schemes may perform decent passband restoration, but provide no bandwidth extension at all.

The various approaches in general attempt to code the a priori knowledge to be used by specifying an object model or a set of constraint functions, and further employ an appropriate optimization criterion to guide in the search for the best estimate of the object. Recent research has established that an efficient utilization of the available a priori knowledge comes from iterative processing techniques [3,4]. As noted above, two popular avenues that have been followed in constructing iterative processing algorithms with demonstrable restoration and super-resolution performance are based on statistical optimization and set-theoretic estimation. A few major algorithms developed following these approaches will be briefly outlined in the following.

### *2.2. Statistical Optimization Methods*

One of the early attempts at employing a statistical framework for image enhancement was the application of prolate spheroidal wave functions to linear systems by Slepian and Pollack [5] in 1961. Following this came the Maximum Entropy (ME) approach pioneered by Frieden [6,7], which has led to several extensions, modifications, and implementations (eg., the MEMsys package popular with the astronomical community). Employing a Bayesian formulation, Richardson [8] and Lucy [9] independently developed an iterative algorithm that is simple to implement even for large-sized images. A generalized version of this approach, resulting in an Expectation-Maximization (EM) algorithm of statistical estimation, was presented in 1977 by Dempster, Laird and Rubin [10]. The EM algorithm is an iterative algorithm that is guaranteed to converge to the local maximum of the likelihood function given some "incomplete" data set. A version of this algorithm particularly suited for medical images (specifically, PET images) was developed by Shepp and Vardi [11]. Another modified version of this algorithm, which in some cases can construct significantly better estimates, is the maximum a posteriori (MAP) algorithm due to Hunt [12,13]. Development of this algorithm utilizes Poisson distributions for both likelihood and prior distributions in order to opti-

mize the posterior density. The landmark paper of Geman and Geman [14] that appeared in 1984 represents a major advance in Bayesian image restoration and offers the potentially rich theory of Markov Random Fields (MRFs) for constructing appropriate models for the prior. A closely related algorithm that also performs MAP restoration is the pixon method due to Pina and Puetter [15]. Among these various algorithms that originated from using statistical optimization approaches, the Maximum Likelihood (ML) algorithm (commonly referred to as the Richardson-Lucy iteration) has emerged as the most popular algorithm, particularly for military and industrial applications, due to the several advantages it enjoys (notably ease of implementation, guaranteed convergence, and quality of achievable restoration). More recently, several variants of this algorithm specifically suited for military applications (e.g., a blind ML algorithm that is useful for image restoration and super-resolution when the Point Spread Function (PSF) of the imaging process is not exactly known, as in the case of a tactical sensor mounted on a platform subject to vibrations) have been developed by Sundareshan and others [16–19].

For a precise description of the iterative steps used by the ML algorithm to construct the super-resolved image, let f(x) denote the object's intensity function, $x \varepsilon X$, where X defines the region over which intensity is defined, and let g(y) denote the intensity detected in the image, $y \varepsilon Y$, where Y defines the region over which intensity is detected. If {h(y,x), $y \varepsilon Y$ and $x \varepsilon X$ } denotes the point spread function (PSF) of the imaging sensor, then accounting for the presence of noise in the imaging process, one can model the imaging process by the convolution model

$$g(y) = \sum_{x \in X} h(y,x) f(x) + noise \tag{2}$$

(where an additive noise is assumed for the sake of simplicity). The classical restoration problem is to find the object intensity estimate { $\hat{f}(x)$ } given the data {g(y)}. The ML algorithm attempts to obtain a maximum likelihood (ML) estimate { $\hat{f}(x)$ }, i.e. the object intensity estimate that have most likely created the measured data {g(y)} with the PSF process {h(y,x)}. This estimate is in turn developed by maximizing an appropriately modeled likelihood function $p(f/g)$, i.e. by solving the statistical optimization problem

$$f = \arg\max_{f} p(f / g). \tag{3}$$

A particularly simple iteration algorithm can be obtained by modeling $p(f/g)$ as a Poisson distribution. For a discretized formulation of the imaging equation (2) obtained by a lexicographic ordering of the object, image and sensor PSF, with g(j) and f(j), j = 1,2,…,N, denoting the N pixels of the image and the object respectively, and h(j) denoting the PSF of the sensor, the updating of the object estimates takes the form

$$\hat{f}_{k+1}(j) = \hat{f}_k(j) \left[ \left\{ \frac{g(j)}{\hat{f}_k(j) \otimes h(j)} \otimes h(j) \right\} \right], j = 1,2,3,...,N \tag{4}$$

where k denotes the iteration count and $\otimes$ denotes discrete convolution. The initial estimate $\hat{f}_0(j)$ is taken as the image g(j) to commence the iteration.

### *2.3. Set-Theoretic Estimation Approaches*

Another direction for image restoration and super-resolution that is attaining greater popularity in recent times is the use of set theory-based methods that attempt to construct the image estimate iteratively by enforcing known constraints. Perhaps the earliest known super-resolution procedure of this type was published by Gerchberg [20] and later by Papoulis [21], which has come to be known as the Gerchberg-Papoulis algorithm.

This algorithm attempts to exploit any available a priori information on the object $f$ to be estimated for constructing spatial and spectral constraints on the estimate $\hat{f}$, which in the algorithm implementation are alternately applied once in the space domain and next in the frequency domain in order to progressively correct the image estimate.

While set-theoretic estimation has a long history, it is the introduction of a new iterative procedure for an organized synthesis of set theoretic estimates, which has now come to be known as Projection Onto Convex Sets (POCS), that marked the beginning of the interest in this approach and secured its steady growth. Although the POCS framework was first developed by Gubin et al. [22], and was further expanded by Lent et al. [23], a set of three pioneering papers by Youla and others [24–26] was responsible for making this approach popular among the signal processing community. A comprehensive discussion on this approach together with several useful extensions can be found in a recent tutorial paper by Combettes [27]. In the field of image processing, more interest in this approach has so far been shown by medical imaging researchers in handling such problems as tomographic image reconstruction [28].

The primary objective in set-theoretic estimation is to employ each piece of prior information $\Psi_i$ that may be known to intelligently guide the estimation process. For a mathematical description of the image restoration procedure employing this idea, starting once again with the discretized model of the imaging process given by Eq. (2), the primary objective is to model the known information $\Psi_i$ as a closed convex set $S_i$ and seek an estimate $\hat{f}$ vector (for the sake of simplicity, the argument denoting the pixel location will be deleted in the following discussion) of the object by solving the optimization problem

$$\hat{f} = \arg\min, \sum_{i=1}^{M} w_i J_i(\hat{f}, S_i). \tag{5}$$

In Eq. (3), M denotes the number of closed convex sets employed (corresponding to the number of known information sets $\Psi_i$ utilized in the restoration processing), $w_i$ are a set of weights (indicating the relative importance of the individual sets to be used as constraints in guiding the restoration process), and $J_i(\hat{f}, S_i)$ denotes a proximity measure that determines how well the current estimate satisfies the $i^{th}$ constraint. The

proximity measure can be uniquely defined for a closed convex set $S_i$ modeling the $i^{th}$ constraint as

$$J_i(\hat{f}, S_i) = \left\| \hat{f} - f_p \right\|^2, \tag{6}$$

where $f_p = P_i(\hat{f})$ denotes the projection of $\hat{f}$ onto the set $S_i$.

Some discussion of the parallel that exists between this approach and the statistical optimization method described earlier is useful to note the similarities and differences. Both approaches attempt to solve the estimation problem underlying the image restoration objective as a mathematical optimization problem; while the statistical approach attempts to maximize a chosen statistical measure (likelihood, for instance) under the constraint of known information, the set-theoretic approach attempts to minimize the weighted proximity measure (which essentially models the distance of the estimate from the known properties that the correct estimate should satisfy) under the constraint of known information.

The first step in applying the method of POCS to an image recovery problem is to define a closed convex set for each of the a priori constraints in such a way that the members of the set are consistent with the associated constraint and each set contains the actual image distribution. An estimate of the image distribution is then defined to be any member of the intersection of the constraint sets. Finding an estimate by POCS is then equivalent to the problem of finding a point in the intersection of a number of closed convex sets.

Suppose that we have several a priori known constraints that can be associated with closed convex sets $S_i$, i = 1, 2,…, M, and let their respective projection operators be denoted $P_i$. Then the estimate $\hat{f}_n$ generated at the $n^{th}$ iteration by a sequential application of the projections on the previous iterate $\hat{f}_{n-1}$, that is

$$\hat{f}_n = P_M(P_{M-1}(P_{M-2}.....(P_1(\hat{f}_{n-1}))..)) \tag{7}$$

converges to an estimate $\hat{f}$ in the intersection set $S_0 = S_1 \cap S_2 \cap S_3 .....\cap S_M$ It is further demonstrated that a faster convergence to an estimate $\hat{f}$ lying within the intersection set can be achieved in some cases if, instead of using the projection operators $P_i$ in the construction of the estimate, one uses "relaxed projections" $T_i$, i = 1, 2,…, M, defined by

$$T_i = (1-\lambda)I + \lambda P_i \tag{8}$$

where $\lambda$ is a parameter suitably selected within the range $0 < \lambda < 2$, and the estimate $\hat{f}$ is constructed iteratively by implementing the algorithm

$$\hat{f}_n = T_M(T_{M-1}(T_{M-2}.....(T_1(\hat{f}_{n-1}))..)) . \tag{9}$$

As pointed out in the literature [26,28], an appropriate value of the relaxation parameter $\lambda$ needs to be used for realizing improved convergence, and the choice of this parameter is rather sensitive to noise. It may be noted that for the value of the relaxation parameter $\lambda$=1, $T_i$ reduces to the ordinary projection $P_i$.

## 3. Enhanced Procedures and More Efficient Algorithms

While the two approaches outlined in the last section offer two parallel directions for synthesizing iterative algorithms for restoring and super-resolving degraded images, the challenges presented by tactical imagery data in practice, such as poor inherent resolution in acquired images, limits on computational time, large image formats (example, data formed by focal plane arrays as in the Third Generation Forward Looking Infrared (FLIR) sensors presently under development), etc. often demand more efficient procedures to be used. In this section, we shall briefly describe three recently developed enhancements that can be employed within the framework offered by the two approaches described above.

### *3.1. Method of Parallel Projections with Adaptive Relaxation*

In the traditional implementation of the POCS approach, the enforcement of constraints is in a sequential manner (as described in Eqs. (7) and (9)), which may in practice require larger computation times, particularly if a number of constraint sets are available for use in guiding the restoration process. Such an implementation fails to take advantage of a parallel processing architecture, since during any given iteration each constraint set will be acted upon one at a time. A more efficient execution will involve employing a parallel implementation procedure in which the projections onto the different constraint sets can be simultaneously executed and the next iterate is computed as a weighted sum of these projections. One version of this approach, popularly referred to as the Method of Parallel Projections (MOPP) [30,31], has been found in our investigations to possess attractive convergence properties. In this scheme, an elementary iteration consists in projecting the current estimate simultaneously onto selected sets and forming a relaxed convex combination of the projections.

For a brief description of the method, given the image to be restored, denoted as the starting image $\hat{f}_0$ for commencing the POCS iterations, and two numbers $\varepsilon$ satisfying $0 < \varepsilon < 1$ and M a positive integer, MOPP implements the recursion

$$\hat{f}_{n+1} = \hat{f}_n + \lambda_n \left[ \sum_{i \in I_n} w_i P_i(\hat{f}_n) - \hat{f}_n \right] \tag{10}$$

together with the following conditions:

1. $\varepsilon \le \lambda_n \le 2-\varepsilon$ and
2. $\sum_{i\in I_n} w_i = 1$, where $0 \neq I_n \subset I$ and $I \subset \bigcup_{k=0}^{M-1} I_{n+k}$, with I denoting the set of constraints used.

The control sequence $(I_n)_{n\ge 0}$ dictates which sets are to be activated at the $n^{th}$ iteration. Naturally, for the iterate to converge, appropriate conditions must be imposed in order to ensure that every set is activated repeatedly. However, one can employ the simple static control case, where all the sets are activated at each iteration. It can be shown in this case that using over-relaxed projections (i.e. relaxed projection $T_i$ given by Eq. (8) with $\lambda > 1$) can lead to a faster convergence. Choosing an appropriate value of $\lambda$ can be quite difficult, however. At first sight, it may seem obvious to choose a value of $\lambda$ close to 2. Although this selection may initially prove beneficial, as the iterate approaches the final solution, a large value of $\lambda$ may not necessarily be an optimal choice. Thus, an adaptive selection of $\lambda$ based on the rate of change or the gradient of the proximity function is a recommended approach. A specific algorithm that provides an adaptive relaxation of the projections for use within a MOPP framework is outlined in the following table. The key idea behind this implementation is to initially start with a large value of the relaxation parameter (say, $\lambda$ = 1.999) and at each iteration compare the gradient of the proximity function with its value at the next iteration, and progressively reduce the value of $\lambda$ by a certain percentage (say, 25%) until the difference between the gradients is more than a preset value. Details of a specific implementation are given in Table 1 below.

**Table 1.** Steps in implementation of parallel projection algorithm with adaptive relaxation

1. *Choose an initial guess* $\hat{f}_0$, *M constraint sets, and a set of weights* $w_i$ ( $\hat{f}_0$ = *g, the acquired image to be processed, will serve as a good initial estimate).*
2. *Set n = 0,* $\lambda_n$ = *1.999, and* $\nabla\Phi(\hat{f}_n) = \hat{f}_n - \sum_{i=1}^{M} w_i P_i(\hat{f}_n)$.
3. *Set* $\hat{f}_{n+1} = \hat{f}_n - \lambda_n \nabla\Phi(\hat{f}_n)$.
4. *If* $\Phi(\hat{f}_n) - \Phi(\hat{f}_{n+1}) < \frac{1}{2}\lambda_n \left\|\nabla\Phi(\hat{f}_n)\right\|^2$, *set* $\lambda_n = 0.75\lambda_n$ *and return to step 3*
5. *Set n = n+1.*
6. *Repeat steps 2–5 for N iterations.*
7. *Save processed image.*

### *3.2. Scene-Derived Information Sets for POCS Restoration*

Fundamental to the reliable estimation of the high spatial frequencies that provide an expansion of the image bandwidth is the utilization of a priori known information dur-

ing the processing steps. Indeed, since image restoration is inherently an ill-conditioned inverse problem, it is long realized that the quality of restoration and the extent of achievable super-resolution depend on the accuracy and the amount of a priori information the processing scheme can successfully employ. In practice, one finds that a lot of information typically exists to assist in guiding the restoration process in a satisfactory direction. This includes, (i) Information about the solution, such as non-negativity of pixel intensities, spatial extent of objects present in the scene, known ranges for signal intensities, etc., (ii) Information about the imaging system, such as sensor phenomenology, aperture size and the resulting shape of the modulation transfer function (MTF), etc., and (iii) Information about the imaging conditions, such as the extent of observation noise, object motion during imaging, etc. These are information sets that have traditionally been used in the development of image restoration algorithms. For tactical imagery applications, however, one needs to typically employ more information, which can be readily extracted from the image being processed. To distinguish from the other types of information, we will call this scene-derived information. Extraction of such information and formulation of a constraint set for use in a POCS restoration algorithm will be outlined below.

In several scenes of interest, objects of interest will be superimposed as foreground against a primarily flat background. Thus, a sharp transition in gray level intensity exists in the actual scene imaged. Due to blurring caused by the sensor, this sharp edge will be lost and image energy from the foreground region flows into the background pixels, thus making it difficult to decide exactly where the foreground ends and the background begins. Thus, if one can determine the spatial extent of the object, or the border pixels bounding the object, one can sharpen the edge between the foreground and the background. In the spectral domain, this sharpening corresponds to recovering the higher frequency components, and hence super-resolution. The starting point in the implementation of this process is to identify the border pixels, which can be simply performed by any edge detection algorithm, a popular one being the Canny algorithm [32] (which implements a gradient-based edge detection procedure). Once the set of border pixels is identified, one can proceed with the second stage of the border information modeling which involves modeling the border pixel information as a convex constraint set and defining a corresponding projection operator. The constraint set is defined as one that enforces a background-foreground separation, and can be mathematically formulated as

$$S_{border} = \{\text{foreground of image f is bounded by } \xi\} \tag{11}$$

where $\xi$ denotes the set of border pixels. The projection operator associated with this set exists provided that the set is closed and convex, which can be easily ensured if we are dealing with solid convex shapes. The projection operator is then given by

$$P_{border}(f) = f_p \text{ where } f_p(i,j) = \begin{cases} f(i,j) : (i,j) \text{ lies inside } \xi \\ 0 \quad\quad : (i,j) \text{ lies outside } \xi. \end{cases} \tag{12}$$

### *3.3. Hybrid Statistical and Set-Theoretic Approaches*

As discussed in Section 2, the two popular approaches that have been followed for design of iterative image processing algorithms are statistical optimization and set-theoretic estimation. Each of these parallel approaches enjoys specific advantages. Unfortunately, each approach also suffers from specific limitations arising from the mathematical development of the iterative algorithm (for instance, use of a simple probability distribution function (such as Poisson distribution) for modeling the likelihood function when the underlying emission process may not be accurately described by such a model), which in turn places a limit on the restoration and super-resolution achievable in practice. Superior restorations of highly degraded images that are typical in tactical applications require the design of hybrid algorithms that can efficiently combine the advantages of statistical optimization methods and set-theoretic estimation methods. In the following, we shall outline the design of a hybrid algorithm that attempts to leverage the strong points of both the ML iteration scheme (simplicity of execution, known convergence properties, etc.) and the POCS adjustments (utilization of diverse types of information in guiding the restoration process, faster decay of restoration error, etc.).

In this algorithm, which will be called POCS-assisted ML algorithm, the iterative processing is characterized by the implementation of the ML iterations given by Eq. (4) as the main image restoration scheme while including a POCS adjustment of the ML estimate for enforcing constraints after every cycle of ML iterations is completed. More specifically, one executes L iterations of POCS adjustment after each cycle of K iterations of the ML algorithm. Thus, each iteration cycle of the combined algorithm applies an ML iteration cycle comprising K number of ML iterations followed by a POCS adjustment cycle comprising L number of POCS iterations. The combined algorithm is run for a total of N iterations. It is easy to see that this algorithm in effect performs a constrained maximization of the likelihood function in an iterative manner. The implementation details are summarized in the following table.

**Table 2.** Steps in implementation of POCS-assisted ML algorithm

| | |
|---|---|
| 1. | *Read the blurred image data (g).* |
| 2. | *Set initial conditions to commence ML iterations,* $\hat{f}_0 = g.$ |
| 3. | *Perform ML restoration implementing the updating algorithm given in Eq. (4).* |
| 4. | *Compute the likelihood and* $l_2$ *-norm values at the end of each iteration.* |
| 5. | *Repeat steps 3–4 for K iterations.* |
| 6. | *Perform POCS adjustment using MOPP scheme and implementing updating algorithm given in Eq. (10).* |
| 7. | *Repeat step 6 for L iterations.* |
| 8. | *Repeat steps 3–6 for N iterations.* |
| 9. | *Save processed image.* |

## 4. Demonstration of Resolution Enhancement Performance

It should be emphasized that the three distinct enhancements outlined in the last section are not mutually exclusive. Although each one can be used independently to augment

an iterative restoration algorithm, they can be combined in a single algorithm to considerably enhance the efficiency of the restoration and super-resolution scheme. In particular, the POCS-assisted ML algorithm discussed above in Section 3.3 is sufficiently versatile to incorporate the strengths of the two other enhancements outlined in Sections 3.1 and 3.2. One may note that the implementation of the MOPP scheme in step 6 of the iterative algorithm described in Table 2 can be performed with adaptive relaxation using the steps described in Table 1. Furthermore, the flexibility offered by the POCS adjustment iterations permit the inclusion of scene-derived information sets, such as the border constraint set defined in Eq. (11). A combination of the three enhancements outlined in the last section hence provides the necessary strengths required for a satisfactory processing of tactical imagery in practice. As an illustration of this performance, in this section we shall present restoration results from processing two-types of digital imagery data acquired from state-of-the-art tactical sensors. In each case, a comparison with possible restorations achievable from an existing popular algorithm will be given.

**Figure 2a.**

**Figure 2b.**

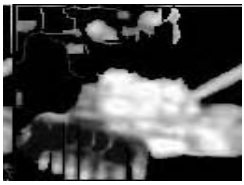


**Figure 2c.**

**Figure 2.** Acquired PMMW image and its super-resolved versions after processing with ML and POCS-assisted ML algorithms (2a. Acquired image; 2b. Image processed by ML algorithm; 2c. Image processed by POCS-assisted ML algorithm).

Fig. 2a shows a passive millimeter-wave (PMMW) image ("Tank Image") of size 96×64 pixels acquired from a state-of-the-art single detector radiometer with 1 ft diameter aperture operating at 95 GHz. Fig. 2b shows the restored image obtained at the end of 100 iterations of the ML algorithm described by Eq. (4). While the resolution enhancement achieved is noteworthy, the computation time required for the execution of 100 iterations could be rather excessive to support tactical applications. The result of processing the same image with the POCS-assisted ML algorithm using 5 ML iterations alternating with 5 POCS adjustment iterations implemented for 3 cycles is shown in Fig. 2c. For implementing the POCS adjustments a border constraint set as well as the adaptive relaxation scheme for executing MOPP were used. It may be noted that the resolution is considerably enhanced as evidenced by the improvements in the wheels, the gun barrel, and the top of the tank. It is also of interest to note that due to the use of the background-foreground separation implemented by the enhancement discussed in Section 3.2, the foreground (tank) has been sharpened while the background (trees and sky) is less emphasized. The algorithm is quite powerful in producing an overall image from which better features of the object of interest can be extracted than from the original image for any further analysis such as target recognition, sensor fusion, or aimpoint selection. However, what is more significant also is that these improvements are realized together with a substantial saving in computations, since the computation time in obtaining the processed image shown in Fig. 2c is roughly equivalent to that required for executing 15 iterations of the ML algorithm. This should be compared with the processed image shown in Fig. 2b, which is the result after the execution of 100 ML

iterations (without the presently developed enhancements), in order to appreciate the improved efficiency resulting from these enhancements. For a detailed discussion on quantitatively estimating the computational time requirements for executing ML iterations (given in Eq. (4)), one may refer to Sundareshan et al. [33].

The second experiment involves the application of the same algorithm to a SAR image taken from the MSTAR database [34]. Fig. 3a shows the image of a T-72 tank in a vegetation background (obtained from the file hb03333.015.tif extracted from the image archive tatate15t7sn_132.tar). The poor inherent resolution in the image is particularly noteworthy as it provides a challenge to any super-resolution processing algorithm. An application of 25 iterations of the ML algorithm described by Eq. (4) results in the image shown in Fig. 3b. While the resolution has improved somewhat, the generation of processing artifacts in the background is not desirable and could make the execution of further analysis steps rather difficult.

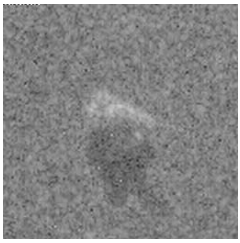


**Figure 3a.**

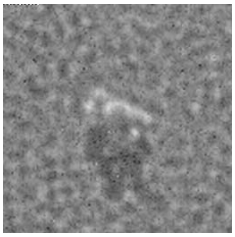


**Figure 3b.**

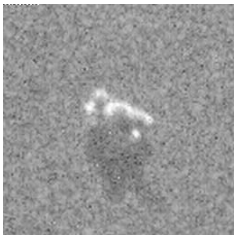


**Figure 3c.**

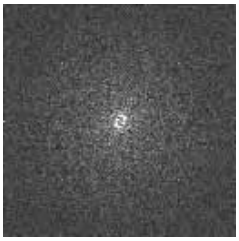


**Figure 3d.**

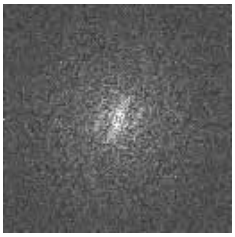


**Figure 3e.**

**Figure 3.** Acquired SAR image and its super-resolved versions after processing with ML and POCS-assisted ML algorithms (3a. Acquired SAR image; 3b. Image processed by ML algorithm; 3c. Image processed by POCS-assisted ML algorithm; 3d. Spectrum of acquired image; 3e. Spectrum of restored image).

Fig. 3c shows the image obtained from processing with 5 cycles of the POCS-assisted ML algorithm, each cycle comprising of 3 iterations of the ML algorithm followed by 3 iterations of the POCS adjustment process. The resolution enhancement together with the suppression of artifacts is clearly evident. To illustrate the extent of super-resolution achieved from the iterative processing, the frequency spectra of the original image (in Fig. 3a) and the processed image (in Fig. 3c) are shown in Figs. 3d and 3e respectively. The spectrum extrapolation provided by the POCS-assisted ML algorithm is clearly remarkable. Of greater interest to fusion applications is that a more reliable set of features can be extracted from the image in Fig. 3c for combining them with information extracted from other types of imagery obtained for the same scene.

## 5. Conclusions

Preprocessing aimed at enhancement of resolution in data acquired from imaging sensors will almost always be needed prior to subjecting these images to any fusion operations. Use of existing algorithms for restoration and super-resolution poses several practical challenges that include poor inherent resolution in acquired data (which in turn requires that a sufficient number of iterative steps be executed in order to gain desired resolution) and large data formats and limitations on permitted computation times (which precludes satisfactorily completing a sufficient number of processing iterations). Consequently, these procedures need to be fortified with enhanced image processing steps that provide capabilities for efficiently processing the data in tactical applications. Three such enhancements that are simple to execute, viz. parallel projection implementation with adaptive relaxation, use of scene-derived information for constraint set design, and a hybrid statistical and set-theoretic estimation procedure were described in this article. Illustration of the restoration and super-resolution performance of an iterative algorithm that incorporates these enhancements was given by application to realistic imagery data acquired from tactical sensors (viz. images acquired from state-of-the-art Synthetic Aperture Radar (SAR) and Passive Millimeter-Wave (PMMW) sensors). These results conclusively demonstrate the utility of the present enhancements to achieve resolution gains in the processed imagery before fusing them with other data, especially in surveillance and tracking applications.

## Acknowledgements

Research reported in this article was partially supported by a grant from the Air Force Office of Scientific Research (AFOSR Grant # F49620-00-1-0167). Support for this research from the AFRL Munitions Directorate at the Eglin AFB, FL is also gratefully acknowledged.

## References

[1] J. Goodman, Introduction to Fourier Optics, McGraw-Hill, 1996.
[2] K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, 1989.
[3] M.K. Sundareshan, "Performance of iterative and noniterative schemes for image restoration and super-resolution processing in multispectral seeker environments," RDL Report SREP-96-0844, Air Force Office of Scientific Research, Bolling AFB, 1997.
[4] J.D. Silverstein, "Passive millimeter-wave image resolution improvement by linear and nonlinear algorithms," Proc. of 2001 SPIE Conf. on Passive Millimeter-Wave Imaging Technology, Orlando, Florida, Vol. 4373, pp. 227–238, April 2001.
[5] D. Slepian and H.O. Pollack, "Prolate spheroidal wave functions, Fourier analysis and uncertainty – I," Bell Systems Tech. Journal, Vol. 40, pp. 43–62, 1961.
[6] B.R. Frieden, "Band-unlimited reconstruction of optical objects and spectral sources," Journal of Optical Society of America, Vol. 57, pp. 1013–1019, 1967.
[7] B.R. Frieden, "Restoring with maximum likelihood and maximum entropy," Journal of Optical Society of America, Vol. 62, pp. 51–55, 1972.
[8] W.H. Richardson, "Bayesian-based iterative method of image restoration," Journal of Optical Society of America, Vol. 62, pp. 55–60, 1972.
[9] L.B. Lucy, "An iterative technique for the rectification of observed distributions," Astronomy Journal, Vol. 79, no. 6, pp. 745–759, June 1974.
[10] P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal of the Royal Statistical Society – Series B, Vol. 39, pp. 1–38, 1977.

[11] L. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," IEEE Transactions on Medical Imaging, Vol. 1, pp. 113–122, 1982.
[12] B.R. Hunt, "Bayesian methods in digital image restoration," IEEE Transactions on Computers, Vol. C-26, pp. 219–229, march 1977.
[13] B.R. Hunt, "Super-resolution of images: Algorithms, principles and performance," Int. J. of Imaging Systems and Technology, Vol. 6, pp. 297–304, 1995.
[14] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," IEEE transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, pp. 721–741, 1984.
[15] R.K. Pina and R.C. Puetter, "Bayesian image reconstruction: The pixon and optimal image modeling," Pub. of Astronomical Society of the Pacific, Vol. 105, pp. 630–637, 1993.
[16] H.Y. Pang, M.K. Sundareshan, and S. Amphay, "Super-resolution of millimeter-wave images by iterative blind maximum likelihood restoration," Proc. of 1997 SPIE Conf. on Passive Millimeter-Wave Imaging Technology, Orlando, Florida, Vol. 3064, pp. 227–238, April 1997.
[17] M.K. Sundareshan, H.Y. Pang, S. Amphay, and B. Sundstrom, "Image restoration in multi-sensor missile seeker environments for design of intelligent integrated processing architectures," Proc. of 1997 SPIE Conf. on Image Reconstruction and Restoration, San Diego, California, August 1997.
[18] H.Y. Pang, M.K. Sundareshan, and S. Amphay, "Optimized maximum likelihood algorithms for super-resolution of passive millimeter-wave imagery," Proc. of 1998 SPIE Conf. on Passive Millimeter-wave Imaging Technology, Vol. 3378, pp. 148–160, Orlando, FL, April 1998.
[19] M.K. Sundareshan and P. Zegers, "Role of oversampled data in super-resolution processing and a progressive upsampling scheme for optimized implementations of iterative restoration algorithms," Proc. of SPIE Conf. on Passive Millimeter-wave Imaging Technology, Aerosense'99, Orlando, FL, April 1999.
[20] R.W. Gerchberg, "Super-resolution through error energy reduction," Optica Acta, Vol. 21, pp. 709–720, 1974.
[21] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation," IEEE Trans. on Circuits and Systems, Vol. CAS-22, pp. 735–742, 1975.
[22] L.G. Gubin, B.T. Polak, and E.V. Raik, "The method of projection for finding the common point of convex sets," USSR Comput. Math. and Math. Physics, Vol. 7, No. 6, pp. 1–24, 1967.
[23] A. Lent and H. Tuy, "An iterative method for extrapolation of band-limited functions," Journal of Math. Anal. and Applications, Vol. 83, pp. 544–565, 1981.
[24] D.C. Youla, "Generalized image restoration by the method of alternating orthogonal projections," IEEE Transactions on Circuits and Systems, Vol. 25, No. 9, pp. 694–702, 1978.
[25] D.C. Youla and H. Webb, "Image restoration by the method of convex projections: Part 1. Theory," IEEE Transactions on Medical Imaging, Vol. MI-1, No. 2, pp. 81–94, 1982.
[26] M.I. Sezan and H. Stark, "Image restoration by the method of convex projections: Part 2. Application and numerical results," IEEE Transactions on Medical Imaging, Vol. MI-1, No. 2, pp. 95–101, 1982.
[27] P.L. Combettes, "The foundations of set theoretic estimation," Proc. of IEEE, Vol. 81, No. 2, pp. 182–208, 1993.
[28] H. Stark (Ed.), Image Recovery: Theory and Application, Academic Press, San Diego, 1987.
[29] M.I. Sezan, "An overview of convex projections theory and its application to image recovery problems," Ultramicroscopy, Vol. 40, pp. 55–67, 1992.
[30] G. Crombez, "Image recovery by convex combinations of projections," Journal of Mathematical Analysis and Applications, Vol. 15, pp. 413–419, 1991.
[31] P.L. Combettes and H. Puh, "Iterations of parallel convex projections in Hilbert spaces," Numerical Functional Analysis and Optimization, Vol. 15, pp. 225–243, 1994.
[32] J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, pp. 372–381, 1986.
[33] M.K. Sundareshan, S. Bhattacharjee, R. Inampudi, and H.Y. Pang, "Image preprocessing for improving computational efficiency in implementation of restoration and super-resolution algorithms," Applied Optics: Information Processing, Vol. 41, pp. 7464–7474, 2002.
[34] "M-STAR (Public) Targets: T-72, BMP-2, BTR-70, SLICY," see http://www.mbvlab.wpafb.af.mil/public/MBVDATA.
[35] R.H. Vollmerhausen and R.G. Driggers, Analysis of Sampled Imaging Systems, SPIE Press: Bellingham, WA, 2000.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Design of Fusion Architectures for Surveillance and Tracking Using Information Value Mappings

## *Information Value Mappings for Data Fusion*

Malur K. SUNDARESHAN and Timothy J. PETERSON
*Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ*

**Abstract.** Distributed data fusion architectures and algorithms are of particular interest in the design of surveillance and monitoring systems that are used for incident detection, situation monitoring, and security management functions. The fusion details in these systems are often predicated by the types of sensors chosen for deployment and their characteristics (which may include sensor cost, robustness, redundancy, physical size limitations, and the processing requirements/limitations they may introduce). For instance, a typical surveillance and monitoring scenario might include a multitude of rather inexpensive sensors that are randomly distributed over the geographical area under surveillance, and might include several vibration sensors (that detect seismic activity), acoustic sensors, RF energy sensors (using omni-directional antennae), in addition to conventional imaging sensors. Individually these sensors may not enjoy a high degree of reliability or possess very extensive performance capabilities; however, when deployed in a large number they can form a distributed sensor network (even forming an *ad hoc* network) capable of collectively providing a fused performance that fulfills the objectives of the specific application. Fusion architectures and algorithms for such applications require novel concepts and methodologies that are significantly different from available methods (most of which were developed during the past decade in the context of military applications involving a handful of expensive and high capability sensors such as radar systems). In this article, we shall describe our recent work in the design of such architectures and algorithms focusing on the concept of "information value maps." In assessing a fused sensor system, one considers the quality of the system architecture most often by the capabilities of the individual sensors and the attributes of the fusion algorithm. Though it is possible to evaluate system performance in an idealized context and model real world perturbations as random disturbances, it may be advantageous to treat predictable events as deterministic. Additionally, the a priori information one may have need not be limited to constraints on the objects of interest but also can be applied to constraints on the scene in which the object resides. In general, the environment scanned by a set of real world sensors is not homogeneous with regard to sensor performance. The qualitative effect of the non-homogeneous environment could be quantified through assignment of a value, termed information value (IV), that corresponds to the amount of trust associated with a sensor's measurement when observing a particular location and further to describe information value maps, which provide a graphical representation of ordered collection of IV assignments. A few general guidelines for the development of information value maps in surveillance and monitoring applications will be outlined in this article.

**Keywords.** Information value, a-priori information, sensor limitations, scene analysis, fusion architectures

## 1. Introduction

When considering the use of various sensors in a distributed system, designers choose sensors that best suit the needs of the application. Those needs may include system robustness, redundancy, monetary cost, physical size limitations and processing limitation fitness. The general claim of sensor fusion is that by correctly combining the measurements of several sensors, the system synergy produces a "better" result than that which would have been achieved by the best sensor [1,2]. The hope is that the statistical nature of sensors does not change so drastically during an on-line operation such that the employed fusion algorithm no longer fulfills the sensor fusion claim.

Recent works have expressed the need to manage the sensor fusion process with the possibility of sudden and drastic changes in sensor responses [5,6]. Indeed, ideas on how to deal with predictable and unpredictable changes in measurement variances are established. Besides abrupt changes in measurement statistics, there is the possibility that artifacts that can be sensed may be confused as the object of interest. When this happens, the measurement statistics may not change, and like the path-crossing problem in the multiple target tracking scenario, this is not readily noticeable in the measurement. Also, an aspect that has not been given adequate attention in the past is that changes in sensor measurement processing may occur while the fusion system is on-line. For example, the image from an imaging sensor may be super-resolved before being processed for fusion. This particular case is of specific interest to researchers at present [3] and will be briefly explored in Section 5.

An alternate, and perhaps novel, approach is to treat predictable changes in measurement statistics, predictable confusion, and predictable effects of processing changes as subclasses of a larger form of a priori information, herein called information value. The term arises from the notion that while sensors are a source of information about the environment, they do not always collect information of equal or consistent value. In many applications, such as in the design of surveillance and tracking systems, one will have a lot of a priori information on the sensors that will be deployed, on the environment within which these sensors will be operating, the expected performance from them, and the constraints and limitations that will degrade their performance from these expected levels, etc. The only uncertainties that make surveillance and tracking problems difficult to handle are the actions induced by a target (that enters the environment from outside) and unforeseen changes to the environment that make the actual operational conditions for the sensors somewhat different from the conditions that were planned for. It is hence logical to seek quantitative methods by which one can exploit the known a priori information in the fusion of data collected by the various sensors. The notion of information value that will be described in this article provides a mechanism for achieving this goal.

The structure of the present article is as follows. Section 2 contains some general definitions and brief examples of events that cause information value changes. In Section 3, the information value concept is more formally introduced and some characteristics of its mapping are given. Section 4 contains general guidelines to be followed in building information value maps in specific applications. The use of information value maps for designing fusion architectures to implement a multi-sensor tracking system is discussed in Section 5. A simple demonstration of information value usage is given there as well. Finally, the article concludes with a summarizing discussion in Section 6.

## 2. Predictable Events and Available Information

Generally, the environment scanned by a set of real world sensors is not homogeneous with regard to sensor performance. Real sensors have limitations that may vary from one type to another, thereby exhibiting weaknesses at different locations in the sensed scene (environment). This discussion follows the idea that the sensed scene is spatial, but the analogies could still hold for non-spatial environments (e.g., stock market). There may be stationary obstructions, which at some point may hide the object of interest. These obstructions may also cause confusion. These may not change the sensors' measurement statistics (esp. variance) but may cause the system to falsely attribute the obstruction's properties (location, texture, etc.) to the object of interest. It may be noted that such changes (example, texture-related) are of particular importance in surveillance and tracking scenarios supported by imaging sensors.

Measurement statistics may vary spatially as well. Disturbances may cause patches or pockets to exist where objects at any adjacent locations are clearly viewed but inside the pocket the view of the object is distorted, or covered in noise (low SNR). Performance may degrade progressively with range as in the case of resolution of imaging sensors. Range limitations could be more abrupt, as with radars, where it is primarily due to the length of the pulse repetition interval (PRI) [4]. Examples of weak spots for particular classes of sensors include infrared sensors looking at a horizon over the water (high background contrast), radar against ground clutter (background noise), video looking into shady areas (low luminosity), and fading (weak signal, range limitation).

In a number of signal processing tasks (such as in image enhancement, image compression, image retrieval, etc.), it is widely recognized that the processing efficiency and the quality of processing results depend on the accuracy and the amount of a priori information the processing scheme can successfully employ. In practice, one finds that a lot of information typically exists to assist in guiding the individual steps of a processing algorithm in a satisfactory direction. This includes, (i) Information about the signals, such as non-negativity of pixel intensities, spatial extent of objects present in the scene, known ranges for signal intensities, etc., (ii) Information about the sensor that collected the signal, such as sensor phenomenology, aperture size and the resulting shape of the modulation transfer function (MTF), etc., and (iii) Information about the signal measurement conditions, such as the extent of observation noise, object motion during imaging, etc. In addition, as shown in our earlier work [14], a variety of information can be extracted from the signal being processed; in image processing applications, such information can be generally called scene-derived information, and include information that can be extracted from some elementary operations such as extraction of borders of specific objects present in the image, matching templates, etc. In the exploitation of the available information for achieving the signal processing goals, the rule – "more the merrier" generally holds. Utilization of additional and diverse types of information consequently provides a mechanism for cross-validation during the execution of processing steps. Employing all of the available information does not come without a price, however. More information used during the iterative processing steps almost always translates into increased computational complexity and poorer convergence speed of the signal processing algorithms. It is easy to see that if two different types of information that are desired to be utilized are not appropriately modeled for inclusion in the algorithm (as an example, two information sets modeled such that they do not have a proper intersection), it will have detrimental effects on the convergence of the algorithm. Consequently, information analysis and information modeling is a

critical step before subjecting the signal measured by a sensor to other processing operations, such as fusion with other data. The assignment of information value and the use of information value maps, as will be described in this article, offer organized mechanisms for translating the a priori known information to gain efficiency in fusion operations.

As noted in the introduction, signal processing techniques may change the effectiveness of a sensor's measurement. If a fusion architecture is designed specifically for the expected performance of a sensor, changes in that performance may not necessarily be accounted for. That is to say, that the fusion architecture could be static with regard to expected sensor performance. In reality, improvements in performance may enhance the value of the measurement given, and likewise, degradations in performance may degrade the measurement's value. Improvements such as noise filtering and resolution enhancement may come at the cost of sampling rate and field of view reduction. Consideration of the object of interest's location may play a role in determining the usefulness of a processing operation. Hence, due to the additional processing, the value of information from a sensor may be improved at some locations, but degraded at others. The monitoring of the value can be a means of deciding when to switch a processing function on or off, if that capability exists. For example, super-resolution algorithms are useful in enhancing the resolution of a blurred image, but at the cost of additional computational load [9]. In a tracking application this computational load may be bearable when the target (object of interest) is distant and its change in bearing is small. However, when the target is closing in, its rate of bearing change may be too fast for the extended measurement update interval that comes with the processing.

## 3. Information Value Maps

The qualitative effects of the non-homogeneous environment could be quantified through the assignment of a value that corresponds to the amount of trust associated with a sensor's measurement when observing a particular location. The location need not be physical, but can be any unique measurable condition of the object of interest. Throughout the remainder of this discussion, we shall refer to such an assigned value as the information value.

The notion of information value should be distinguished from confidence level. Information value is a more general term than confidence level. Confidence level is directly related to the relative amount of error that a measurement possesses. It appears that information value and confidence level are identical when defining the amount of regard a sensor's measurement receives solely in the presence of disturbances. However, information value also captures the effects of processing on fusion performance. The super-resolution example cited earlier demonstrates clearly that even if our confidence level in the measurement of a "close" object is good, the usefulness of that measurement is tainted by its latency.

Information value should also be distinguished from an attribute recently used in the sensor fusion literature called perceivability, which is essentially the degree to which an object is perceived by a sensor or system [6]. Perceivability is assigned according to some predetermined threshold. It is not a priori information, but recursively calculated in order to decide whether to terminate track maintenance.

### *3.1. Assignment of Information Value*

Information value (IV) is the value assigned to a location in the sensed environment based on the perceived or predicted performance of a sensor. The information value, r, when defined as a scalar, lies between zero and one. Here, r =1 indicates the sensor measurement has full trust, and r = 0 indicates complete distrust, i.e.

$$\begin{aligned} &0 \le r \le 1, \\ &\text{with} \quad r = 0 \Leftrightarrow \text{complete distrust} \\ &\text{and} \quad r = 1 \Leftrightarrow \text{complete trust} \end{aligned} \tag{1}$$

When defined in the vector case the information value r is the vector of the same dimension as the measurement vector z for a sensor. This is the case where a sensor measures more than one attribute of the target. The rationale for allowing r to be multi-dimensional comes with the idea that not all measurements from one sensor will be affected the same way by a disturbance. The elements of r are the predicted values of a sensor's measurement value to the system. Each of the elements in the vector r satisfies the definition of Eq. (1). For precision, consider a target motion model employed by many existing tracking algorithms, given by a linear system whose state vector x is determined by the equation

$$\dot{x} = Ax + Bu + w, \quad x \in \Re^n, u \in \Re^p$$

where x represents the state (position and velocity variables in two- or three-dimensions), u is some external input, and w represents an additive white-gaussian-noise (WGN). The measurement z obtained by a sensor may be a partial subset of the state and is defined here as the linear combination

$$z = Hx + Gy + v, \quad z \in \Re^m, m \le n \tag{2}$$

where v is the measurement noise, y is an n-dimensional artifact in the scene, and H and G are the weighting matrices for the object of interest and scene artifact respectively.

H and G are related by the constant $H_0$ such that $H + G = H_0$. In this case, r would be an m-dimensional vector, and as will be shown, is a function of v, H, G, and y.

The assignment of information value can be made differently in various cases depending on the source of the information and the trust the designer places on a specific piece of information. In the following, we shall discuss some specific cases that are of interest in surveillance and tracking applications.

#### *3.1.1. Sensor-Based Information Value*

When Eq. (2) is void of any scene-based disturbances then G = 0, and H = $H_0$. In this case, Eq. (2) becomes simply

$$z = H_0 x + v \tag{3}$$

If $H_0$ is known, the only source of error in Eq. (3) is the variance of v. A suitable assignment of r that would satisfy Eq. (1) can then be made as

$$r_j = \frac{c}{c + \sigma_j^2}, \qquad j = (1,2,\ldots m) \tag{4}$$

where $\sigma_j^2$ is the predicted variance of the $j^{th}$ element of v. The scalar c is a parameter that is optimized for the specific application.

*3.1.2. Scene-Based Information Value*

When the measurement in Eq. 2 is corrupted by an artifact in the scene, the weighting matrix $G \neq 0$. Effects solely from the artifact can be approximated by neglecting the variance in v. The artifact is given by the state vector y, and is established as part of the mapping process. The predicted response to the artifact results in the weighting matrices H and G. If at any time H = 0, the artifact will be called an obstruction. From the conditions above, Eq. 2 can be written

$$z = Hx + Gy \tag{5}$$

Here the information value depends on the error introduced by the artifact and the sensor's response to it. If the artifact and the sensor's response are known with great certainty, i.e. Gy is accurately known, then the error will be small since Hx can be determined. There could be large errors if H = 0, since x is then hidden.

Consider for instance two measurements $z_1$ and $z_2$, denoting respectively the measured range and bearing of a target in a typical surveillance environment. Then consider the case when an obstacle is mistaken for the object of interest because the object went behind the obstacle (H= 0). Let the position of the object at time t be given in polar coordinates as $(d_x(t),\theta_x(t))$, and the position of the obstruction is constant at $(d_y,\theta_y)$. Assume the obstruction is narrow so that its angular width ($\Delta\theta$) is zero. Additionally, assume that the sensor does not regard the obstruction until time t such that

$$G(t) = H_0, \quad G(t^-) = [0]$$

which means it changes instantaneously. (Recall that the sum of H and G is $H_0$). So, when $\theta_x(t)= \theta_y$ and $d_x(t) \geq d_y$, and neglecting v, then the measurement z can be written from Eq. (5) as

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = H_0 y = \begin{bmatrix} d_y \\ \theta_y \end{bmatrix} = \begin{bmatrix} d_y \\ \theta_x(t) \end{bmatrix}$$

and the resulting mean squared error is $\sqrt{(d_x(t) - d_y)^2}$ .

Assuming that the only source of error is y when it acts as an obstruction, then an expression that satisfies Eq. (1) for the elements of vector r would be

$$\begin{aligned} &\text{when } (\theta_y - \tfrac{1}{2}\Delta\theta) \le \theta \le (\theta_y + \tfrac{1}{2}\Delta\theta) \text{ and } d \ge d_y \,, \\ &\text{then } \quad r_1 = \exp(-a(d - d_y)) \quad \text{and} \quad r_2 = \exp(-b(\Delta\theta)) \end{aligned} \tag{6}$$

where $\Delta\theta$ is the angular width of obstruction, $\theta_y$ is the center angle of the obstruction and a and b are arbitrary constants to be set in order to optimize a particular application.

### *3.1.3. Signal Processing Based Information Value*

For any particular application, the cost of additional signal processing may vary spatially (measurement space) as well. Any measurement that may be well conditioned and have high sensor-based and scene-based information values could still be ill-suited for the fusion process. Additional processing time may be more costly when the lengthened sampling period results in aliasing at points where rates of change are high.

In summary, the IV is a form of a priori information that relies mostly on the sensor's known capabilities and the particular setup of the environment in which it is placed. It is asserted that a sensor's IV is a possible way of predicting the regard a particular sensor should have when the object of interest is in a particular location. As discussed earlier, the information value can be characterized as having a sensor-based part and a scene-based part. Combining the two requires a consideration of the application. One way to get a single r for a location is to select the smaller of the two parts.

In general, the fused decision $X_{\text{fused}}$ in a given application will be not only a function of the conglomeration of the individual sensor measurements (z), their statistics (V), and dynamic relationships (A), but will also include the information value (r), as described below.

$$X_{fused} = f(z, V, A, r) \tag{7}$$

### *3.2. Information Value Maps and Their Characteristics*

The ordered collection of IV assignments form a map whose dimension is higher than the dimension of the sensed environment by a factor equal to the dimension of r (one in the scalar case). For example, if the environment is two dimensional and the IV is a scalar, the map would be 3 dimensional. The combinations of the sensor IV maps could provide an indication of the predicted system performance.

For any specific suite of sensors assigned to a given environment, one can develop a methodology for assigning the IV; however, in order not to be too specific, the discussion will be given only in general terms in this article. The building of the map is a separate function from the application and can be accomplished by a detection operation or assignment from some other form of higher intelligence, while the system is off-line.

Several characteristics of the IV maps can be stated at this point. The map may be static, continuously updated or periodically updated. The static case describes the event where the map is built off-line and left alone when the fusion operation is taking place. The continuous and periodic updating cases describe the rewriting of the map on-line. There are various reasons to consider map updating. These include changing environmental conditions, changing processing methods, and map correction or fine-tuning.

The map can be stored as a look-up table or a function approximating the desired map. The former may require much memory while the latter may be more complex to

calculate. In the look-up table approach, the number of different values that can be assigned will describe the number of map levels, M. If only two values are assigned, then a binary level assignment is used, whereas if the values are continuous then the interval between levels is infinitesimally small. The relationship between r and M can be specified as

$$r_i = \frac{i}{M-1}, \quad i \in \{0,1,2 \ldots M-1\} \tag{8}$$

where $r_i$ denotes the $i^{th}$ level of r. These levels are used to discretize the results of Eqs. (4) and (6).

## 4. Building the Information Value Map

### *4.1. Some Basic Steps*

The steps to building an IV map are given in the following. As indicated earlier, since the discussion is intended to be general, the steps will be broadly outlined and the details within each step will be omitted. Each step may be individually optimized depending on the application.

a) determine system capabilities and limitations. This step should consider the number of information levels needed (M), and the resolution with which the scene is partitioned. Intrinsic system weaknesses such as limited memory and memory access time should be kept in mind when determining how to build the map;
b) partition the scene into regions that will be labeled with a single information value. If adjacent values are the same then the regions can be joined to form a larger one;
c) detect and determine weak areas and artifacts in the scene. This could be phrased as an entire detection problem in itself or treated with foreknown generalities, such as sensor range capabilities;
d) assign a sensor based IV and a scene based IV to each region. A combination of the two information value parts should be adapted for the particular application. A formal approach to deciding what degradation is more costly (a greater weakness) should be defined;
e) a transition boundary is where two regions of unequal IV meet. According to Eq. (8), the transition would be instantaneous if the number of levels, M, is finite. Perhaps this crisp transition is undesirable or not optimal. Alternate transition modes will be discussed later;
f) a decision should be made as to whether or not to combine individual sensor maps into an overall system map, since this could affect the overall system architecture.

### *4.2. Transition Boundaries*

When an object of interest (target) traverses from one level of IV to another, it crosses a transition boundary. The manner by which these transition boundaries are treated is

an issue of particular interest. Sudden changes in a sensor's weighted output could cause undesirable transients. Moreover, there may be reasons to have particular transition boundary types to enhance some desirable trait.

Three types of boundaries are proposed here, the crisp transition boundary, the smoothed transition boundary, or the fuzzy transition boundary. The three types of proposed transitions are shown in Fig. 1. The figure is a cross section of the information value map. As seen in this figure, the crisp transition boundary is the unaltered state, the smoothed transition boundary requires some interpolation.

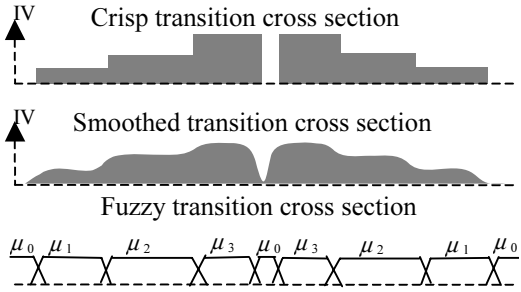



**Figure 1.** Transition boundary types.

### *4.2.1. Smoothed Transitions*

The smoothed values may be calculated and stored before the map is used. For the 2-D environment, the map could be stored as a gray-level image where the intensity is inversely proportional to the information value, i.e. $r_{M-1}$= "black," and $r_0$ = "white." Then the map image could be blurred to any desired degree. This would require that the number of output levels of the mapping device is substantially greater than M.

Another smoothing method could be to perform a moving average of the IV values, which would require a simple filtering of the mapping device's output. For example, in the digital domain the information value for time k could be determined with simple FIR filtering of the mapping device's output r, such as

$$f(k) = \frac{1}{n+p+1}\begin{bmatrix} \hat{r}(k+p) + \hat{r}(k+p-1) + \cdots \\ \cdots + r(k) + r(k-1) + \cdots \\ \cdots + r(k-n+1) + r(k-n) \end{bmatrix} \tag{9}$$

where p and n are integers ≥ 0 and the terms with p are predicted values based on estimation of the object's future locations (therefore bearing the "^"). This would be simplified by letting p = 0, but then predictive attributes would no longer exist.

### *4.2.2. Fuzzy Transitions*

Sub-maps for each of the M information values could be kept in order that fuzzy logic could be applied to the final information value assignment. Each sub-map would be the membership function of an information value level, such that $\mu_i(x)$ is the degree of belonging of location x to fuzzy set $r_i$. This would allow any location to be a member of up to M fuzzy sets. For example, a location could have a 0.5 degree of belonging to a full trust set ($r_{M-1}$ = 1) and a 0.5 degree of belonging to another set (e.g. $r_i$ = 0.7).

The use of fuzzy sets could be beneficial in treating the situation when the object of interest "disappears" behind an obstacle. This would be the case when transitioning from any set $r_i$ to an $r_0$ set, (esp. when i >1). It would be possible to design algorithms that incorporate the information value membership functions and membership functions of other system parameters to form a more comprehensive rule base. An example rule would be: if the object of interest is fast and low-maneuvering, and whose location has information values of $r_{M-1}$ and $r_0$, then the likelihood of correctness for location estimation employing a linear motion model is good. In the preceding example, all fuzzy sets are identified in italics.

### *4.3. A Two Dimensional Mapping Example*

This section presents an example of an information value map generated using the guidelines of Section 4.1. The example shown here illustrates the effects of three weaknesses that may be present in the scene. They are: degradation of sensor trust due to increasingly poor range performance, a region of high noise for unspecified reasons, and the effects of an impenetrable object (sensor is not capable of looking past it).

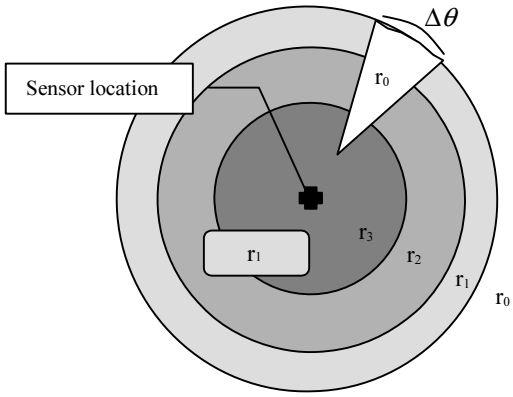



**Figure 2.** An example IV map.

Figure 2 shows the results of the mapping according to the following conditions. Information value is assigned according to some agreed upon metric. The map is stored in a look-up table, and the number of levels is set to M = 4. An artifact with angular width Δθ is detected in the upper right quadrant. The sensor's response to this artifact is negligible until it becomes an obstruction. A region of partial distrust (noisy, occluded) is detected in the lower left quadrant. The range capabilities of the sensor are considered to decrease steadily due to resolution limitations. These resolution characteristics are used to set the values resulting in concentric circles centered about the sensor.

In assigning information values, the obstruction is considered to obliterate any trust of the sensor from the area behind it, i.e. parameters a and b in Eq. (6) are very large. The region of partial distrust is considered as an occlusion and measurements within are considered less trustworthy than those outside it. The transition boundary is left crisp. Information values will be equally spaced according to the definition in Eqs. (1) and (2), i.e. $r_0 = 0.00, \quad r_1 = 0.33, \quad r_2 = 0.67, \quad r_3 = 1.00.$

## 5. Information Value Maps in Fused-Sensor Systems

### *5.1. Implementations of IV Mapping in a Fused Target Tracking System*

For the sake of discussing the details, let us consider an illustrative application scenario of a target tracking system supported by more than one sensor, as discussed in [13]. The basic idea behind adapting a multisensor target tracking system to utilize the information value concept is summarized in two cases shown in the block diagrams of Figs. 3, and 4. The leads from the sensors to the IV mapping block represent the function of the sensors in building and/or maintaining the map.

Case 1: Fig. 3 depicts a two-sensor high-level fused system. This case is considered a sensor-to-sensor track fusion architecture in the literature on target tracking, and involves combining the estimates produced by each sensor's tracking filter at every measurement update cycle, whereas the previous state estimate of the fused system is not used. For the sake of brevity, other cases, such as the sensor-to-system track fusion architecture are not presented here. For an informative discussion on the difference between the various architectures, one may refer to [10].

In the specific architecture considered, the decisions of each tracking filter ( $\hat{x}_1$ and $\hat{x}_2$ ), i.e. the target state estimates, are fused to produce a system state estimate, $\hat{x}_f$ . The fusion mechanism could be as simple as computing the weighted average,

$$\hat{x}_f = (r_{S1}\hat{x}_1 + r_{S2}\hat{x}_2)(r_{S1}r_{S2})^{-1} \tag{10}$$

where $r_{S1}$ and $r_{S2}$ are the retrieved information values for sensor 1 and sensor 2 respectively (scalar case).

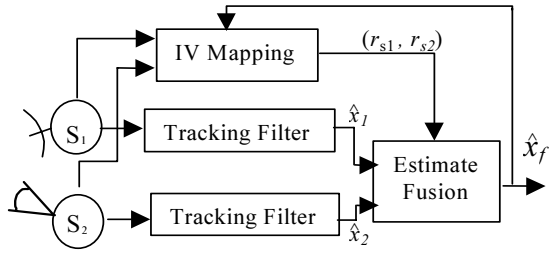



**Figure 3.** A high level fused tracking system incorporating IV Map.

Observe that there are two sensors, and each measures the same single state of x. Recalling Eq. (4), the parameter c that would best suit each information value r would be the predicted variance of the other sensor. In other words,

$$r_{s1} = \frac{\sigma_{s2}^2}{\sigma_{s2}^2 + \sigma_{s1}^2}, \quad \text{and} \quad r_{s2} = \frac{\sigma_{s1}^2}{\sigma_{s2}^2 + \sigma_{s1}^2} \tag{11}$$

Another fusion mechanism that could be used in this situation is the "winner takes all" method which is described by

$$\hat{x}_f = \hat{x}_j, \quad j = \arg_i \max(r_{Si}) \tag{12}$$

Case 2: In this case, lower level fusion occurs before processing by the tracking filter. The data from the two sensors are fused and the state estimation is based on the fused measurement. Several methods for this type of fusion are discussed in the literature and include neural network methods, weighted averaging, coordinate warping and wavelet techniques.

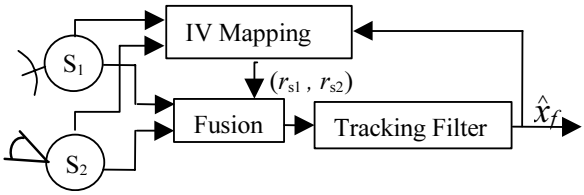



**Figure 4.** A low/medium level fused tracking system incorporating IV map.

Evidently, the goal of this architecture is to improve the fused measurement given to the tracking filter. One aim of including the information values is to decrease the uncertainty in the fused measurement by reducing the influence of the less valuable sensor. Another goal is to keep untrustworthy but statistically well-behaved measurements from one sensor from corrupting trustworthy but statistically poor measurements of another.

### *5.2. An Illustrative Fusion Experiment*

In this section, we shall give a simple demonstration of fusing measurements from two sensors as in the architecture depicted in Fig. 4. Consider the scenario: two co-located (at the origin) sensors that each measure the Cartesian location of the target that horizontally traverses at a constant velocity maintaining a constant y value of 10 units while covering a range of –100 units to 100 units in the x direction.

Response of Sensor One: The first sensor has a two dimensional measurement response according to Eq. (2) where v is zero mean normally distributed with an identity covariance matrix. The obstacle centered at a position (–5,5) with an angular width of half a radian, causes the term G = I when the bearing of the target $\phi$ is in the interval $(3\pi/4) \pm 0.25$. G=(0) elsewhere.

Response of Sensor Two: The second sensor has a measurement noise v that is zero mean and normally distributed with a covariance matrix d*I where d is proportional to the Euclidean distance of the target from the sensor location. The obstacle is invisible to sensor two, i.e. G=(0) always. The sensor responses and the target path are shown in Fig 5.

Information value map: For simplicity, let the number of map levels, M, be arbitrarily large and let Eq. 11 be used for points where G = 0, for points $G \neq 0$ let a and b in Eq. 6 have an arbitrarily large value. The assigned information values along the target path are shown in Fig. 6 below. Crisp transition boundaries are used for the sake of simplicity. Note the discontinuous jump to zero in the first sensor's information value. This is the prediction that the measurement will be unacceptably corrupted by the obstacle.

Comparison of two simple fusions: For the first method, no a priori knowledge is used and the two sensor measurements are directly averaged. For the second, a weighted average in the form of Eq. (10) is employed. Fig. 7 depicts plots of the MSE of the fused measurements. Note the direct averaging MSE in the top plot, which

shows the disturbance caused by the large variance of sensor two at far distances, and also the error caused by the obstacle seen by sensor one. The weighted average method (lower plot) compensates for discrepancies in sensor variation and undesirable responses to scene obstacles.

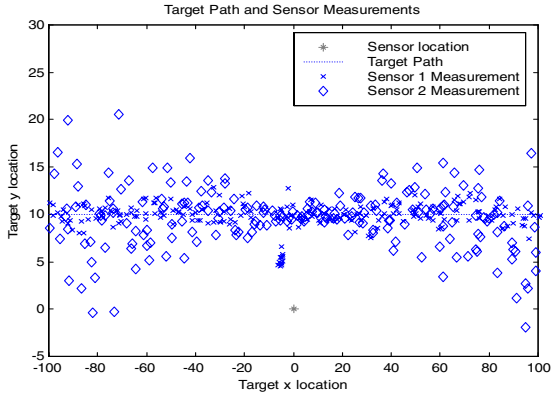



**Figure 5.** The target path and sensor measurements.

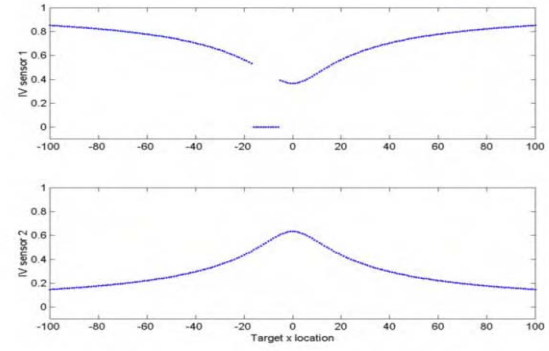



**Figure 6.** Information values along the target path.

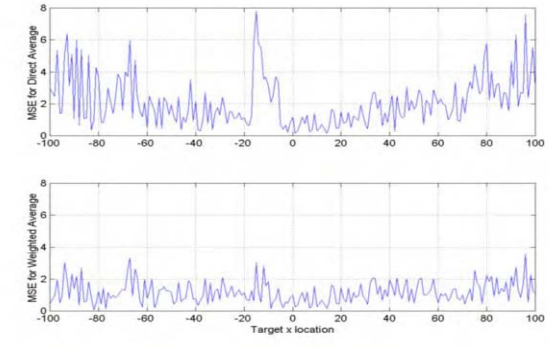



**Figure 7.** Resulting mean squared error of fused position measurements.

### *5.3. Modified IV Mapping Due to Signal Processing*

Modified information value maps can be constructed to accommodate improvements to sensor measurements due to processing, as alluded to in Section 3.1. One example is the changes to information values when an image acquired from a sensor undergoes a resolution enhancement using digital processing methods such as super-resolution processing. Super-resolution algorithms are designed to perform not only passband restoration of the captured image but also to extrapolate the frequency content in the image that suffers attenuation due to diffraction limited imaging operations. Several iterative processing algorithms that employ statistical optimization methods [11] and convex set theoretic estimation methods [12] have been developed in the recent past. This type of processing is increasingly used for tactical imaging sensors (such as in passive millimeter-wave imagery and SAR imagery) due to the poor resolution in the captured images and the harsh operating conditions under which these sensors are deployed [3].

As noted in [13], a popularly employed method for tracking maneuvering targets using image data is to extract position information by computing the target centroid. When the acquired image is blurry, there may exist some uncertainty regarding the target's boundary, and consequently the computed centroid value may be erroneous. Since super-resolution enhances the angular resolution of a given image, the uncertainty of the measured bearing angle is decreased. It can be shown that the subsequent decrease in uncertainty of target location is more profound at further distances. Information value maps originally built for unaided sensors would change when the imagery was processed using super-resolution algorithms. The changes can be incorporated in two ways: either by redrawing the map (so that regions affected by range are extended outward), or scaling the information values at the map output. The latter is evidently simpler; however, one may note that the former would increase the maximum range by extending the lowest non-zero IV level into r = 0 territory. Fig. 8 shows a modified IV map that is adjusted for the inclusion of a super-resolution algorithm with arbitrary angular resolution enhancement in the original scenario depicted in Fig. 2. The changes to the information value map can be seen in Fig. 6. Redrawing the map to extend the range-limited values is shown in Fig. 8 by concentric dashed lines which represent the old transition boundaries. It may be seen that the new boundaries extend outward.

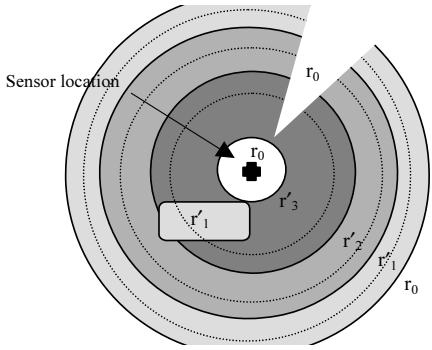



**Figure 8.** Map changes due to super-resolution.

Also, a region with the lowest information value $r_0$ is inserted around the sensor location to show the relative uselessness of the measured information where bearing rates of change exceed Nyquist limits of the diminished sampling rate due to additional processing.

The scaling method for changing the map can be seen as the re-labeling of the information values, i.e.

$$r_0 = 0.00\ ,\ r_1 < r_1',\quad r_2 < r_2',\quad r_3 \leq r_3'\ .$$

It is notable to point out that while super-resoultion improved some of the performance, it did not change the values due to the obstruction in the upper-right quadrant.

## 6. Conclusions

The principal contribution in this article is the introduction of a form of a priori sensor and scene-based information, called information value, as a useful concept in the design of fusion architectures for practical surveillance and tracking applications. This classification includes known effects on sensor performance and sources of confusion that exist on the scene and in the system before sensing operations are performed on the object of interest. A general definition for information value was given and examples of events affecting this value were discussed. A procedure for mapping the information value was introduced along with a simple example. Uses of the information value map in fused tracking architectures were discussed as a specific illustration. Finally, the effects of signal processing on the information value map were discussed where image super-resolution was used as the processing event.

Several useful investigations to extend the concepts and methods outlined in this article seem possible. The effects on the performance of the fusion architecture resulting from the employment of different transition boundaries discussed in Section 4.2 can be examined. Formal ways for modifying the information value maps as a consequence of introducing specific signal processing steps are to be developed. Also useful to investigate are the roles played by specific complexities arising in a multi-sensor environment, which may include the need for image registration and sensor synchronization, as well as the development of specific fusion architectures for multi-sensor target detection, classification and tracking applications.

## Acknowledgement

This research was supported in part by a grant from ONR (# N00014-97-1-0965) and by a grant from AFOSR (# F49620-97-1-0243).

## References

[1] B.V. Dasarathy, Decision Fusion, IEEE Computer Society Press, Los Alamitos, CA, 1994.
[2] J.J. Clark and A.L. Yuille, Data Fusion for Sensory Information Processing Systems, Kluwer Academic Pub., Norwell, MA, 1990.

[3] M.K. Sundareshan and S. Bhattacharjee, "Data Fusion for Imagery – Part 2: Super-resolution of Tactical Surveillance and Tracking Data for Fusion of Images," Proc. of NATO ASI on Data Fusion, Armenia, August 2003.
[4] P. Rohan, Surveillance Radar Performance Prediction, Peter Peregrinus Ltd., London, 1983.
[5] V. Nimier, "Supervised Multisensor Tracking Algorithm," Proc. CSREA Multisource-Multisensor Information Fusion, Las Vegas, Nevada, Vol. 1, July, 1998, pp. 149–156.
[6] N. Li and R. Li, "Target Preceivability: An Integrated Approach to Tracker Analysis and Design," Proc. CSREA Multisource-Multisensor Information Fusion, Las Vegas, Nevada, Vol. 1, July, 1998, pp. 174–181.
[7] M.K. Sundareshan and F. Amoozegar, "Neural Network Fusion Capabilities for Efficient Implementation of Tracking Algorithms," Optical Engineering, Vol. 36, March 1997, pp. 692–707.
[8] M.K. Sundareshan, "Fusion Architectures for Intelligent Integrated Processing of Multisensor Data in Target Surveillance and Tracking Applications," Proc. CSREA Multisource-Multisensor Information Fusion, Las Vegas, Nevada, Vol. 1, July, 1998, pp. 73–77.
[9] M.K. Sundareshan, H.Y. Pang, S.V. Amphy, and B. Sundstrom, "Image Restoration in Multi-Sensor Missile Seeker Environments for Design of Intelligent Integrated Processing Architectures," Proc. of 1997 SPIE Conf on Image Reconstruction and Restoration, San Diego, Aug 1997.
[10] C.Y. Chong et al., "Architectures and Algorithms for Track Association and Fusion," IEEE AES Systems Magazine, Vol. 15, Num. 1, January, 2000, pp. 5–13.
[11] M.K. Sundareshan, H.Y. Pang, and S.V. Amphay, "Optimized Maximum Likelihood Algorithms for Super-resolution of Passive Millimeter Wave Imagery," Proc. of 1998 SPIE Conf. on Passive Millimeter Wave Imaging Technology, Orlando, FL, April 1998.
[12] M.K. Sundareshan and S. Bhattacharjee, "Super-resolution of SAR images using Bayesian and Convex Set Theoretic Approaches," Proc. of 2000 SPIE Conf. on Algorithms for Synthetic Aperture Radar Imagery, Orlando, Fl, April 2000.
[13] M.K. Sundareshan and Y.C. Wong, "Data Fusion for Imagery – Part 1: Neural Network-based Fusion of Image and Non-image Data for Surveillance and Tracking Applications," Proc. of NATO ASI on Data Fusion, Armenia, August 2003.
[14] M.K. Sundareshan and S. Bhattacharjee, "Super-resolution of passive millimeter-wave images using a combined likelihood optimization and Projection-Onto-Convex-Sets approach," Proc. of SPIE Conf. on Passive Millimeter-wave Imaging Technology, Aerosense'01, Orlando, FL, April 2001.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Intelligent Fusion of Visual, Radio and Heterogeneous Embedded Sensor Information Within Cooperative and Distributed Smart Spaces

Carlo S. REGAZZONI, Reetu SINGH and Stefano PIVA
*Biophisycal and Electronic Engineering Department (D.I.B.E.), University of Genoa, Via Opera Pia 11/a, Genova 16145 Italy, carlo@dibe.unige.it*

**Abstract.** This paper discusses the main issues of Smart Space design with particular attention to different hierarchical fusion tasks and context information representation. General definitions are given to identify a common framework for exploring Smart Space problems. On these bases, a general architecture is defined along with its functionalities, and a practical example is outlined to demonstrate the implementation possibilities of the logical architecture.

**Keywords.** Smart spaces, data fusion, contextual information

## 1. Introduction

Human beings have a tendency to search for something innovative and new to improve their lives in various respects: freedom from work and undesirable activities as well as improvements for optimising their interaction with a variety of environments. These are examples of development lines that have long been the primary focus of a vast body of research. We have become highly dependent on various forms of technology wherever we are, in an office, in our cars, in the market place, at home, etc. The Information and Communication Technology era has not changed this trend; rather, it has augmented the potential that each space in which we exist can offer in terms of additional services. The emergent need for continuous communication and access to heterogeneous information has been a strong catalyst in this respect.

In particular, the development of portable personalized communication devices is a basic step in this direction, as it facilitates an anywhere/anytime connection to an unlimited electronic world of data. However, the mobile availability of a large amount of heterogeneous information is somehow contradictory to the throughput processing capability of a mobile user: a mobile user is typically already involved in tasks that require a large extent of his attention capabilities. Therefore, information used in interactions between a user and a space offering a service must be organized and selected according to a user-centred strategy aiming at satisfying his spatial-temporally variable goals with reference to the environment in which he acts.

The development of so called "smart spaces" elaborates on this concept by attempting to merge the most advanced results of research on sensors, context representa-

tion, decision and communication techniques within innovative information processing and communication architectures. One of the issues in designing an intelligent smart space system is how to develop fusion methods aimed at producing and processing contextual information coming from heterogeneous sensors. Possible *multi-modal sensing and perception techniques* and related advantages that a smart space can apply to optimise its interaction with potential users are analysed in this paper, with a focus on considering synergies among data provided by video cameras and sensors monitoring electromagnetic communications occurring in a space, as well as data obtained from virtual sensors describing the status of the communication network used by the smart space to interact with the user and to collect sensory data.

This paper is the first introductory part of two other papers, one which is explicitly devoted to multi-camera video data [1], while the other describes techniques based on spectrum analysis [2] for cooperative communication mode identification.

The paper is divided into two parts: in the first part, attention is paid to providing a general definition of the smart space design problem, an introduction to the concept of context representation, as well as the role of data fusion within such a framework, while the second part describes the architecture defined and used by our research group as a basis for developing new tools for smart space infrastructure design (namely, a smart "campus" scenario composed of outdoor and indoor facilities).

## 2. Smart Spaces

A *smart space* can be defined as an environment with sensing and action capabilities structured to provide augmented services to users [3]. This general definition can be applied to different domains. Each domain can be classified in terms of different characteristics: extension, sensing capabilities, range of action and communications, etc. Physically, Smart Spaces (SSs) can be considered as environments where sensors, computers, and information appliances collect and represent data, making decisions that trigger actions. SSs are designed to allow people to either perform new tasks or to perform old ones more efficiently. Heterogeneous sensors distributed in a smart space environment provide information that permits the SS to define a representation of the occurring situation, namely, its context.

*Contextual information* can be inferred from data collected by various sources such as video, radio, audio, virtual interfaces and heterogeneous sensors.

The creation of contextual information requires the SS to have the ability *to fuse* heterogeneous data to provide the system with a complete view of what is taking place in its premises.

The reason for which contextual information is important in a smart space is that the variability of services potentially offered by a SS has attained such a level of complexity that service *adaptability and personalization* have become increasingly important. Before revisiting this point, let us define some key concepts.

A *service* provided by a smart space can be defined as a procedural sequence of actions aiming at making a set of products available to one or more interacting entities.

We define *interacting entities* as objects that fall in either the sensing or the action/communication domain of a smart space.

A *product* can be defined as a *valued benefit* provided to the interacting user. The value of the benefit is correlated to the augmentation associated with the service. It is possible to distinguish between *virtual* and *physical products.*

1. *physical products* can be described as products that physically change the status of the environment by adapting it to user needs. For example, the status of a room is comprised of its temperature, the open-closed state of the doors, the location of users, their identity, shape, size and various other physical features;
2. *virtual products* are products that change the interacting user's perception of the environment status without necessarily changing any physical part of the ambience status. Examples of virtual products are video images representing a remote environment displayed on the PDA of a mobile surveillance user;
3. we can now discuss in further detail why contextual information is important when a SS delivers a service. The intuitive factor is that a product is generally not a self-contained fixed object; it can be parameterized with respect to different aspects, such as its content, the person to whom it is provided, the way by which it is delivered, etc.

Adaptability and personalization are therefore strictly dependent on the capabilities of a SS:

- to collect data from a set of heterogeneous sensors;
- to describe and represent sensor information within a contextual representation;
- to use contextual information to adapt and personalize products within specific (context-aware) services provided by the SS;
- to produce actions necessary to deliver physical and virtual products to interacting users;
- to learn/use specific action-communication models depending on the user;
- to provide the user with different levels of immersion depending on his position inside the SS;
- to augment the user's capacity to take action.

### *2.1. Service Metrics and Augmentation*

Physical and virtual services can potentially result in augmented traffic in the space in which they exist. The concept of augmentation is relative to normal environment usage, i.e. cases where no information technology is available to the smart space.

Another issue to consider is the definition of metrics to evaluate services that the smart space provides. Metrics can be applied to a specific service as well as to the global set of services. It is also reasonable to assume that the performance of a service depends on the performance of each elementary action defining that service. Therefore, each action can be evaluated as well.

If specific evaluation measurements are considered, they can be characterized by a large variability. However, taking into account the definition here proposed, it is possible to identify major characteristics to help define metrics, such as:

- time for service availability (temporal acceptability);
- number of people for whom the service is available (extension);
- number of products associated with the service (service complexity);
- number of services associated with a smart space (smart space complexity);
- quality of delivered products (product quality).

On this basis it is possible to better define the concept of augmentation of services as an improvement in one of the dimensions defined above.

## 3. Context Aware Information

A fundamental problem for designing SSs is the complexity of generating dynamically updated contextual information by collecting and fusing data acquired by multiple heterogeneous sensors.

Contextual information can be defined as *an ordered multilevel set of declarative information* concerning events occurring both within the sensing domain and within the communication and action domains of the SS. Relations among events are also included explicitly or implicitly within contextual information.

An *event* can be defined as the occurrence of some fact that can be perceived by, or communicated to, the SS; an event is characterized by attributes that basically answer questions about "where (position)" and "when (time)." Other attributes include "what (core)" the event consists of, "who (identity)" is involved in the event, and "why (reason)" the event has occurred.

Events can be used to represent any information that can characterize the situation of an interacting user as well as a part of a SS, i.e. an entity.

An *entity* can be a person, a place, or an object that is relevant to the interaction between a user and an application. The user and the SS components themselves are entities. The multilevel nature of contextual information is related to the possibility of detecting and representing events at multiple abstraction levels.

The SS reacts, adapts and presents itself to the user according to the available contextual information. Therefore, in a context aware SS, contextual information can either be used as a product, i.e. it can have its own value for the interacting user, or it can be used to optimize the adaptation and personalization process.

Within the range of services provided by a smart space, two main types of contextual information can be distinguished: fixed information and dynamic information.

1. fixed information can be viewed as stationary knowledge, with reference to the time in which the smart space provides its services. Therefore, we could say that the quantum of fixed information can be described by an event that remains constant during the time in which the service occurs. Fixed information can be in itself a product of a service, and it can also be combined together with additional knowledge in the form of a product. For example, information about the map of a campus can be considered *fixed*: a car that remains parked within the campus area during the delivery of the service is a fixed event. In much the same way, methods for presenting this knowledge on different types of a priori defined terminals can be represented by fixed procedural knowledge (e.g. a DLL library);
2. dynamic information represents declarative knowledge that can vary with a dynamic behavior comparable with or faster than the time during which the smart space provides its services. As a service is usually activated in a burst way, this definition implies that dynamic information can vary during the delivery of a product to the interacting user. In this case, the quantum of information is an event that can vary during the service. For example, if a student would like to access a study room and he wishes to know whether the room is

full or not, then he could ask for information about the current number occupants to decide whether or not to make use of that room. Accordingly, the change in position of a student may cause a change of transmission modality due to the optimization of the quality of service provided in the served areas. Therefore, declarative dynamic information can be used either as an information value by itself or as meta knowledge useful to index declarative or procedural databases.

We can now introduce a further classification of knowledge.

Knowledge available *prior to the start of a service session* is referred to as ***a priori knowledge***. Knowledge that is learned *during system runtime* is referred to as ***on-line instantiated knowledge***. For example, a label attached to an object can be considered as instantiated knowledge if it is estimated from data originating within the sensing subsystem, while knowledge attached to the label, which can be inherited by the object, is referred to as *a priori* known. The nature of an interacting object (e.g. vehicle or a person) is an example of instantiated information that can be attached to an object. The profile of a given user recognized by the smart space can be a priori known by the system, but becomes active only when it is triggered by user recognition. *A priori* knowledge can be *learned* on the basis of on-line instantiated knowledge or off-line recorded versions of such knowledge.

It may be useful to discuss what information should be produced (by sensing and reasoning) for augmenting different classes of services. Let us first examine instantiation of dynamic knowledge by means of sensing and perception-oriented reasoning processes. In smart space applications, sensing and reasoning are geared toward obtaining an updated description of the environment itself and the users that interact with the environment. Real-time and robustness constraints are the key requirements of the design of such processes. This dynamic knowledge can be useful both to initialize smart space services and to maintain an up-to-date context perception during the time of service.

Let us now examine the categorization of dynamic context information. Knowledge that can vary at a faster rate than the change rate of a smart space service can be correlated to specific object properties as well as to relational properties among objects, i.e. events and their relations.

The two types of objects can be differentiated as follows:

- the *status of objects that are components* of the environment can be directly observed and controlled by the environment itself;
- *the status of objects that interact* with the environment *cannot be directly controlled* by the smart space itself, but *can be estimated by appropriate sensing actions*.

Let us refer to the former as *environmental objects* and the latter as *interacting objects*. From now on we will examine interacting objects, as they often represent the most important part of dynamic information. For example, people and vehicles present in a parking area, or students in a campus, are instances of interacting objects.

Information about interacting objects can be categorized as dynamic information associated with single interacting objects, relationships among interacting objects, and relationships between interacting objects and the smart space itself – including its components, i.e., environmental objects. Dynamic context information about objects can be expressed in terms of absolute object properties, such as location, class information, behavior, etc. These absolute properties can also be defined as either *static or temporal*.

*Static* properties designate information that holds for a single observed time instant, whereas *dynamic* properties hold for ensembles of multiple time instants. Therefore, different dynamic properties can be computed on the basis of the same observable data but on different time sets.

The simplest case of dynamic property refers to a pair of time instants, designated as *second order* properties to distinguish them from *first order properties* which refer to single time instants.

Note that fixed information can be associated with interacting objects: it can be learned or it can be a priori known. Consider a user profile including preferred trajectories for reaching a given objective. This knowledge can be used by the smart space to interact with the user e.g. by optimizing interface modalities for keeping in touch with him once he has been recognized.

### *3.1. Reference Systems*

Let us consider the *first order properties* of dynamic context information. These properties can be identified in static information about a single object at a given time, e.g. location and identity. In general, before we can define object location at a given time we must first define an appropriate location reference (LR) system.

The reference sensing information (RSI) set can be defined as the set of reference systems needed to describe different information perceived by a smart space. This RSI can be used both to define metrics associated with a quantitative evaluation of first order information, as well as to define limits in the sensing areas of a smart space. The RSI set is important in defining smart spaces because it makes it possible to provide virtual products and to define particular subsets of instantiated dynamic knowledge.

For example, in a given smart space, one can decide that object identity is an important information for distinguishing among humans, vehicle, etc. One can decide as well, that the presence of a dog or a cat should not be considered by the LR system for that smart space. Likewise, the sensing area associated with a LR system can be divided into sub-areas where the presence of an interacting object generates specific responses by the system.

A further observation can be made regarding the *resolution level* of a first order property. Given a reference system, available sensors may perceive a given information only up to a given level of precision due to quantization and physical limits. Therefore, the scale used for a given reference system can be defined only with reference to the particular resolution at which information is made available by sensors. The first order information is assumed to be a fractal: *i.e.* zooming down along a given information with more powerful "sensing" lenses makes it possible to discover new information subspaces, within which new first order information can be collected and represented.

For example, the detection of a specific part of the human body (e.g. a face) can be activated with the appropriate sensor setup so that whenever a human is detected, sensors seek out his face and a facial recognition algorithm generates information about the interacting object on a finer scale than if sensors had been scanning for information about his identity as a human. Subparts can thus be tagged with identity information and services related to the specific level of abstraction can be triggered.

## 4. Data Fusion

Data fusion has been defined as the seamless integration of data from disparate sources. The typical application of data fusion techniques is the synergetic use of sensory data retrieved from multiple sensors.

These techniques are mainly applied in systems which consist of multi-sensors, such as detection, tracking, and identification in a video surveillance system [4].

In order to estimate the quality of data from multiple sensors, it is important to understand the distinctive features of the different sensors. The application of multiple sensors to instance-authentication, location estimation, surveillance, *etc.* offers several possible performance benefits over traditional single-sensor based approaches. These performance benefits must be evaluated taking into account additional cost, complexity, and interface requirements for any given application.

The amount of benefit that can be drawn from fusion depends on the type of sensors, the methodology used for fusion and the kind of environment the system is operating in.

There are many data fusion issues related to smart space design both at the physical sensing level and the virtual network level. Some of these issues will be discussed below for a better understanding of data fusion benefits. The problems considered in our discussion are merely a subset of the total number of possible problems; in particular, we address the issues of video and radio data exploitation for context awareness in smart space applications. Without loss of generality, most of the following problems are common to all sensors used in the field of smart spaces.

The discussion on fusion must take into account that data fusion can be implemented at completely different levels which can be classified into five groups, according to the JDL (Joint Director of Laboratories) model [3,5–8]. Within the scope of this text, we focus on fusion level 0 and 2, namely signal and situation assessment levels (figure 4).

## 5. Description of the Architecture

### *5.1. Modeling a Smart Space: A Logic-Functional Architecture*

The architecture here proposed for a smart space is a user-centred closed loop composed of four main functionalities: sensing, analysis, decision and action. The loop is presented in figure 1. Once a semantic representation of events of interest is evaluated based on inference, the system takes a decision and performs the associated action.

There are two types of actions: *informative messages* directed toward the user, who becomes the subject of a multimode communication, and *physical actions* that the system performs on itself (the controlled environment) through various kinds of actuators.

Future development should consider the third level of the proposed JDL fusion model, *impact assessment*, to exploit the system's acquired knowledge about the user and its analyses of the user's reactions in order to evaluate its decision and the undertaken communication action.

In this global context, the system can differentiate between two types of users: *registered* users, namely users equipped with multimode communication devices already registered by the Smart Space, and *guest* users which are not equipped, or own unknown devices.

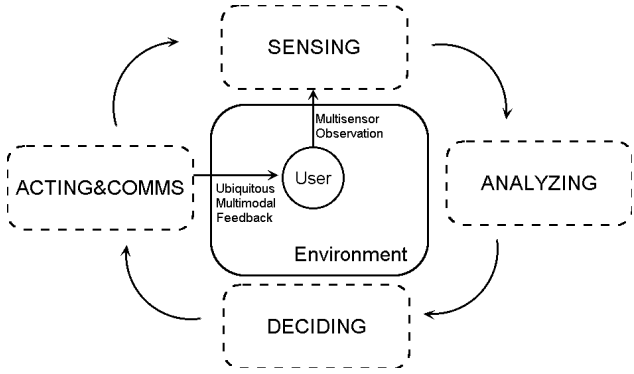



**Figure 1.** A user-centred closed loop is taken as a model for SS architecture implementation.

To further examine the proposed system, we refer to the diagram outlined in figure 2. The progressive levels of abstraction are classified as follows: from the *physical service space* level (the environment), to intermediate levels of the architecture, the *sensing*, *communications* and *human-computer interface (HCI)* logical blocks and the *context services* layer, and finally, the *applications* level. There are six main blocks divided into four levels, each related to a given data abstraction level.

Starting from the lowest level of abstraction, the first block of the model represents the *physical service space*, which can further be subdivided into *internal* and *external world.* The former denotes the system itself and all parts of the environment that are directly controlled by the system; the latter is everything acting inside the observation domain, namely, targets of the smart space. This distinction is discounted in the diagram because the description is mainly centred on the data fusion devoted to producing system context awareness.

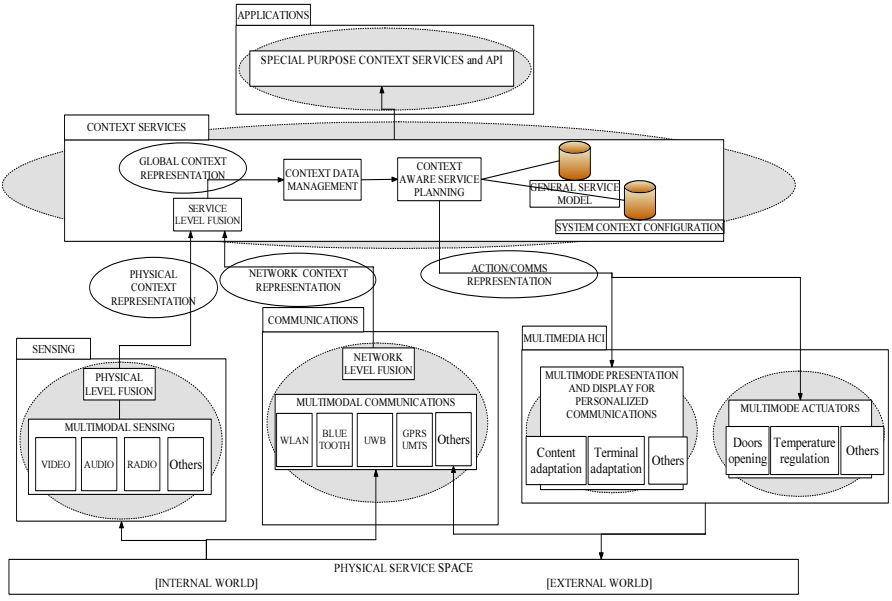



**Figure 2.** Smart Space logical-functional architecture.

The salient physical properties of the environment are sensed by the smart space through its receptors (in the diagram this is represented by the data flow arrow towards the sensing block), while the other connections are representative of the direct link between the environment and the communication devices and the inference system on the physical service space itself. The receptors provide fundamental input to the fusion system, and as such, they represent a wide array of different physical or logical devices, varying with the application to support the design of the particular smart space.

In general, sensors can be logically divided into two main groups: *exogenous* and *endogenous* sensors. In our view, sensors do not differ topologically; their type depends on the monitored object. Thus by *Endo-receptors* we define all sensors acting within the system itself, namely devices fit for analysing internal components or variables relevant to devices that constitute parts of the whole organism. Examples of these kinds of sensors are: network load observers, devices on/off status sensors, PC access sensors, PC processor computational load sensors, thermal sensors, diagnostic routines, lighting sensors, safety-oriented sensors (smoke, gas, fire, water infiltration, etc.).

*Eso-receptors* represent all devices used by the structure to keep track of the events occurring in the observed domain and to collect data about targets of analysis, either humans or other external varying objects. Eso-receptors may be controlled by feedback signals from the regulation system in order to adapt to the changes taking place in the observed environment. Some examples are: video sensors (working in visible or infrared fields), Global Positioning Systems (GPS) [9], RAdio Detection And Ranging (RADAR) systems [10], standard or directional microphones, weight sensors, magnetic badge readers, fingerprint readers, electro-magnetic waves emission scanners, photoelectric cells.

Within the sensing block, the next step, after multimode sensing acquisition, is that of first level data fusion.

At the core of the system's processing ability are the data fusion and decision centres that process observations coming from receptors sensing the internal and external domains and on stored data (memory manager).

*Data fusion* within the system is essentially performed in three separate parts of the architecture: the system sensing task corresponds to multi-sensor physical level fusion; the result of this low level fusion task is *physical context representation,* namely information on objects sensed by the smart space through multimode sensors [11–13]. Parallel to the networking area, we have a network level fusion producing what we call the *network context representation*. The two representations are then fused together by the service level fusion to obtain a *global context representation.*

It should be noted that, in the case of sensing, there is a direct correspondence between the high level closed loop smart space diagram and the logical architecture.

Fusion of the video and radio sensors will be discussed further on.

The *Communications* logical block combines all message delivery functionalities of the smart space. Communication modality has evolved from an "anytime anywhere paradigm" to a "context-aware approach" in which sensing capabilities influence the communication modality.

The adjective "multimode" means that information must be as scalable as possible and maximum compatibility with different receiving devices must be assured in order to fulfil the widest immersive environment paradigm. Multimodality, in short, refers to *content* (voice, video, text, etc.), *terminal device* (PDA, fixed terminal, etc.) and *transmission* (WLAN, GSM, direct audio synthesis, etc.).

In this general description, the bridge between the smart space and what we called the physical service space (the controlled domain and all external objects interacting with it) is represented by the multimedia Human Computer Interface (HCI). This block is in turn divided into a user interaction oriented part, namely *multimode presentation and display for personalized communications*, and a more physical inference of the system on the controlled domain and on itself: *multimode actuators*. The actuator manager makes it possible for the system to act on several of its physical components: decisions taken by the system can trigger mechanical contraptions such as door or window openers, lighting activators as well as a change in the software parameters of internal parts of the smart space. *HCI* is becoming a focal point in the research on smart space development due to a continuous increase in system functionalities as well as the need to conceal these new complex features from the end user and let him believe that he is acting in a natural environment. The focus of this research is in the implementation of the pervasiveness paradigm so that the powerful possibilities offered to the user do not require any intervention on his part to exploit the augmented services he is provided with. This significant feature of the smart space calls for the development of something straightforward the user can accept without question and that has an imperceptible learning curve. Therefore, considering the increasing complexity of present day technology, the only way to make research progress in the right direction is to shift the point of view from a "system centred" to a "user centred" design. To this end, it is essential that a smart space provide connections to a diversity of devices to allow for the widest possible group of users and to facilitate the involvement of human actors in the environment without recourse to artificial means. This can be achieved through an interpretation of user gestures and movements or simply by monitoring the relative positions of users and the system's sensing parts.

Consider one of the most advanced forms of HCI technology, the use of animated characters – or faces – called *Avatars* [14]. At D.I.S.T. laboratory of the University of Genoa, the use of synthetic 3D human faces to naturally communicate with the user is under extensive study. These synthetic interfaces are outfitted with a high number of facial animation parameters (FAP) to reproduce the expressiveness of human conversation, e.g. lips that move in close synchronization with the actual words pronounced by the speech engine. The final result of such work is the possibility of achieving an interactive synthetic interface that can adapt the complexity of its animation to the power of the device owned by the user (high degree of scalability) [15].

In comparison of the closed loop general diagram and the system's logical architecture, we can say that the previously described blocks (Communications and HCI) are designed in accordance with "Acting and Communications" requirements.

At a higher level of abstraction, the *physical context representation* and the *network context representation* provide input to *service level fusion*, which is the final step of fusion and produces a *global context representation*. The function of the *context services* block is to implement the *analysing* and *deciding* functionalities of the smart space within the general scheme. Once a high level context representation has been obtained, the above knowledge is ready to be used to make the system react. In this case, *context data management* is the basis of *service planning. Context data management* thereafter closes the described logical loop in the physical service space with the use of the Communications and Interface levels.

The description of the model/system/scheme is completed by closing the diagram with the application level. At the topmost level of the hierarchy, processed information

can be used by *special purpose context services* with the intent of adding high level functionalities in the form of applications and API.

### *5.2. Mapping the Concepts into the Physical Architecture*

For the purpose of practical demonstration, the conceptual model has been implemented [16] and briefly described in figure 3. It represents the mapping between the user-centred closed loop (figure 1) and the abstract SS model; different computational units (PC 1-2-3-4 and PDAs) have been used to support SS functionalities.

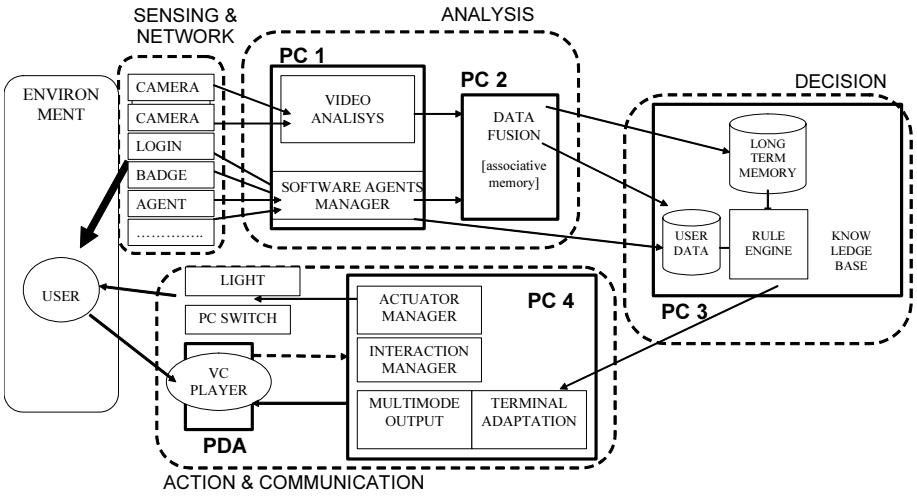



**Figure 3.** Functional/physical architecture of the Smart Space infrastructure.

A network infrastructure (combination of LAN and 802.11 WLAN) connects the different physical units and makes real-time data exchange possible. In particular, tasks are allocated in 5 different computational units:

- **PC 1**: Image Processing and Resource Management algorithms implementing the Sensing capabilities of the SS;
- **PC 2**: Information Conversion and Data Fusion methods finalize SS Sensing;
- **PC 3**: rule-based engine & databases;
- **PC 4**: communication server that manages multimodal output generation and interaction with mobile terminals;
- **PDA**: Personal Digital Assistant that maintains a communication link with the SS infrastructure and displays audio/visual feedback to the user.

The need for separate computational units is motivated by the fact that different integrated sensor systems grant scalability in the number of sensors, actuators and user devices.

### *5.3. Contextual Information*

Contextual information can be divided into two parts: fixed and dynamic [17–20]. Fixed contextual information is information which does not change with time, for example:

- *monitored area map:* the basis of the *a priori* knowledge for keeping track of the movements of interacting objects;
- *map of restricted areas*: restricted areas in the controlled area map;
- *network architecture*: network topology and device parameters;
- *network capacity:* system bandwidth limitations and user rights;
- *list of registered users*: list of users for the login procedure;
- *number of personal computers:* to discriminate the maximum number of users.

Dynamic contextual information is used to enumerate several real-time observed variables and perform continuous adaptation by on-line mapping of contextual information.

Examples of dynamic contextual information are listed below:

- *user location:* ascertained through the use of cooperative cameras with partially overlapping fields of view and collaborative radio sensors;
- *instantaneous computational load of laboratory personal computer processor*: dedicated software agents that control processor activity to discriminate resource usage levels;
- *dynamic load of local area network:* network software agents designed to keep track of the data transmitted on the net;
- *power state of personal computers*: binary ON/OFF value;
- *the distribution of people in internal area:* activity is monitored through internal smart cameras to manage system resource allocation;
- *user login:* by monitoring logins, the system controls authorizations and current users.

The above described list of contextual information refers to what we have defined as *second order information*, e.g. information characterized by its instantaneous variation. The system focuses on the modification of such parameters, to react and adapt to new situations in accordance with the concept of context awareness.

Context knowledge is the backbone of the smart space, as it really makes the system aware about the environment it operates in, thus substantiating the use of the terminology *Smart* Space. There is virtually no limit to the complexity of sensor topology; due to the use of distributed computation and data abstraction the elaboration load can be significantly reduced. The core problem is how to manipulate information in order to find useful decision rules and smart techniques to make correct use of collected data.

### *5.3.1. Reference Systems*

In the general description of the SS provided in the first part of this paper, the majority of the information collected by the multi-sensor systems is characterized by their own reference systems. The collection of these different reference systems contributes to a Reference Sensing Information (RSI) set. The map of the environment where multi-sensors co-exist is a good example of a common reference system (Location Reference system – LR). To better understand the relation between different events occurring in a diversity of reference systems, these reference systems can be translated into a single reference system. This will affect the definition of the network architecture. To conclude, each reference system can be respectively translated into a common reference system to achieve a better understanding of context related information.

### *5.4. Contextual Information Representation Levels*

In the architecture proposed above, contextual information is divided into different hierarchical abstraction levels (summarized in figure 4).

*Multi-sensor level context representation*: this is the zero level of contextual information. The context is represented by the fusion of data from homogeneous sensors, such as multi-camera setups, multiple base stations radio architectures, etc. This level of representation is obtained through signal assessment techniques [21–30].

*Physical level context representation:* this is the first level of contextual information. Multi-sensor level contextual information is combined with measurements from single physical sensors to build a unique physical level representation. This information abstraction is equivalent to object assessment.

*Network level context representation:* at the same level as that of physical abstraction, we consider fusion of all the data derived from the sensing devices in the scope of communication network management: network load measurements (obtained through virtual sensors managed by appropriate software agents), the computational load of personal computers (real-time data managed again by software agents), and user login information. These measurements together with *a priori* system knowledge can be used to produce a general network description.

*Service level context representation:* this higher fusion level is what really defines smart space context awareness. The data composed of the information collected at lower levels reaches its maximum level of abstraction and the global context becomes one tag among a proper number of known situations. There are seven *super-states* considered in our example. For each context representation (physical, network and global context representations) appropriate data fusion methods must be defined. In the proposed smart space application, data fusion is performed at the levels described by the outlined logical architecture in figure 4:

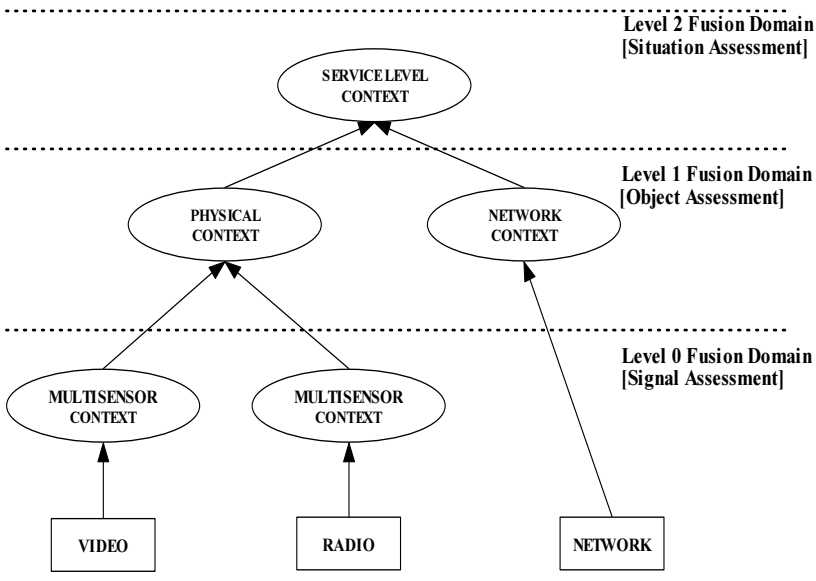



**Figure 4.** Hierarchical contextual information levels representation.

*Multi-sensor data fusion:* at this level the first problem considered is how to provide a single representation for the objects perceived by multiple identical sensors [4]. Examples of signal assessment are the following:

- multi-camera video-based fusion: the process of integrating observations from multiple video cameras into video events; the goal is to augment human visual observation capability. We consider the output of two cameras observing the laboratory to obtain information about the number of people crowding the room. The positions of the field of views of the two video sensors are chosen to improve area coverage so as to span the entire room. The area characterized by the intersection of the two fields of view is exploited by fusion rules to produce augmented information and simultaneously solve the problem of occlusion by utilizing information coming from two different data sources. Further cooperative multi-camera issues are discussed in [1];
- multimodal radio fusion: the process of detecting and locating radio terminals/base stations within a smart space with the aim of augmenting SS/user communication capability. The multiple radio base stations are used to calculate the position of mobile transmitting devices and identify the transmission mode of the user by estimating its power and using a-priori knowledge. The result obtained in the *radio events* are used for real-time transmission adaptation.

A detailed technical discussion of the problems of radio data fusion in cooperative homogeneous and heterogeneous multi-sensor configuration sensorial networks can be found in [2,31–33].

*Physical level fusion*: this is the second hierarchical level of fusion comprised of the fusion of video and radio events along with the data from other physical sensing devices into a single coherent representation. The information used in this object assessment level is feature centered. An example of video and radio data fusion is presented in figure 5. The video objects (blobs) are composed of such features as position, shape, colour etc., whereas radio objects are represented by their transmission modality, position, transmitting power, etc. The specific goal is make it possible for the SS system to establish appropriate communications/actions with the user. The obtained data and observations coming from all sensors contribute to building the global *physical context representation*.

*Network level fusion:* the process of obtaining an integrated representation of the status of the communication network and related resources. The goal is to provide the SS system with an awareness of its global communication capabilities by integrating information present in the local net. The network context representation comprises all the information needed for interaction and communication between objects (interacting or part of the smart space) and the system. In our system this information is formed by the previously quoted network load measurements, computational load of personal computers and user login information. The system applies this data to perceive its interaction with the users acting on its components, e.g. laboratory devices. All of this information is sent to the higher level fusion task which is devoted to the extraction of a complex context representation.

*Service level fusion*: the process of establishing a coherent representation framework that integrates different lower level context representations. The aim of the fusion task is to make available a structured context representation (global context representation) for adapting and personalizing delivery. In the proposed smart space, highest level

fusion is represented by the use of an associative memory to build the contextual *super-state.* This *state* is employed by the system in making a decision about the situation of the observed domain and thus closing the loop by reacting accordingly.

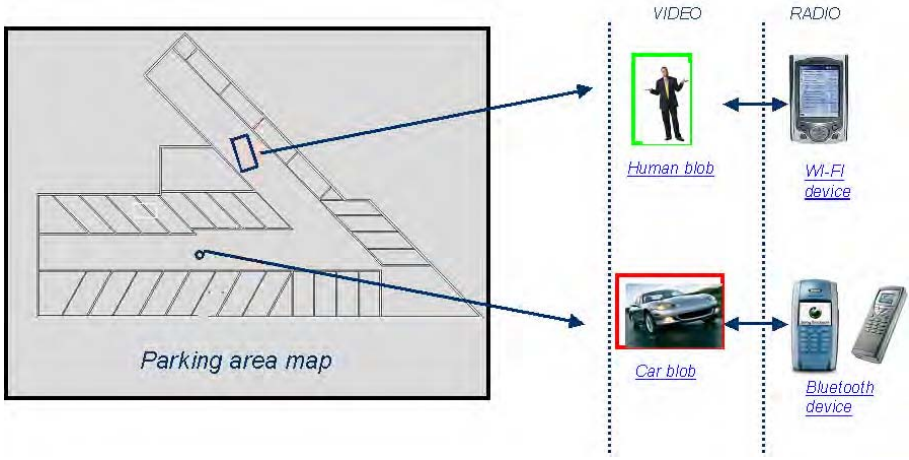



**Figure 5.** Example of physical level fusion: two interacting objects (a man and a car) in a parking area are recognized through different sensing devices and results are associated.

The associative memory successively collects pre-processed data from various sensing interfaces. This information is exploited to obtain *a priori* knowledge about a certain number of known events. This is defined as service level fusion, which is used to obtain a global context representation (summarized by the super-states). The controlled situations defined by the super-states concern lab occupation and resource usage, as in above stated example, wherein the system controls the number of PCs that have been switched on, user login information and the usage of network and computational resources. A number of internal cameras are also used to monitor the number of people present in the laboratory room at a given time. The result of the global assessment is summarized in one of the tags reported in table 1. The associative memory is implemented in practice through a Self Organizing Map (SOM) [34,35], a neural network technique, while the internal data management is performed by a society of software agents.

**Table 1.** Super-states and actual meanings

| super-state | real world |
|---|---|
| WHF | low human work |
| WHL | high human work |
| WAF | low machine work |
| WAL | high machine work |
| ARRIVE | laboratory incomes |
| intrusion | unauthorized presence |
| empty | everything stopped |

An example of the SOM output where decision clusters have been highlighted is shown in figure 6 [16].

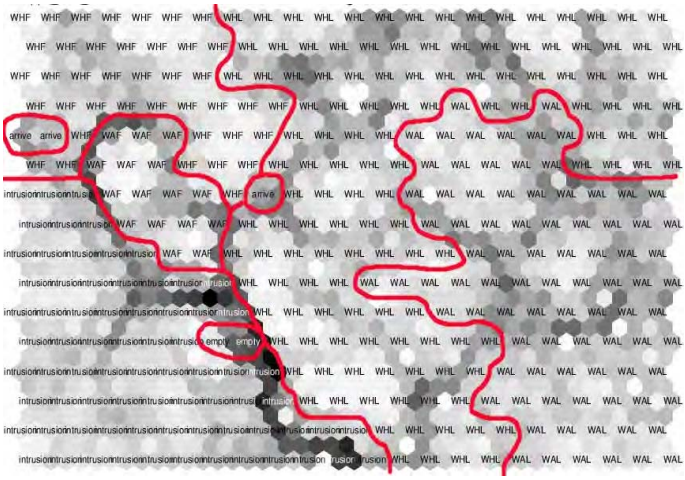


**Figure 6.** Example of SOM output.

To extract maximum information, more complex strategies for exploiting contextual information have to be defined, e.g. by increasing the dimension and the complexity of the state vector.

**Table 2.** Data fusion steps and practical system issues

| | FUSION LEVEL | | | |
|---|---|---|---|---|
| | **MULTISENSOR** | **PHYSICAL** | **NETWORK** | **SERVICE** |
| **DETECTION** | *video-cameras, radio access points* | *multisensor context representation* | *network context representation* | *physical representation + network representation* |
| **ASSOCIATION** | *video: multicamera* | *physical sensors' data* | *network devices' variables* | *associative memory* |
| | *radio: multiple base stations* | | | |
| **STATE ESTIMATION** | *position* | *global physical state* | *network internal state* | *events* |
| **ATTRIBUTE** | *interacting objects' coordinates* | *video: color, shape* | *CPUs usage, logins, network* | *lower level physical* |
| **CLASSIFICATION** | | *radio: trasmission mode* | *load* | |
| **ASSESSMENT** | *interacting objects' movement* | *temporal analysis* | *network resources employment* | *superstates* |

## 6. Conclusions

The key issues of a smart space system have been described. In particular, a general logical structure has been considered for implementing a system able to acquire information from different intelligent sensors and integrate it using data fusion at different levels of abstraction. The highest fusion abstraction level, representing the system intelligence, is implemented through an associative memory module.

In the description of practical issues particular attention has been given to video and radio sensors but the general framework can be applied to all the common problems arising from the design of a complex contextual-based intelligent system. The definitions of the various problems are intended to give a general, theoretical approach to smart space design issues.

## Acknowledgments

This work was partially supported by the VICom (Virtual Immersive Communication) research project, financed by the Italian Base Research Funds (FIRB), and by the University and Scientific Research Ministry (MIUR) of the Italian Government under the National Interest Scientific Research Program.

## References

[1] L. Marchesotti, G. Scotti and C. Regazzoni, Issues in Multicamera Dynamic Metadata Information Extraction and Interpretation for Ambient Intelligence, NATO ASI in Data Fusion, Armenia, 2003.
[2] M. Guainazzo, M. Gandetto, M. Musso, C. Regazzoni Multimodal Cooperative Modulation Estimation and Terminal Localization for Multisource Wireless Sensor Networks, NATO ASI in Data Fusion, Armenia, 2003.
[3] Alan N. Steinberg, C.L. Bowman, and F.E. White, Revisions to the JDL Data Fusion Model, NATO IRIS Conference Proceedings, Quebec Canada, Oct 1998.
[4] A.K.Hyder, E. Shahbazian and E. Waltz, Multisensor Fusion, Proceedings of the NATO Advanced Research Workshop on Multisensor Data Fusion, Pitlochry, Perthshire, Scotland, June 25–July 7, 2000.
[5] Alan N. Steinberg, C.L. Bowman, and F.E. White, Revisions to the JDL Data Fusion Model, NATO IRIS Conference Proceedings, Quebec Canada, Oct 1998.
[6] Kam, M., Zhu, Q., Gray, W.S., Dept. of Electr. & Comput. Eng., Drexel Univ., Philadelphia, PA, USA "Optimal Data Fusion of Correlated Local Decisions in Multiple Sensor Detection Systems," Aerospace and Electronic Systems, IEEE Transactions page(s): 916–920, July 1992 Volume: 28 Issue: 3.
[7] Aarry L. Van Trees, Detection, Estimation and Modulation Theory, vol. 1, John Wiley and Sons, 1968.
[8] Robert R. Tenney and J.R. Nils, R. Sandell, "Detection with distributed sensors," IEEE Transaction on Aerospace and Electronic Systems, vol. AES-17, pp. 501–510, July 1981.
[9] P. Enge, and P. Misra, "Special issue on GPS: The Global Positioning System," Proc. of the IEEE, pp. 3–172, Jan. 1999.
[10] Paramvir Bahl and Venkata N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," IEEE Infocom 2000, volume 2, pages 775–784, March 2000.
[11] E. Waltz and J. Llinas, Multisensor Data Fusion, Artech House Inc., 1990.
[12] Weaver, C.B., "Sensor Fusion," Proc. SPIE, Vol. 931, Orlando, FL, April 1988.
[13] Atwater, F.M., "An Expert system approach to Situation Assessment Using Sensor Fusion," Proc. AFCEA Seminar on AI Applications to the Battlefield, Ft. Monmouth, NJ, May 1985.
[14] F. Lavagetto, R. Pockaj, The Face Animation Engine: Towards a High-level Interface for the Design of MPEG-4 Compliant Animated Faces, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 9, N. 2, March 1999.
[15] F. Lavagetto, VIDAS Analysis/Synthesis Tools for Natural-to-Synthetic Face Representation, ECMAST-99, Madrid, May 25–27, 1999, pp. 348–362.
[16] M. Gandetto, L. Marchesotti, S. Sciutto, D. Neuroni, C. Ragazzoni, From Multi-sensor Surveillance Towards Smart Interactive Spaces, ICME 2003, Baltimora, USA, 6–8 July 2003.
[17] Paul Castro and Richard Muntz, "Managing Context for Smart Spaces," IEEE Personal Communications, October 2000.
[18] O'Sullivan, A Smart Space Management Framework – Wade (2002).
[19] http://www.nist.gov/smartspace/.
[20] http://www.cc.gatech.edu/fce/. Homepage of Future Computing Environments (FCE) Group at Georgia Institute of Technology. Includes information about their projects and publications.
[21] I. Haritaoglu, D. Hardwood and L.S. Davis, W4S: a real-time system or detecting and tracking people in 21/2d, in European Conference On Computer Vision, pp. 877–892, 1998.
[22] http://www.cs.ukc.ac.uk/people/staff/nsr/mobicomp/. University of Kent at Canterbury's (UKC) mobile & context-aware computing web pages.
[23] http://www.it.kth.se/edu/Ph.D/LocationAware/aware.vt98.html. Link collection to many papers about context-awareness.
[24] http://mcs.open.ac.uk/drm48/chi2000.html, CHI2000 Workshop 11: The What, Who, Where, When, Why and How of Context-Awareness, Monday April 3rd 2000, The Hague, The Netherlands.
[25] http://www.teco.edu/chi2000ws/, CHI2000 Workshop: Situated Interaction in Ubiquitous Computing, Monday April 3rd 2000, The Hague, The Netherlands.

[26] Dey, A.K. and Abowd, G.D. (1999). Toward a better understanding of context and context-awareness. GVU Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology. ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf.
[27] Dey, A.K., Salber, D., Abowd, G.D., Futakawa, M. (1999). The Conference Assistant: Combining context-awareness with wearable computing. 3rd International Symposium on Wearable Computers, San Francisco, California, 18–19 October, 1999, pp. 21–28.
[28] Long, S., et al. (1996). Rapid Prototyping of Mobile Context-aware Applications: The Cyberguide Case Study. 2nd ACM International Conference on Mobile Computing and Networking (MobiCom'96) 1996 November 10–12, 1996.
[29] Pascoe, J., Ryan, N.S. and Morse, D.R. (1999). Issues in Developing Context-Aware Computing. Proceedings of the International Symposium on Handheld and Ubiquitous Computing (Karlsruhe, Germany, Sept. 1999), Springer-Verlag, pp. 208–221.
[30] Bill N. Schilit, Norman I. Adams, and Roy Want, Context-Aware Computing Applications, in Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994. IEEE Computer Society.
[31] Paul Castro, Patrick Chiu, Ted Kremenek, and Richard Muntz, "A Probabilistic Location Service for Wireless Network Environments," In Proceedings of Ubicomp 2001, pp. 18–24. Springer-Verlag, September 2001.
[32] P. Prasithsangaree, P. Krishnamurthy, and P.K. Chrysanthis, "On Indoor Position Location With Wireless LANs," The 13th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2002), Lisbon, Portugal, September 2002.
[33] http://www.shosteck.com/news/location.html.
[34] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, Self-Organizing Map Program Package, University of Technology – Laboratory of Computer and Information Science, Helsinki, Finland, April 1995.
[35] Kohonen, T., Automatic formation of topological maps of patterns in a self-organizing system. In Oja, E. and Simula, O., editors, Proceedings of 2SCIA, Scand. Conference on Image Analysis, pages 214–220, Helsinki, Finland. Suomen Hahmontunnistustutkimuksen Seura r.y., 1981.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Multimodal Cooperative Modulation Estimation and Terminal Location for Multisource Sensor Networks

Marco GUAINAZZO[a], Matteo GANDETTO[a], Maristella MUSSO[a],
Carlo S. REGAZZONI[a] and Pramod K. VARSHNEY[b]
[a] *Department of Biophysical and Electronic Engineering (DIBE)*
*University of Genoa, Italy*
[b] *University of Syracuse*

**Abstract.** Modulation/mode identification is an interesting topic in wireless communications. The main problem related to this topic is the recognition of the transmission mode employed in a radio environment. In this work, this problem is addressed in the context of Smart space (SS) applications for a distributed wireless sensor network, in which the sensors are the user radio reconfigurable terminals (UTs as mobile nodes) and the base stations (BSs as fixed nodes). A stand-alone and a cooperative approach to modulation detection by using multiple sensors exposed to different signal levels dependent on their positions are presented. In the case of cooperative identification, the proposed methodology is based on the distributed detection theory. Another problem considered in the paper is the problem of identification of terminal location. While special attention is given to the recognition problem, a preliminary study of the identification of terminal location with both stand-alone and cooperative UTs/BSs is also presented. Theoretical aspects and some numerical results are reported to support the proposed methodologies.

**Keywords.** Modulation identification, location estimation, distributed detection, cooperative method, smart space, data fusion

## 1. Introduction

A Smart space (SS) can be defined as an environment in which sensing and operating capabilities are structured to provide augmented services to users. Smart spaces are work environments with embedded computers, information appliances and multi-modal sensors allowing users to perform tasks efficiently. In the smart space context, physical objects defined as interacting entities are considered. These physical entities are exploited by the space to improve its level of knowledge. The smart space also contains a communication link to present information to the user. We consider applications utilizing radio, video, and audio sensors and identify particular objects in the SS by using the received electro-magnetic (em) signals. In our case, interactive entities present in the SS are the UTs (transmitters/receivers) and the BSs (base stations).

There are two kinds of problems to be considered:

- object location estimation;
- object-attribute identification.

Information on the locations of the radio objects (UTs and BSs) at evens time is important for providing the SS with a good level of knowledge of the radio entities and for enhancing context awareness.

Object-attribute identification requires one to define the properties of the radio objects (BSs and UTs). In particular, for the BSs it is important to know which kind of transmission standard has been adopted to communicate with the user (for example, wireless LAN, Bluetooth, etc.). For the terminals (UTs), it is useful to know the type of modulation employed for transmission within the SS and, therefore, techniques to perform automatic modulation/mode problem should be designed. Clearly, the problem of recognition can be viewed both as a "user-centered" issue (i.e. a user device should be able to detect which mode/modulation is present or to locate an object) and as a "Smart Space-centered" issue (i.e. the same operations are performed but from the SS point of view).

Figure 1 shows a possible representation of a smart space with BSs and UTs.

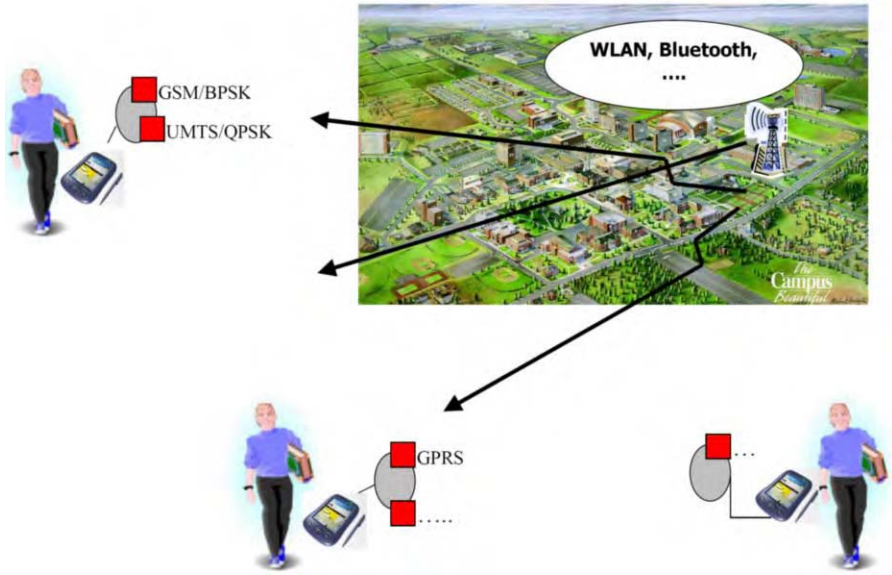



**Figure 1.** The SS radio context.

We can consider the following approaches to solving problems of object location and object attribute identification:

- a stand-alone approach;
- a cooperative approach.

In the state-of-the-art system, designed with a stand-alone approach, mode/modulation recognition is performed in two modules: a feature extractor and a classifier. The feature extractor is able to characterize modulated signals by using moments from the first to the fourth order [1], constellation shape [2], and time-frequency analysis [3]. The classifier chooses the modulation type based on the output data of the first module. The recognition process can be carried out using a decision theoretical approach or pattern recognition methods.

Polydoros and Kim [4] proposed a method to classify digital modulation by obtaining sufficient statistics in both the coherent and non-coherent cases. Tests were carried out for binary phase-shift keying (BPSK) and quadrature and orthogonal-quadrature phase-shift keying (QPSK, OQPSK). The procedures used three classifiers: a phase-based, a square-law and a decision theoretical classifier.

Using a Maximum Likelihood approach, Wei and Mendel [5] applied a classifier for 16-V29,16/32/64 quadrature amplitude modulation (QAM). The main target of this work was to define an upper bound for performances for any kind of classifier and a way to determine the minimum resources required to obtain the desired quality of service.

Other works make use of neural network classifiers. Nandi et al. [6] defined a method to classify analogical and digital modulation (AM,DSB,VSB,LSB,USB;FM, 2-4 ASK, 2-4 PSK, 2-4 FSK) by employing a multilayer perceptron with different numbers of hidden layers and for different Signal to Noise Ratio (SNR) values (10–20 dB).

Entropy analysis of wavelet packets performed in Shull [7] identified BPSK and QPSK signals. These signals were classified in a hybrid system in which two or more back-propagation NNs working as classifiers were combined in a NN integrator.

The location estimation problem was also addressed in the literature. The existing technologies can be grouped into four categories: [8]: space-based radio navigation systems, wireless LAN (WLAN) and the short-range connectivity, the sensor technology, and the cellular network based method.

No methodology mentioned above is able to guarantee good performance and availability in different (indoor and outdoor) scenarios. For example, techniques for GPS systems allow for good efficiency in outdoor situations [9] but fail in indoor scenarios. The use of radio cellular networks provides better efficiency if the mobile terminal is exposed to a sufficient number of base stations. However, it does not guarantee good performance in a rural environment in which the cellular coverage is not sufficient for trilateration of signals [10]. Similarly, WLAN and short-range technology allow good performances only in indoor situations [11,12].

There are also technologies for location identification based not on the transmission/reception of radio signals but on scene analysis methods produced by video sensors (e.g. camera sensors). Similarly to other location identification techniques, this technology is also susceptible to various kinds of issues such as obstructions in the line of sight [13]. In Table 1, a summary of location estimation methods is presented.

The location estimation methods require non-standard features in both the mobile terminal and the network. Solutions based on cellular systems and wireless LAN are in general problematic due to their location estimation accuracy, as shown in Table 1. Recently, a new location estimation method has been proposed [14]. This method is based on the statistical modeling of the received signal power linked to the distance. Each technique presented here for the state-of-the-art modulation/mode identification and location was designed using a non-cooperative scenario, and cooperative methods can be used to improve results. In recent years, signal processing for cooperative scenarios, refered to as collaborative signal processing, has been proposed [15].

**Table 1.** Location estimation techniques

| technology | technique | accuracy and precision | scale | limitations and barriers |
|---|---|---|---|---|
| GPS | trilateration of radio signals coming from satellites | 1–5 meter | 24 satellites worldwide | works well only in open spaces |
| WLAN | received signal strength (RSS) measurements and trilateration | 1–10 meters | 3 or 4 access points | covers short ranges only, susceptible to various interferences |
| CELLULAR PHONE | trilateration of signals from network base stations. | 10–100 meters (depending on network configuration) | one or more base stations per ~10 km^2 area in urban areas, fewer in rural areas | restricted by bandwidth available. susceptible to interferences |
| VIDEO CAM-ERA | object tracking, camera calibration | variable | single camera and multi-camera systems | susceptible to impediments and obstacles in view |

In [16], an example of cooperative/collaborative detection, classification and tracking of targets is described. This new kind of processing has led to an evolution in signal processing techniques, allowing for distributed and collaborative networks of devices and sensors. Studies in this direction are under development and are receiving great interest for their usefulness and potentials for performance and efficiency.

A cooperative approach would also be useful in an SS scenario for both attribute identification and location. Multiple users present within a smart space with their devices may be thought of as a distributed network. Several users could be interested in taking the same connection and devices would thus cooperatively try to identify the transmission mode offered by the SS. At the same time, BSs can try to detect in a collaborative manner the type of modulation format employed by the terminals. The location problem can thus be dealt with by using collaborative methods in which terminals as well as BSs can determine their position, but also share their position information with the other devices.

Due to the distributed nature of the problem, data fusion strategies can be used within the SS context. Possible fusion strategies applied to the problems of identification and location will be presented in Section 2 in greater detail and a possible reference scenario will be described.

## 2. System Model and Problem Statement

The concept of SS contains unresolved issues within the communication framework. The communication network infrastructure considered here is typically wireless to allow greater flexibility between users and the smart space, and vice versa. At the same time, the smart space communication infrastructure allows multimode communications because users can access it with different mobile devices based on different standards.

Multimodal communications allow for the possibility of designing devices able to support more than one communication standard. This new trend in communications makes it possible to provide a user with more powerful and reconfigurable UTs and BSs equipped with intelligent and adaptive signal processing techniques [17]. Clearly, the use of such techniques implies a new design methodology. Higher levels of flexibility, reconfigurability and scalability are necessary. A promising technology is the Software Defined Radio (SDR) [18], which allows implementation of programmable radio transceivers (reconfigurable by software) that are able to support multistandard and multimode communications.

In this SS reconfigurable context, two types of users can be present in the space: "cooperative" users, namely, users equipped with reconfigurable devices, and "non cooperative" users, who are not provided with any device.

In the first case, user devices form a mobile sensor network in the SS. Specifically, they can be considered as mobile nodes of the network, whereas the SS provides fixed access points (i.e., base stations for communications). In our work, attention is focused on cooperative users.

As a reference scenario of the model, an intelligent university campus is shown in Figure 2.

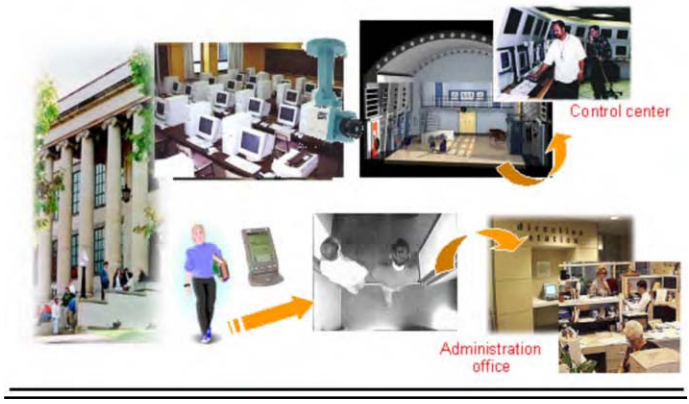



**Figure 2.** The SS: a possible reference scenario.

A student equipped with an intelligent and reconfigurable terminal enters the campus. The system contacts him with the best communication standard available to inform him as to the the easiest way to reach unknown destinations or about the SS resources that can be made available to him depending on personal tasks to be performed (e.g. a meeting with a professor, attending a lecture, etc.).

The UT contains detection capabilities and, after channel monitoring, tries to identify which kind of transmission mode is in use at that time. At the end of this process, the reconfigurable UT knows which mode the SS is using and thus identifies an attribute of the BS (the transmission mode). In a cooperative scenario, the task of identifying the mode or modulation used is performed by several UTs or BSs.

Location identification issues must also be considered in the process of producing the required information. The SS determines the context based on the user's position and offers certain services related to his location. The student might want to know his/a friend's location inside the campus. In this scenario, there exist two possible situations: SS location or user centered location.

The problems of mode identification and location involve two different levels:

1. sensing level: sensors perform sensing operations, e.g. detecting a radio phenomenon, the presence/absence of a signal, or location of the UT/BS to which a sensor belongs;
2. data fusion level: one can fuse decisions from the various sensors in the case of identification, or one can fuse information about different positions, as will be explained later on in this section.

The next two sections deal with the above two issues, i.e. mode identification and location, taking into account the previous division into two levels.

### *2.1. Attribute Identification*

First, the mode identification problem is considered. At the sensing level, mode identification techniques are used inside the UT/BS to make a decision on the available mode or modulation. Keeping in mind the cooperative scenario mentioned in the introduction, the identification process can be extended to more UTs/BSs forming a distributed network over the SS. Therefore, due to the distributed nature of the detection problem, a distributed decision-theory framework can be employed [19]. Specifically, in this work, the Bayesian theory is adopted [19]. This approach can be inserted into the theoretical decision methods of classification mentioned in the introduction. On the basis of the observed received signal corrupted by noise, the likelihood ratio at each detector is obtained. At this stage, a local decision is made at each detector.

To obtain greater efficiency of the identification procedure, information exchange regarding local decisions made by the BSs or UTs can be performed before making the final decision. In particular, this solution could be viewed in the context of a reconfigurable terminal based on SDR technology, for which a UT or a BS is equipped to support more than one radio communication standard.

It is possible to establish a connection for transmitting local decisions to the radio entities involved in the decision process. For example, such entities can be considered as a cluster (made up of a certain number of UTs or BSs) that can be thought of as a Personal Area Network (PAN) in which it is possible to use Bluetooth [20] for communications between the nodes. In this case, the radio object (a single node) should support the Bluetooth standard in a transparent and flexible way. At the same time, it can maintain the connection (using a different standard) with adjacent nodes (UTs or BSs) to continue the identification process.

In this scenario, data fusion techniques support the decision process in two ways: 1) physical data fusion, i.e. the various decisions are fused to improve the final decision regarding the attributes or the estimation of the final position; 2) the data fusion framework contributes to increase the level of knowledge of the SS.

The physical data fusion strategy can also be employed to make the final decision. The local decisions are sent to a centralized unit for further data fusion processing. At this stage, the combined use of distributed detection and data fusion improves the system's performance with respect to probability of detection.

The application of data fusion techniques can also be useful for the process of information association in the SS. More precisely, it is possible to make an association between the terminal with its identified attributes and the user to whom it belongs. In particular, we consider both the information provided by the detectors about the transmission mode and the information provided by the other sensors, e.g., the video cameras present in the SS, as depicted in Figure 3.

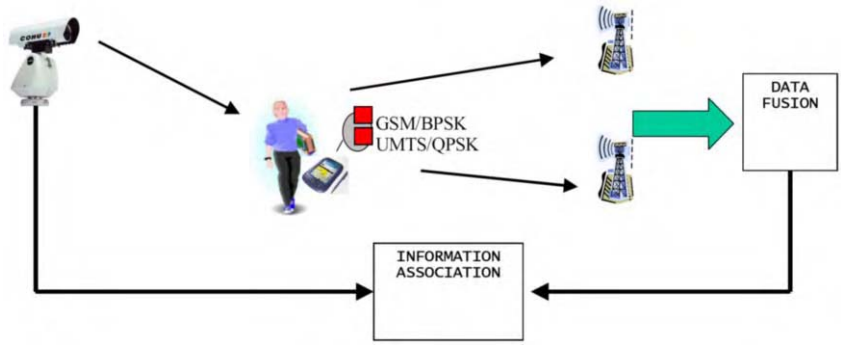



**Figure 3.** Data fusion for information association in an SS.

### *2.2. Location Estimation*

For location estimation at the sensing level, a BS/UT uses signal processing techniques for position determination based on different methodologies, as presented in Table 1. Two approaches to location estimation are proposed: a heterogeneous multisensor scenario and a homogeneous one.

In the heterogeneous multisensor environment, a data fusion method can be used to provide continuous position service in the SS. The proposed system is set in a multisensor environment, as any sensor is suitable for this environment. In realistic scenarios, it is impossible for single sensors to work accurately in all kinds of areas. Sensors have limited capabilities and are affected by their individual shortcomings. A multisensor solution improves location-determination performance as compared to a single sensor. Multiple-sensor systems provide operational benefits in specific applications: robust operational performance, extended spatial coverage, extended temporal coverage, increased confidence, reduced ambiguity, improved detection performance, enhanced spatial resolution, improved system operational reliability and increased dimensionality [21].

In an inhomogeneous sensor scenario, where the sensor/detector can use a state-of-the-art technique for location purposes, data fusion could assist in determining an improved location of a common radio object. More BSs could estimate the position of a radio device belonging to the user in the Smart Space. Another scenario describes a group of users trying to determine the position of a common object, e.g., a base station.

In both cases, different fusion strategies can be applied:

1. direct fusion of local position estimates;
2. cooperative location estimation approach and further data fusion.

In the first case, each radio object performs its position estimate, which is sent to the fusion center without any communication between detectors.

In the second case, a cooperative position estimation is first performed according to each radio entity position estimates. An information exchange and fusion is then performed to estimate of the position of the common object of interest.

### *2.3. Advantages of This Approach*

In general, position information and the knowledge of the transmission mode could be advantageous, particularly when employing reconfigurable hardware (i.e.: hardware

which is able to reconfigure itself to support the proper identified standard with its modulation and coding and, in general, different signal processing methods). In this context, the knowledge of location and modes by the BSs or terminals should facilitate the reconfiguration process.

For the SS, position information makes it possible to choose the best BS for the user. At the same time, the possibility of having preliminary information regarding transmission mode allows one to reconfigure one's signal processing tools to support the identified mode.

Position estimation allows the use of an adaptive technique for energy/power consumption at base stations or terminals, as energy is also related to the distance between stations and terminals.

A cooperative approach could generalize this method and produce a global knowledge for the SS, of the radio attributes of the BSs and the terminals, and of their positions.

## 3. Case Studies

In this section, two case studies concerning both stand-alone and cooperative identification of the modulation format are presented. In the second case, cooperative and distributed users are considered and a collaborative procedure to identify the modulation format is described. A statistical procedure for location determination that is based on the link between the path-loss model and the probability density function of a received signal is proposed. Special attention is given to the link between dependence on relative distance and the identification procedure.

### *3.1. Modulation Identification: Problem Statement*

Let us consider Figure 1. Users are able to offer two different modulation formats: BPSK and QPSK, belonging to the more general phase-shift keying (PSK) modulation family. The BSs have no previous information on the type of modulation the SS is using for communication. Therefore, a modulation identification technique is necessary for classifying the received signal as BPSK or QPSK.

During detection, the common observed phenomenon is the transmission of:

- a BPSK /QPSK signal;
- noise only.

Therefore, a binary hypothesis test under two possible hypotheses is considered: the presence of one of two signals (BPSK or QPSK) or of noise only.

To obtain a possible expression for the observed/received signal so as to carry out the binary test, we need:

1. the expression for the received signal in the two cases of interest (BPSK or QPSK);
2. the characterization of the noise that corrupts the transmitted signal.

These two requirements characterize the decision test. Let the received waveform $r(t)$ be of duration $0 \leq t \leq T$; it is described in a different manner under the signal-present hypothesis or the noise hypothesis. A general statement for the binary test is the following:

$$r(t)=\begin{cases} s_{BPSK}(t)/s_{QPSK}(t)+n(t) & H_1 \\ n(t) & H_0 \end{cases} \tag{1}$$

$n(t)$ is the Additive White Gaussian Noise (AWGN) with a two sided power spectral density of $N_0/2$ W/Hz.

A path loss channel effect has also been considered. In particular, the following model was adopted [22].

$$L_P=\begin{cases} 32.45+20\log(f\cdot d) & d\le 8\,m \\ 58.3+33\log(d/8) & d>8\,m \end{cases} \tag{2}$$

where $f$ is the operating carrier frequency, and $d$ is the relative distance from the source. This model assumes a Line-on-Sight (LOS) propagation for the first 8 meters.

In general, the signal $s(t)$ is the PSK-modulation family format that can be expressed in different ways. In this work, the quadrature representation is adopted. Therefore, the signal is defined by the following more general expression [4]:

$$s(t)=s_I(t)\cos(\omega_c t+\theta_c)-s_Q(t)\sin(\omega_c t+\theta_c) \tag{3}$$

where $s_I(t)$ and $s_Q(t)$ are the in phase and quadrature signal components respectively, and $\theta_C$ is the carrier phase.

In the case of PSK modulation, the two signals assume the following expressions [4]:

$$s_I(t)=\sum_n d_{I,n}p(t-nT_s) \tag{4}$$

$$s_Q(t)=\sum_n d_{Q,n}p(t-nT_s) \tag{5}$$

where:

$$d_{I,n}=\cos\theta_n$$
$$d_{Q,n}=\sin\theta_n$$

and $T_s$ is the time duration of the transmitted symbol.

$$\theta_n\in\{2\pi/M(i-1)+\theta_0; i=1,....,M\} \tag{6}$$

where $M$ is the number of transmitted symbols, and $\theta_0$ is an arbitrary rotational phase. In the case of BPSK modulation $\theta_0=0$ and $\theta_n\in\{0,\pi\}$ whereas in the case of QPSK, $\theta_0=\pi/4$ and $\theta_n\in\{i\pi/4, i=1,3,5,7\}$.

Under the $H_1$ hypothesis, the received signal is described by the following expression (7):

$$r(t) = L_p s_{PSK}(t) + n(t) =$$
$$= L_p \left[ \sum_n d_{I,n} p(t - nT_s - \varepsilon T_s) \cos(\omega_c t + \theta_c) - \sum_n d_{Q,n} p(t - nT_s - \varepsilon T_s) \sin(\omega_c t + \theta_c) \right] + n(t)$$

where $\varepsilon$ is the timing offset between the transmitted and received signals.

Taking into account the signal shape seen before and after its substitution in the test, it is possible to obtain the likelihood ratio. Before proceeding with the definition of the problem of finding the likelihood ratios and the thresholds in the stand-alone and distributed contexts it is necessary to describe the detector. We can classify the various detectors in three cases [4]:

- synchronous coherent case: the carrier phase and the symbol timing offset are assumed to be known at the detector;
- synchronous non coherent case: the carrier phase is unknown and considered as a random variable, while the timing offset is assumed to be null;
- asynchronous case: both the timing offset and the carrier phase are regarded as random variables.

In our work, we have considered the first case. Studies are currently under development to extend the proposed cooperative modulation identification to the other two cases.

### *3.2. A Stand-Alone Technique Based on Decision Theory*

In this subsection, a stand-alone technique based on the classical detection theory [23] is described. Considering the general statement presented in the previous section, a detection procedure based on the likelihood ratio is proposed. A single detector is considered. The likelihood ratio and the threshold test are obtained in the case of an AWGN channel with a path loss term given by equation (2).

Let $\Lambda(r(t))$ be the likelihood ratio at the detector, defined as follows:

$$\Lambda(r(t)) = \frac{p(r(t)/H_1)}{p(r(t)/H_0)} \tag{8}$$

The conditional probability density functions $p(r(t)/H_j)$ $j$, $i$ = 0, 1, are Gaussian.

Given the expression for the binary test in (1), the following log likelihood ratio is obtained:

$$\ln \Lambda_1(y_1) = \frac{2}{N_0} \int_0^T L_p r(t) s_j(t) dt - \frac{1}{N_0} L_p^2 \int_0^T s_j^2(t)\, dt \underset{H_0}{\overset{H_1}{\gtrless}} 0 \tag{9}$$

After some mathematical steps, we have the following expression:

$$\ln \Lambda_1(y_1) = \frac{2}{N_0} \int_0^T L_p r(t) s_j(t) dt \underset{H_0}{\overset{H_1}{>}} \frac{L_p T}{2N_0} \tag{10}$$

where $s_j(t)$ $j = 1, 2$, is a BPSK signal for $j$ = 1 or a QPSK signal for $j$ = 2. As can be seen, the likelihood ratio and the threshold depend on the relative distance contained in the path loss that should be estimated.

### *3.3. The Cooperative Modulation Identification Procedure*

The binary Bayesian distributed detection can be obtained by two detectors. We consider a parallel distributed detection, as presented in Figure 4.

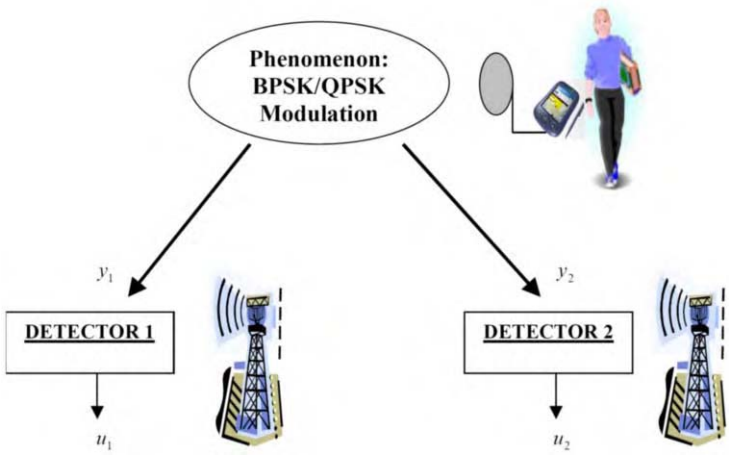



**Figure 4.** Distributed modulation identification.

The local observations at each detector are denoted by $y_1$ and $y_2$, with a joint conditional density $p(y_1, y_2 / H_i)$, $i$ = 0, 1, 2. In our case, $y_1$ and $y_2$ are the received signals $r(t)$.

The *a priori* probabilities are $P_0$, $P_1$ and are taken to be equal, supposing the same probability for the two hypotheses. The local decisions at the two detectors are indicated as $u_i$, $i$ = 1, 2. and are given by:

$$u_i = \begin{cases} 0 & \underset{y_1}{decide} \quad H_0 \\ 1 & \underset{y_2}{decide} \quad H_1 \end{cases}$$

The cost of the detector 1 deciding $H_i$, and the detector 2 deciding $H_j$, when $H_k$ is present, is denoted by $C_{ijk}$, $i,j,k = 0, 1$.

We also make the reasonable assumption that the cost of a wrong decision made by one sensor is higher than the cost of making a correct decision regardless of the decision of the other sensor. Local decision rules have to be obtained that minimize the Bayes risk function $\Re$ given by [19]:

$$\begin{aligned}\Re &= \sum_{i,j,k} \int_{y_1,y_2} p(u_1,u_2,y_1,y_2,H_k)C_{ijk} \\ &= \sum_{i,j,k} \int_{y_1,y_2} P_k C_{ijk} p(u_1,u_2/y_1,y_2,H_k)p(y_1,y_2/H_k)\end{aligned} \tag{11}$$

Finding the expressions for the two detectors in terms of the likelihood ratio of the risk, a set of equations (the decision rules) can be obtained for the two detectors. These equations depend on the likelihood ratios at the two detectors. A simultaneous solution of the decision rule inequalities yields the observation space partitions at the two local detectors.

In this part, we omit all steps to obtain the final expression for the Bayes risk function (in terms of likelihood ratio) that can be found in [19].

Let $\Lambda_1(y_1)$ and $\Lambda_2(y_2)$ be the likelihood ratios at detector 1 and detector 2, defined as follows [19]:

$$\Lambda_1(y_1) = \frac{p(y_1/H_1)}{p(y_1/H_0)} \tag{12}$$

$$\Lambda_2(y_2) = \frac{p(y_2/H_1)}{p(y_2/H_0)} \tag{13}$$

Developing all necessary mathematical steps and assuming the following costs [19]:

$$\begin{aligned}&C_{000} = C_{111} = 0 \\ &C_{001} = C_{100} = C_{011} = C_{101} = 1 \\ &C_{110} = C_{001} = k\end{aligned}$$

as suggested in [19], the following inequalities for detectors 1 and 2 are obtained:

$$\Lambda_1(y_1) \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0}{P_1} \frac{(k-1)-(k-2)p(u_2=0/H_0)}{(k-1)-(k-2)p(u_2=0/H_1)} = t_1 \tag{14}$$

$$\Lambda_2(y_2) \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0}{P_1} \frac{(k-1)-(k-2)p(u_1=0/H_0)}{(k-1)-(k-2)p(u_1=0/H_1)} = t_2 \tag{15}$$

The inequalities (14) and (15) specify the decision rules at detectors 1 and 2.

In order to proceed with the analysis, the previous expressions can be divided into two parts: the right-hand side and the left-hand side. We start with the right-hand side of only inequality (14). The same reasoning can be repeated for (15). The conditional probability density functions $p(y_i / H_j)$, $j$, $i$ = 0, 1, are Gaussian. Therefore, we use for simplicity the natural logarithm of the likelihood ratio in (14); after some mathematical steps on (12), the following expression is obtained:

$$\ln \Lambda_1(y_1) = \frac{1}{\sigma_0^2} \int_0^T L_p r(t) s_j(t) dt - \frac{1}{2\sigma_0^2} L_p^2 \int_0^T s_j^2(t) dt \underset{H_0}{\overset{H_1}{\gtrless}} \ln t_1 \tag{16}$$

where $s_j(t)\ \ j = 1, 2$, is a BPSK signal for $j$ = 1, or a QPSK signal for $j$ = 2, while $\sigma_o^2 = \mathrm{var}(y_1 / H_0) = N_0 / 2$.

We now regard the left-hand side. It is necessary to find the expressions for the conditional probability density functions, namely, $p(u_i / H_j)$. In the present case, an AWGN noise is considered. Thanks to the central limit theorem, these quantities can be considered to be Gaussian with the following statistical parameters dependent on the current hypothesis:

$$H_0 = \begin{cases} m_3 & E\{u_i\} = E\{\Lambda_i\} \\ \sigma_3^2 & \mathrm{var}\{u_i\} = \mathrm{var}\{\Lambda_i\} \end{cases} \qquad i = 1, 2$$

$$H_1 = \begin{cases} m_4 & E\{u_i\} = E\{\Lambda_i\} \\ \sigma_4^2 & \mathrm{var}\{u_i\} = \mathrm{var}\{\Lambda_i\} \end{cases} \qquad i = 1, 2$$

where $m_4$ and $\sigma_4^2$ are different in the case of a BPSK or QPSK signal.

Therefore:

$$p(u_2 = 0 / H_1) = \frac{1}{2}\left(1 + erf\left(\frac{\ln t_2 + C - m_4}{\sqrt{2}\sigma_4}\right)\right) \tag{17}$$

$$p(u_2 = 0 / H_0) = \frac{1}{2}\left(1 + erf\left(\frac{\ln t_2 + C - m_3}{\sqrt{2}\sigma_3}\right)\right) \tag{18}$$

where:

$$C = \frac{L_p^2 T}{4\sigma_0^2}$$

$$erf(k) = \frac{2}{\sqrt{\pi}} \int_0^k e^{-\lambda^2} d\lambda$$

The left-hand side of the inequalities will now be considered. It is easy to demonstrate [19] that the threshold at detector 1 depends on the threshold of detector 2, defined as $t_2$.

In the case of Gaussian probability densities functions, it is possible to obtain the following thresholds:

$$t_1 = \frac{k + (2-k) erf\left( \frac{\ln t_2 + C - m_3}{\sqrt{2}\sigma_3} \right)}{k + (k-2) erf\left( \frac{\ln t_2 + C - m_4}{\sqrt{2}\sigma_4} \right)} \tag{19}$$

$$t_2 = \frac{k + (2-k) erf\left( \frac{\ln t_1 + C - m_3}{\sqrt{2}\sigma_3} \right)}{k + (k-2) erf\left( \frac{\ln t_1 + C - m_4}{\sqrt{2}\sigma_4} \right)} \tag{20}$$

The solution yields the thresholds $t_1$ and $t_2$.

### *3.4. Location Issues in the SS Context*

This work focuses on attribute identification, in particular, modulation identification in an SS context. However, location information appears to be an important quantity to be estimated and to be known by the SS. We have shown that, during the modulation identification procedure, the relative distance between a generic UT and a BS can be linked to the likelihood ratio and the threshold test through the path loss model.

Location may be determined from an SS centered or a user-centered point of view. In this case, a stand-alone or a cooperative position estimation technique may also be used. Obviously, the location estimation procedure should be applied after the identification procedure. The most important location methods are based on the reception of signals. Therefore, as a first step, it is necessary to identify the presence of a signal (using, for example, an identification procedure such as those presented previously) and then to determine the position.

In the case of the stand-alone method, a statistical location estimation procedure can be adopted as a possible solution. The focus of this approach links the channel model, (in particular, the path loss model), to the probability density function of the observed/received signal. In this case, the path loss model is expressed in terms of the relative distance between the user terminal and the access point, as demonstrated in Section 3. More precisely, the binary test presented in this paper involves a composite hypothesis under which the unknown deterministic parameter is the relative distance between a BS and a terminal. The relative distance is linked to the threshold test and the likelihood ratio defined as stated in the above.

The next step consists in linking the distance to the probability density function of the received signal. The path loss model (and hence the relative distance) affects important parameters involved in the design of the probability density function of the observed signal. In particular, it is possible to demonstrate that the distance influences the mean of the distribution. This quantity can change, depending where the user is located with respect to the source. From the expression for the received signal, it is possible to derive the required statistical parameters.

Let us consider the received signal under the $H_1$ hypothesis:

$$r(t) = L_p s(t) + n(t)$$

As shown before, the probability density function of the received signal is Gaussian. A Gaussian density function is completely characterized by its mean and variance.

Another method could be based on the received-power measurement (RSS), which depends on the distance. If the closed-form expression for the RSS is obtained from the measurement, it is possible to compute the relative distance.

Studies to correctly formalize these problems are currently under development.

A cooperative procedure to estimate the position can also be proposed. The aim of this procedure is to obtain:

1. estimates of the position for each BS separately, and then share this information between all BSs in order to establish which BS can optimally serve the user;
2. a cooperative estimation, according to which a group of BSs estimate the position of the common object of interest (the user).

In the first case, the previously described statistical method can be employed as a possible position estimation technique.

A SS-centered location or a user-centered one can also be adopted.

In the first scenario, several BSs try to estimate the position of a UT in the SS, as shown in Figure 5.

Each BS estimates the relative distance separately (using, for example, the presented stand-alone method), and after a collaborative information exchange between the BSs about the position information, the final estimate is obtained. Here a communication link should be established between the two BSs. As seen before, position estimation is also useful in the identification procedure. For example, in the case of cooperative identification, the two BSs should exchange the two thresholds as well as the estimates of the relative distances (in order to solve the problem), because the thresholds depend on the distances as shown in subsection 3.3.

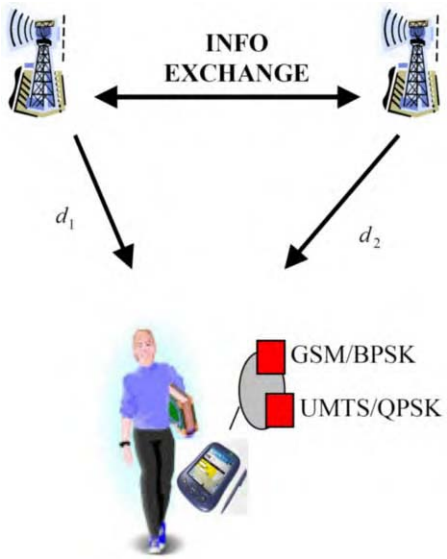



**Figure 5.** Location estimation SS centered approach.

The second scenario includes UTs that try to estimate the BS's position, as presented in Figure 6.


**Figure 6.** Location estimation user centered approach.

In this scenario the same cooperative framework can be adopted, but a substantially different construct is present with respect to the previous scenario.

A communication link has to be established between the two UTs. Due to the geometry of the problem, it is reasonable to choose a short-range communication between the two UTs by employing a standard like Bluetooth for the information exchange. When reconfigurable terminals based on SDR technology are used, such a solution can be realized. However, in the case of a collaborative BS scenario, a transmission link is easier to establish.

A short-range communication between users also imposes a constraint on the relative distance between the two UTs: the constraint can be formalized by a mathematical expression that links the two distances. Such information can be used for the computation of the final position of the BS.

## 4. Numerical Results

In this section, some preliminary numerical results obtained by the stand alone and cooperative modulation identification methods described in Section 3 are presented and discussed.

The system has been set up and simulated in MATLAB 6.0.

The following assumptions have been made:

- AWGN channel;
- a path loss model given by equation (2);
- coherent detection;
- simulation at intermediate frequency fixed at $f_C = 60$ MHz;
- the distance from the source of a signal is known or it is possible to estimate it by a method presented in the previous section.

Following the proposed stand-alone method, a single detector was simulated. The decision rule and the threshold were implemented according to the expressions obtained in the subsection 3.2. In the following, numerical results in terms of probability of detection (expressed as relative frequency) are reported. The detector observes the received signal and has to be able to classify the presence of a QPSK or BPSK signal or of noise only. The relative distance between the detector and the signal source has been fixed at $d = 12$ meters.

In Figure 7, the probability of detection (expressed in terms of relative frequency) in the case of BPSK modulation is presented. In particular, it was obtained for different Signal-to-Noise Ratios (*SNRs*) and a different number *N* of samples accumulated before making the final decision.

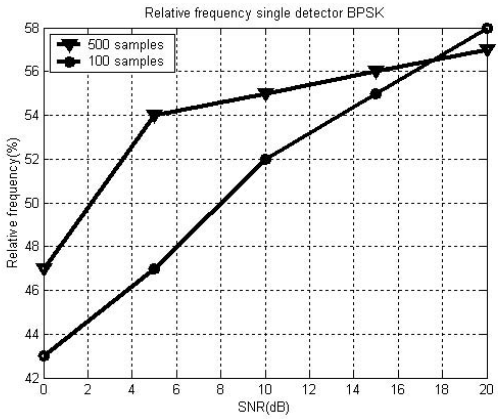



**Figure 7.** Probability of detection of a single detector for BPSK at different *SNRs*.

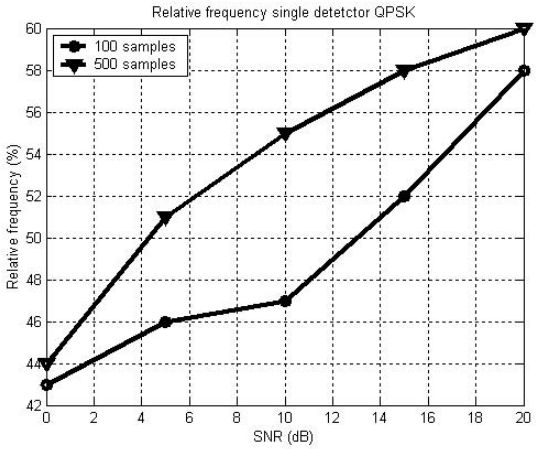



**Figure 8.** Probability of detection of a single detector for QPSK at different *SNRs*.

In Figure 8, results under the same assumptions were obtained in the case of QPSK modulation.

Some results regarding the distributed identification are now presented.

We have assumed the situation depicted in Figure 4. Two detectors (DM1 and DM2) were considered, in particular, two BSs in the SS. In this preliminary work, any information exchange between the detectors is taken into account. The phenomenon to be detected is the presence/absence of the modulation format BPSK or QPSK employed by the user in the SS. More precisely, the following real situation was simulated. Two detectors are placed at different positions with respect to the source of signals. In particular, DM1 is placed at $d = 12$ meters from the source, whereas DM2 is placed at $d = 7$ meters. This implies a different path loss term in the received signal.

According to the distributed detection theory, equations (14) and (15) were simulated and implemented.

The results are given in terms of probability of detection (expressed as relative frequency) for the two detectors. The results were obtained for Signal to Noise Ratios (*SNRs*).

Figures 10 and 11 show the probabilities of detection (expressed in terms of relative frequency) for the two detectors in the case of a BPSK signal. In particular, such results were obtained by varying the *SNR* from 0 to 20 dB, and by considering different number $N$ of samples accumulated before making the decision. In our case, we have considered $N = 100$ and $N = 500$. The simulations will be repeated for different values of the costs, in particular, for different values of $C_{110} = C_{001} = k$ and for one of the two detectors. Such values can be variable and influence the detector thresholds. The shown results are those for which the value of $k$ is the best, namely, the optimum value that allows the optimum threshold according to the distributed detection theory [19]. The decision thresholds have been calculated by the Newton algorithm and for a value of $k$ variable ranging from 5 to 10. In Figure 9, the solutions for the two thresholds are shown.

The case of $t_1 = t_2 = 1$ does not allow the minimization of the risk function. In the other two cases, $t_1$ has been fixed at one value and $t_2$ at another. The minimum value for the risk function is obtained.

Figures 12 and 13 give the resulting probability of detection (expressed as relative frequency) in the case of QPSK modulation.

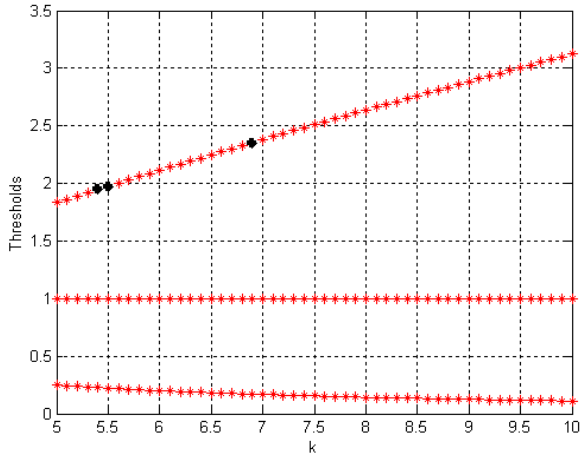



**Figure 9.** The two-threshold test for different values of the cost *k*.

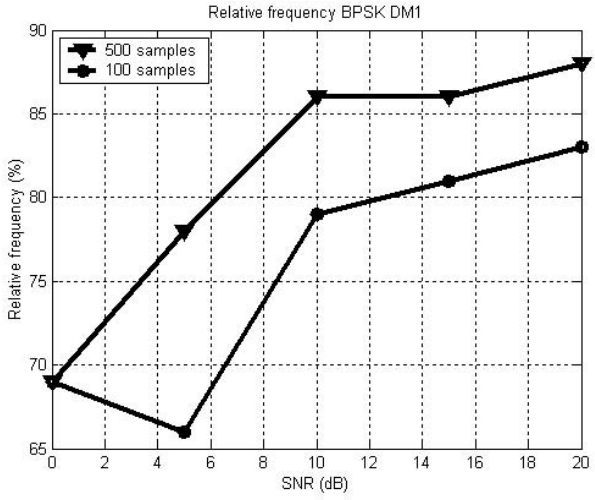



**Figure 10.** Probability of detection for BPSK at DM1.

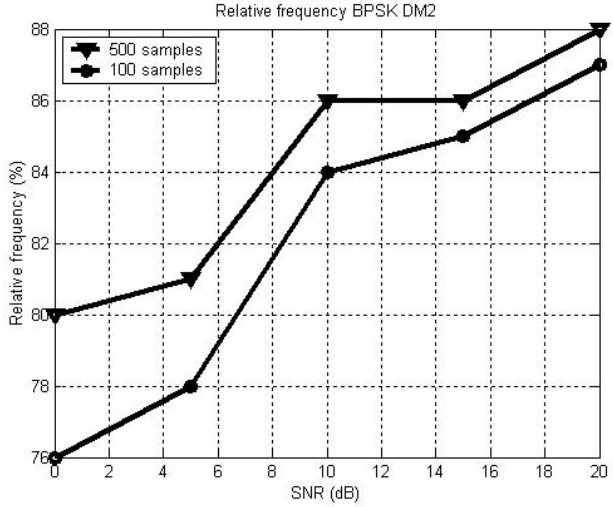



**Figure 11.** Probability of detection for BPSK at DM2.

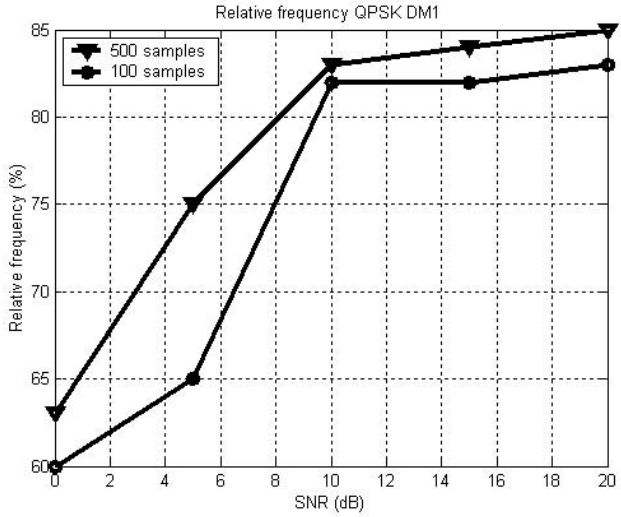



**Figure 12.** Probability of detection for QPSK at DM1.

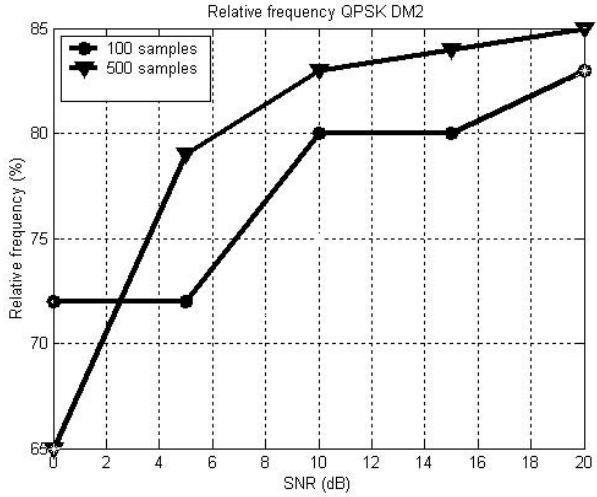



**Figure 13.** Probability of detection for QPSK at DM2.

The results shown in both cases depend on the number of samples accumulated to make the local decision. With more samples, the probability of detection increases. Moreover, the *SNR* influences detection performances. In addition, the proposed distributed identification represents an improvement over identification by a single detector.

In the case of the distributed approach, the results show that the performances of the two DMs are different, depending on their distances from the signal source: DM2 provides better performances than DM1.

## 5. Conclusions and Future Prospects

In this paper, the concept of SS has been defined with special attention to two issues: object attribute identification and object location. In particular, the interest has been focused on radio objects present in the SS context (UTs and BSs), and on the attribute identification problem. Attribute identification depends on knowing which kind of communication standard a BS is using or which kind of modulation a UT can use for transmission in the SS. At the same time, such objects need to be located. The two problems can be addressed from an SS or user point of view.

Moreover, the possibility of using a stand-alone or a collaborative approach to the identification problem has been described. Regarding location, some preliminary studies have been presented and a possible solution has been proposed.

The use of a data fusion strategy can improve system efficiency and performance. Two case studies have been considered: a stand-alone and a cooperative modulation estimation. The first study is based on classical decision theory; the second on distributed detection.

A possible stand-alone location identification technique has been described. This technique is based on the knowledge of channel characteristics and on the possibility of linking the relative distance to the statistical model of a received signal. Some numerical results on the stand-alone and collaborative modulation identification procedure

have been reported. In particular, the modulation format to be recognized was BPSK or QPSK.

Future work on the distributed identification problem will include the possibility of using communications between detectors during the decision process before making the final decision regarding the mode. At the same time, the application of a data fusion strategy to make the final decision will be considered. In the two identification approaches, the extension to a non-coherent detector case will be investigated and the channel model will be modified by taking into account multipath effects.

The same considerations can be made for the location estimation problem. Finally, the proposed stand-alone statistical location method will be developed.

## References


[1] A. Swami, B.M. Sadler, "Hierarchical Digital Modulation Classification Using Cumulants," *IEEE Trans. on Commun.*, Vol. 48, No. 3, March 2000, pp. 416–429.

[2] B.G. Mobasseri, "Digital Modulation classification using constellation shape," *Signal Proc,* Vol. 80. No. 2, February 2000, pp. 251–277.

[3] H. Ketterer, F. Jondral, A.H. Costa, "Classification of Modulation Modes using Time-Frequency Methods," *Proc. of ICASSP* 99', pp. 2471–2474, March 1999.

[4] K. Kim, A. Polydoros, "On the detection and classification of quadrature digital modulation in broadband noise," *IEEE Trans. on Commun.*, Vol. 38, No. 8, August 1990, pp. 1199–1211.

[5] W. Wei, J.M. Mendel, "Maximum likelihood classification for digital amplitude-phase modulations," *IEEE Trans. on Commun.*, Vol. 48, No. 2, February 2000, pp. 189–193.

[6] Nandi Azzouz, "Modulation Recognition using artificial neural network," Signal Processing 56 (1996) 165–175.

[7] D.R. Shull, "Modulation classification by wavelet decomposition entropy analysis," PH.D Dissertation, University of Missouri-Rolla, 1999.

[8] Y. Zhao, "*Vehicle Location and navigation Systems*," Norwood, MA: Artech House, 1997.

[9] Elliott D. Kaplan (Editor), "Understanding GPS principles and applications" – Mobile Communications series, Artech House Publishers, Boston – London 1996.

[10] C. Drane, M. Macnaughtan, C. Scott, "Positioning GSM Telephones," *IEEE Commun. Magaz.*, Vol. 36, No. 4, April 1998, pp. 46–59.

[11] T. Kitasuka, T. Nakanishi, A. Fukuda, "Location Estimation System using Wireless Ad-HOC Network," *Proocedings of the 5th Symposium on Wireless Personal Multimedia Communications*, *WPMC02*, October 27–30 2002, Sheraton Hotel, Honolulu Hawaii.

[12] P. Prasithsangaree, P. Krishnamurthy, and P.K. Chrysanthis. "On Indoor Position Location with Wireless LANs." *Proc. of the 13th IEEE Int'l Symposium on Personal, Indoor, and Mobile Radio Communications*, Lisbon, Portugal, Sept. 2002.

[13] T.N. Tan, G.D. Sullivan, and K.D. Baker, "Recognizing objects on the ground-plane," *Image and Vision Computing*, Vol. 12, No. 3, pp. 164–172, 1994.

[14] T. Roos, P. Myllymaki, H. Tirri, "A statistical modelling approach to location estimation," *IEEE Trans. on Mobile Comp.*, Vol. 1, No. 1, Jan–March. 2002, pp. 59–69.

[15] Special Issue on "Collaborative signal and information processing in microsensor network," *IEEE Signal Proc. Magaz.*, Vol. 19, No. 2, March 2002.

[16] D. Li, K.D. Wong, Yu Hen Hu, A.M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Proc. Magaz.*, Vol. 19, No. 2, March 2002, pp. 17–29.

[17] M. Mehta, N. Drew, G. Vardoulias, N. Greco, C. Niedermeier, "Reconfigurable Terminals: an overview of architectural solutions," *IEEE Commun. Magaz.*, No. 39, Vol. 8, August 2001, pp. 82–89.

[18] J. Mitola III, "*Software Radio Architecture*," John Wiley and Sons, NewYork, 2000.

[19] Pramod K. Varshney, "*Distributed detection and data fusion*," Springer-Verlag, New York, 1997.

[20] "Specification of the Bluetooth System," Bluetooth standard, Specification Volume 1. Available at www.bluetooth.com.

[21] E. Waltz and J. Llinas, *"Multisensor data fusion"*, ISBN 0-89006-277-3, 1990 Artech House, Inc.

[22] A. Kamerman, "Coexistence between Bluetooth and IEEE 802.11 CCK Solutions to Avoid Mutual Interference," Lucent Technologies Bell Laboratories, Jan. 1999.

[23] H.L. Van Trees, "Detection, Estimation, and Modulation Theory – Part I: Detection, Estimation and Linear Modulation Theory," Wiley, New York: 1968.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Issues in Multicamera Dynamic Metadata Information Extraction and Interpretation for Ambient Intelligence

Luca MARCHESOTTI, Giuliano SCOTTI and Carlo REGAZZONI
*DIBE University of Genoa – Via dell'Opera Pia 11a Genova, Italy*
*carlo@dibe.unige.it*

**Abstract.** Video Analysis still represents a challenging issue in automatic event understanding and interpretation in applications of Ambient Intelligence. To enhance the performance of these systems, and to solve related problems, a multisensor approach is presented and ported to the video processing domain using the classic Data Fusion formalisms.

**Keywords.** Smart spaces, multisensor tracking, event analysis, data fusion

## 1. Introduction

Ambient Intelligence (AmI) is nowadays a mainstream concept which gathers several heterogeneous disciplines spanning from Telecommunications to Artificial Intelligence and from Machine Vision to Microelectronics. In addition, AmI is also a way of rethinking traditional technologies with a common denominator: analyze multiple, heterogeneous data and react to appropriate "stimuli" in order to show some degree of "intelligence" in the provision of services to users. As with all major movements, it took origin in a vision: Mark Weiser more than twenty years ago [1] first imagined a set of heterogeneous devices able to cooperate in order to understand the state of a particular environment and appropriately instantiate a customized communication with people populating an environment of interest. In order to formalize Weiser's conceptual framework, ISTAG (Information Society Technology Advisory Group) gives in [2] a definition of AmI that points out that it should provide technologies to support human interactions and to surround users with intelligent sensors and interfaces. According to this, Brooks in [3] states that an Intelligent Environment has to make computation "ready-at-hand," putting computers out into the real world of people, rather than putting people into the virtual world. In [4] Starner focuses on a key issue of Wearable Computing Systems that also plays a fundamental role in Ambient Intelligence: the capability of context sensing.

The EU community provided, in 2001, reference scenarios for Ambient Intelligence [5] in order to highlight possible application fields. This raised a widespread interest in AmI with projects such as VICOM [6] or PER2 [7] whose main goal is to integrate existing technologies in the field of Video Processing with advances in the Data Fusion domain. Other pioneer projects are Aura [8] in the field of distributed computing and oxygen [9]: the first proposes an innovative system for heterogeneous

services which persist regardless of location; the latter develops an architecture enabling pervasive, human-centered computing through a combination of specific user and system technologies.

Another reference AmI system is the "smart classroom" [10]; it proposes a physical experimental environment, integrating a multimodal human computer interface with modules collaborating through an inter-agent communication language in order to provide a smart space for remote learning. Examples of these kinds of systems are directed toward applications able to integrate "awareness" (i.e. identification and tracking), "intelligence" (i.e. adaptivity), and natural interaction. Our vision defines AmI systems as a set of virtual entities that possess three fundamental capabilities: analysis, awareness, and interaction [11].

The architecture that is presented here has been designed based on a scenario compliant with the ones proposed by ISTAG. The inspiring scenario [6] is contextualized in a University Campus, intended as a place in which users (mainly students) receive heterogeneous services. In the scenario, students enter the campus field which is entirely covered by sensors and actuators such as cameras, positioning devices, temperature or presence detectors, directional microphones, WLAN, GSM/GPRS and UMTS antennas, automatic opening/closing systems and automatic informative panels. The AmI architecture covers, with its scope of action/detection, outdoor environments as well as typical indoor university spaces (interactive laboratories and lecture rooms). The student equipped with an intelligent terminal (i.e. PDA, 3-G mobile phone) gets to the campus and the system localizes him with a set of distributed static and dynamic (pan-tilt) video cameras. The system tries to estimate the identity of the user through a variety of different techniques (e.g.: facial recognition, behavior analysis, etc.) and contacts him with the best communication standard available, estimated through a mode identification procedure, which configures itself to exchange information with the user. For example, a student has set out to perform a personal task (e.g. a meeting with his professor, attending a lecture, etc.) at some specific location. The system provides information as to the most efficient route to that or any other location the student may wish to reach relative to his present location. The student is also informed about the AmI's available resources (e.g.: PCs in the lab, new services available or campus news).

## 2. Architectural Issues in AmI Systems Design

Due to its inherent distributed and pervasive nature, an AmI system must be structured using a modular approach specifically designed to surround people that receive AmI services. A user centered closed loop (Figure 1) is here proposed as an inspiring scheme for the design of an AmI architecture. As it can be seen, the loop is composed of passive (i.e.: Sense, Analyze) and active (i.e.: Decide, Act) steps enabling the AmI to create a semantic representation of events of interest, and to make decisions based on inferences about the users or the system itself.

These inferences can take two forms: *informative messages* directed toward the user who becomes the subject of a multimodal communication and *physical actions* performed by the system through various kinds of actuators.

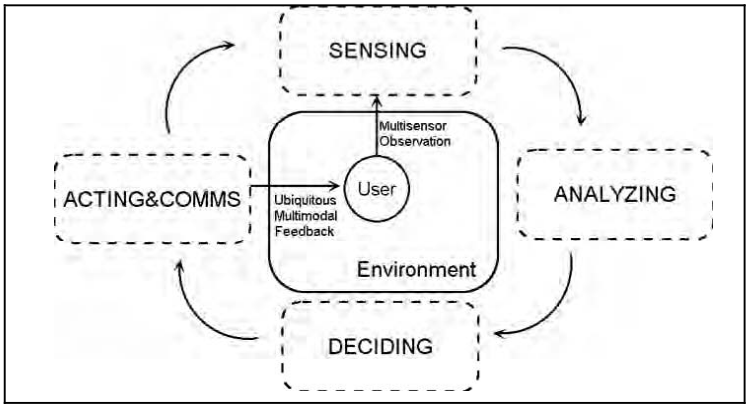



**Figure 1.** The user centered closed loop.

### *2.1. A Logic-Functional Architecture for AmI Applications*

Figure 2 depicts the logic-functional architecture of the presented system. The main steps of the user-centered loop are implemented in four different clusters. A set of heterogeneous sensors (i.e.: Sensing Cluster) collects data about the external world (users) whereas a Video Analysis Module (VAM) processes multimedia data. An Agent Manager Module (AMM) coordinates a society of agents [11] devoted to the collection of contextual information (e.g.: network load on computational units forming the AmI, CPU load, temperature in rooms, etc.). A Data Fusion Module (DFM) completes the Analysis Cluster by fusing, into a common representation, video and contextual data. The third, a Decision Cluster, bases future action/communication decisions on data coming from the DFM and on past experiences stored in a long-term database. The Action and Communication Cluster contains all the tools which are useful for presenting the information to users, and put in practice, through actuators, the decisions of the AmI system.

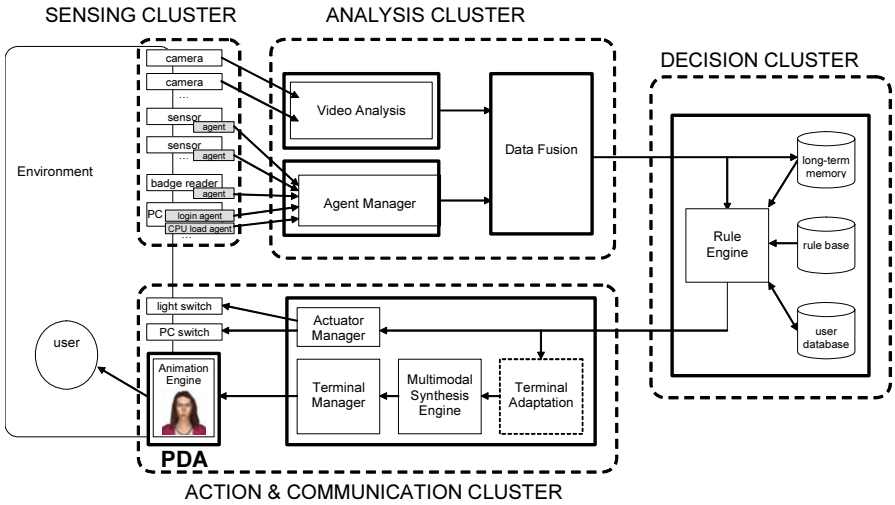



**Figure 2.** The overall logic-functional architecture of the system.

In this paper, the Video Analysis Cluster is the main topic of interest and will be thoroughly analyzed in Section 3.

## 3. The Video Analysis Module

The Video Analysis Module (VAM) in Figure 3 represents the primary source of information for the extraction of semantic data within the Analysis Cluster (Figure 2). It works in parallel with the AMM and handles a homogeneous network of Video Cameras performing synchronized A/D conversion (Grabber) and extracting metadata (Metadata Extractors) which is fed to the Data Fusion Module.

The principal aim of the VAM, however, is to track, classify, and recognize articulated and rigid dynamically interacting objects in complex scenes. Referring to the Intelligent University Campus, the VAM has to locate potential users and follow them through the various covered areas. This issue is a well known problem in the Video Processing domain [12]. To achieve this level of analysis, different architectural approaches have been implemented in the last decades within the context of Video Surveillance [13] and are nowadays being ported to the AmI domain [14]. In the context of the presented system, a Metadata Extractor (ME) submodule (Figure 4) is instantiated for each video camera. The reference ME architecture employed for user localization with monocular cameras is described in [15] and briefly reported in the next section.

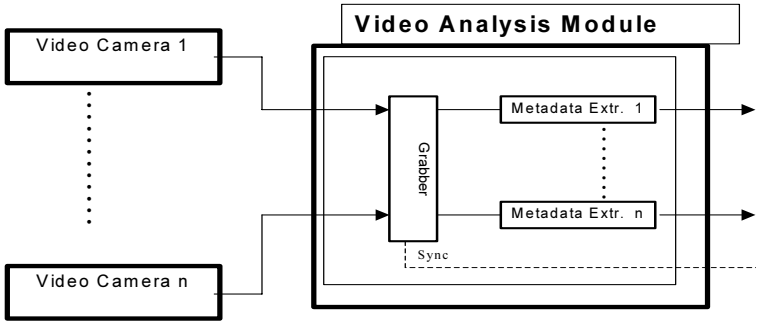



**Figure 3.** The video analysis module is composed of metadata extractors which analyze video streams coming from sensors.

### *3.1. The Metadata Extractor*

Metadata extraction architecture can be described in terms of a logical chain of tasks carried out sequentially. As it can be seen in Figure 4, the first step is a Dynamic Change Detection [16] that extracts the difference between the current image and the reference image (background). In particular, each detected moving area (called Blob) in the scene is bounded by a rectangle to which a numerical label is assigned. Owing to the detection of temporal correspondences between bounding boxes, a graph-based temporal representation (Fig. 5) of the dynamics of the image primitives can be built. The temporal graph provides information on the current bounding boxes and their relations to the boxes detected in the previous frames. Using the temporal graph layer as a scene representation tool, tracking can be performed to preserve temporal coherence between blobs. In addition, to enhance tracking performances in real-time high com-

plexity scenes, a shape based model is used. The method is based on a variation of the Generalized Hough Transform (GHT) [17] proposed by Ballard in 1981, in which corners (high curvature feature points [18]) are used for modeling the shape of the object.

The approach is structured into two steps: learning and detection of the shape of the object. The first step is applied to the objects detected, which are also separated, so as to learn the object shape. When an object is merged with other objects in a successive time instant, the acquired model is used to perform tracking. Additional functionalities are Suspect Trajectory Recognition [12], People Counting [19], and Abandoned Object Identification [12]. The output of the chain is an mpeg-7 stream of metadata describing features of Blobs such as position, ID, class, histogram, etc.

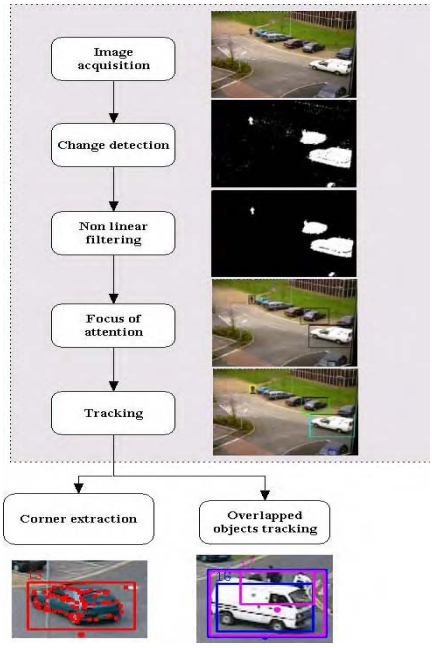



**Figure 4.** Logical chain of modules for metadata extractors.

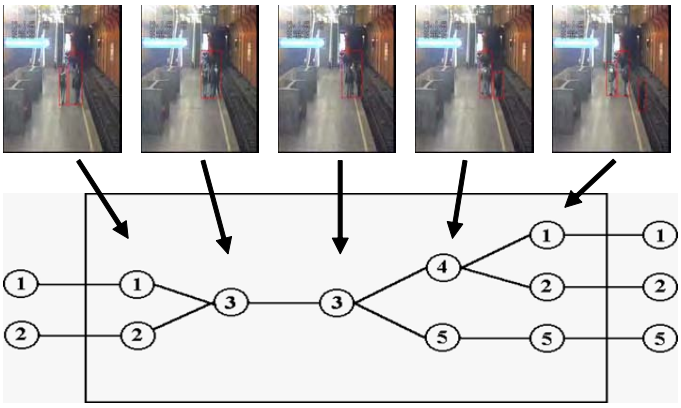



**Figure 5.** Graph based temporal representation.

### *3.2. The Problems of Monocular Approach for Video Analysis*

The monocular systems previously outlined have been shown to be appropriate and performed well in several applications [15]. Nonetheless, they have several native limitations which can be summarized as follows:

a) limited system coverage;
b) 3-D information not available;
c) possible unstable behavior on ID target estimation (i.e.: False Alarms on target detection);
d) low performances in tracking non-rigid occluded targets.

The first problem involves the Video Camera Field of View (FOV) which can monitor in most cases only a subsection of the area of interest, thus losing tracks (Figure 6a). In addition, when a higher image resolution is required, the FOV further restricts its scope.

Issue b) arises when a 3-D model for the blob is required in order to estimate the local surface orientation of a scene detail [20], or to recover the object's initial shape.

Whereas issues a–b) regard the system's extended functionalities, problems c–d) are more closely related to the system's performance in several functioning conditions. In particular, the probability of False Alarm (i.e.: false Blob tracks) in the estimation of a given target dramatically increase when meteorological conditions are unfavorable – shadows and changes in illumination generate noisy artifacts (Figure 6b), leading to misdetection at low levels (i.e.: Change Detection).

The most affecting issue, which in the literature seems to be only partially overcome [21], is the problem of occlusions between Blobs.

An occlusion (Figure 6c) takes place when an object is hidden from the sensor (Video Camera) by a natural or artificial obstacle. There are different types of occlusions. In particular for Intille et al. [22] an object is identified as being occluded if two objects are found to match the same foreground region generated by the Change Detection. Haritaoglu et al. [23] distinguish between *dynamic* and *static* occlusions in relation to the dynamics of the objects generating the occlusion (e.g.: a parked car and a man walking in front of the car generate a static occlusion). In addition, a further classification can be made upon the nature of objects which can be *rigid* (e.g. vehicles, etc.) or *non rigid* (e.g.: pedestrian).

Among all possible kinds of occlusions, those involving non-rigid dynamic objects are the most interesting and difficult to solve. This is due to interactions between persons, as well as to self-occlusions that are present in non-rigid objects. Self-occlusions can be generated for example by human hand, arm and leg movements.

(a) (b) (c)

**Figure 6.** (a) Example of limited Field of View. (b) Shadow artifacts affect blob shapes. (c) Dynamic occlusions reduces the informative content of the image.

### *3.3. The Multisensor Approach*

Adding sensors to a Video Analysis architecture such as the one depicted in Figure 4 is beneficial mainly in that it increases the system's robustness and enhances its functionality, in part covering the principal issues of the monocular approach. There are, however, various drawbacks at the algorithmic and architectural level. In particular, a distributed multisensor architecture is not easy to set-up and maintain (HW/SW). Hence, a set of efficient and possibly real-time techniques to combine multiple instances of the same Blobs have to be developed. A tradeoff must therefore be made between the mono/multi sensor solutions. Within the context of the presented AmI system, the need for a multisensor tracking system is motivated by the fact that the area to be covered by sensors is large (i.e.: University Campus) and a monocular approach is not applicable.

Thus, the possibility of porting techniques developed in the domain of typical data fusion to this field of research is both attractive and innovative.

In Figure 7, radar-scans are displayed (a) along with the map for the parking lot of the Intelligent Campus scenario (b). The relationship between the two cases is clear; in the first, tracks for moving targets in a battlefield have to be estimated, whereas in the second case, Blobs (i.e.: pedestrian or vehicles) have to be located in a more restricted environment. The issue for AmI can therefore be formulated as a Multi-target tracking (MTT) problem. Bar-Shalom et al. define MTT as the process of state estimation of an unknown number of targets. To perform multi-target tracking the observer has at his disposal a large amount of data, possibly collected on multiple sensors [24]. Measurements can be bearings, ranges, delay times, doppler etc., but the main difficulty is in associating a given measurement to a target model. In this way, two problems have to be jointly solved: data association and state estimation. MTT has different application fields, such as: Radar tracking (military applications), Video surveillance systems, and Ambient Intelligence [25,11].

A reference system for Multiple Target Tracking in the video surveillance domain is VSAM (Video Surveillance and Monitoring) [13], a project that began in 1997 under the Defense Advanced Research Projects Agency (DARPA) Information Systems Office. The objective of the VSAM project was to develop automated video understanding technology for use in future urban and battlefield surveillance applications. Technological advances developed through this project enabled a single human operator to monitor activities over a broad area using a distributed network of active video sensors. The sensor platforms are mainly autonomous, notifying the operator only of salient information as it occurs, and engaging the operator minimally to alter platform opera-

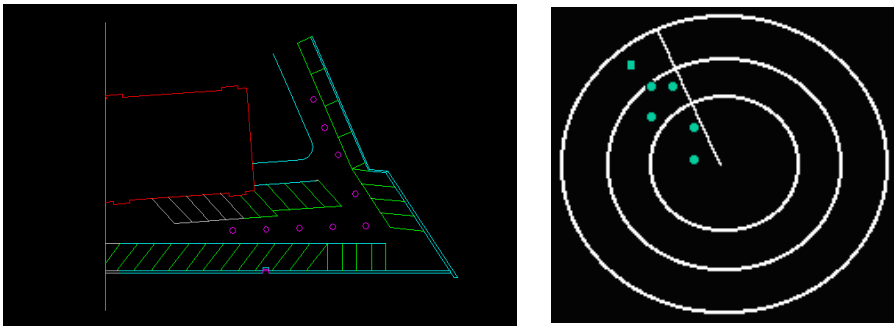


**Figure 7.** (a) Tracks in a parking lot, (b) Radar scans.

tions. Another project exploiting a combination of video and positioning sensors (e.g.: GPS and Video Cameras) for localization is presented in [26].

### *3.4. Balanced and Unbalanced Systems for MTT*

Typical approaches for the implementation of MTT architectures within the context of Video Analysis are the following:

- balanced systems;
- unbalanced systems.

A first distinction between the two systems is based on their use of sensors. In the first case, homogeneous and fixed cameras are used with the same resolution. The outputs of the Video Cameras are processed in parallel and the structure closely follows the Parallel Fusion Network described in [27] (Figure 9).

Consider the system described in [28] as a reference example: eight different cameras are placed in a football stadium in order to track players within the pitch using static cameras with partially overlapping views.

Conversely, Unbalanced Systems are characterized by a combination of static and dynamic (pan-tilt controlled) Video Sensors, which can operate at different resolution levels. Work carried out in [29] is devoted to the acquisition of high resolution video clips containing faces of people walking in a zone of interest. As it can be seen in Figure 8, the system is composed of a widefield static camera (Cam A) which extracts Blobs in low resolution, and a second Dome (i.e.: pan-tilt) Video Camera (Cam B) which focuses on faces and tracks them over time with high resolution. To achieve this, Cam A, given a Blob detected in the wide field frame, controls Cam B by pointing it into the optimal position to start face tracking. Camera B then analyzes the high resolution image and extracts the position of the face within the grabbed frame. Cam B is then repointed using information extracted at low resolution combined with the high resolution analysis.

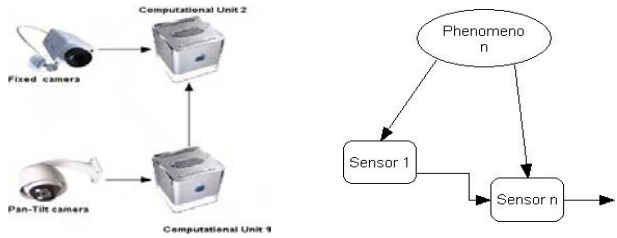



**Figure 8.** Schema for serial fusion network.

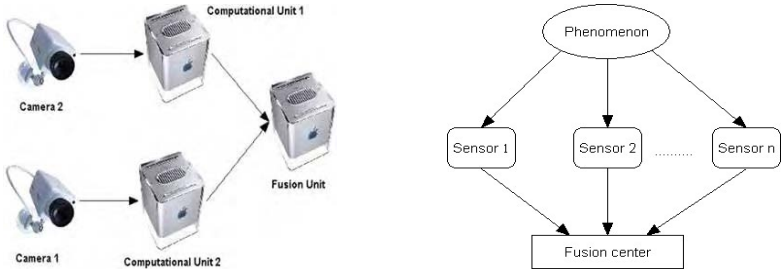



**Figure 9.** Schema for parallel fusion network.

## 4. Data Fusion Module

### *4.1. Introduction*

The Video Analysis Module and the Metadata Agent Manager (MAM) need a common repository where they can successfully combine the collected data. A two-step approach has been adopted here in order to pre-process video data coming from multiple video sensors and then fuse it with metadata coming from MAM.

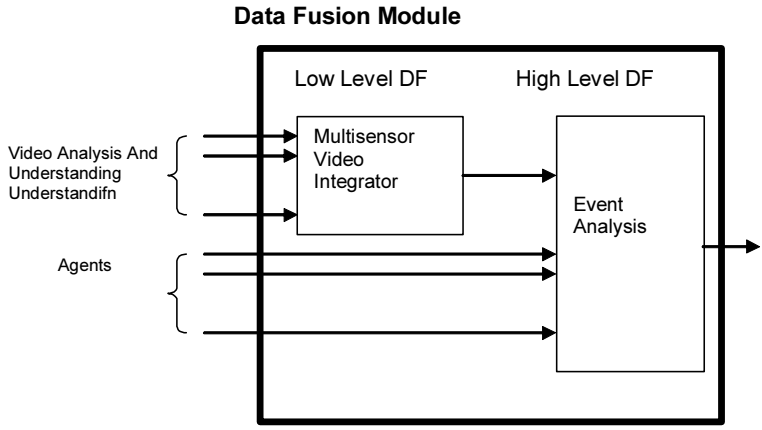



**Figure 10.** Details for the data fusion module.

In Figure 10, the Data Fusion Module (DFM) is reported: a Multisensor Video Integrator (MVI) has been inserted in order to fuse the Video Data (Low Level Data Fusion), whereas an Event Analyzer takes the input metadata coming from agents and from the MVI to generate events with a higher abstraction level.

### *4.2. Low Level Fusion Scheme for Metadata Extraction from Video Data*

Low Level Fusion Analysis (LLFA) is performed in order to combine the output of Metadata Extractors; the final aim is to successfully locate objects of interest and track them over time. The overall scheme for LLFA recalls the general architecture proposed in [30] with a few modifications. Three main steps are performed:

- data alignment;
- data association;
- state estimation.

All three steps are described in detail within the next sections, according to terms and assumption reported hereinafter. It is assumed the architecture is composed of a given set of sensors $S = \{s^j : j = 1,..,N_s\}$; each sensor $s^j$ acquires data and provides *Detection Reports* (DR) $\bar{r}^j_{k,m}$ at time $k$ in relation to objects detected in the monitored environment. In particular, $M^i_k$ represents the total number of reports at frame $k$ extracted by the i-th sensor and aligned both in time and space by the Data

Alignment module. $R_k^i = \{\bar{r}_k^i : i = 1,....,I_k\}$ indicates the set of reports related to the i-th sensor at time *k* whereas $R_k = \{R_k^i : i = 1,...,I_k\}$ incorporates reports produced by all active sensors belonging to *S*.

A DR takes the form of a multidimensional vector composed by different features which describe the object of interest:

$$\bar{r}_{k,m}^i = [\bar{p}_m^i(k), \bar{v}_m^i(k), \bar{c}_m^i(k), l_m^i(k), \bar{h}_m^i(k)]$$

With:

$\bar{p}_m^i(k) = position$

$\bar{v}_m^i(k) = speed$

$\bar{c}_m^i(k) = class$

$l_m^i(k) = id$

$\bar{h}_m^i(k) = histogram$

The first feature in DF is represented by the position of detected objects in terms of Image Coordinates and Map Coordinates components:

$$\bar{p}_m^i(k) = [x_I, y_I, x_M, y_M, z_M]$$

*Speed* is defined in terms of $x_M$ and $y_M$ components:

$$\bar{v}_m^i(k) = [v_{x_M}, v_{y_M}]$$

*Class* is a value that indicates the class that the object belongs to. Possible classes are the following:

1. car;
2. vehicle;
3. pedestrian;
4. other.

*Id* indicates a number that identifies the object univocally in the architecture, whereas the last feature encodes color information of the object in the form of a three dimensional histogram vector:

$$\bar{h}_m^i(k) = [h_R, h_G, h_B]$$

with $h_{R-G-B}$ unidimensional histograms for the Red, Green, and Blue channels. Detection Reports after alignment are selected and associated by DAMs (i.e.: Data Association Modules) to a number *N* of existing tracks $R_{k,n}^t$ which are instantiated for each object present in the scene at time *k*. Tracks are defined as follows:

$$R^{t}_{K,n} = \{\hat{r}^{f}_{k,m_f} : k = 0,..., K\}$$

they are sequences of DF $\hat{r}^{f}_{k,m_f}$ derived from the fusion of reports until time *k*. For each track, the associated estimates for DR coming from the active sensors can be grouped in the set :

$$R^{f}_{k} = \{\hat{r}^{f}_{k,m_f} : m_f = 1,...., M^{f}_{k}\}$$

with $M^{f}_{k}$ total number of fused tracks at time *k*.

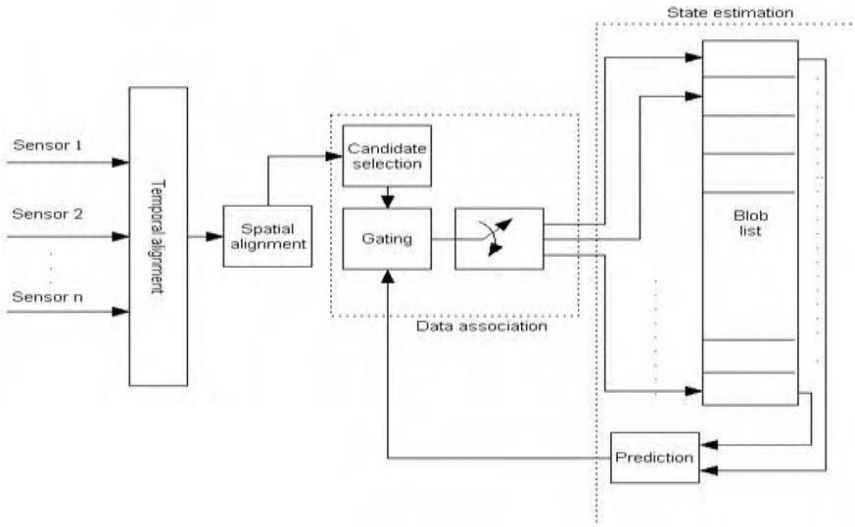



**Figure 11.** Logical architecture for the MVI (multisensor video integrator).

*4.2.1. Spatio Temporal Data Alignment*

The first operation carried out in Figure 11 is a Data Alignment. Given a set of video sensors, the first step to be implemented is a temporal alignment of the acquired data; in the presented case, processes devoted to video data A/D conversion (i.e.: Grabbing Module in Fig. 4) have to be synchronized in different computational units (CU) connected through a 802.11a WLAN.

The problem is reduced to the estimation in each computational unit $c_i$ of the quantity:

$$D(t, c_i) = t_r(t) + t_o(c_i) + t_b(c_i)$$

where :
$c_i$ =*i-th computational unit*
$t_r(t)$ =*drift of CU processor at time t*
$t_o(c_i)$=*offset between* $c_i$ *and* $c_o$
$t_b(c_i)$=*booting time for grabbing process on* $c_i$

A client-server approach based on NTP (i.e.: Network Time Protocol) technology can reduce $t_o(c_i)$ to zero by periodically synchronizing all CUs which grab video frames from cameras to a reference CU (i.e. NTP server). It has been empirically evaluated that for fast processors (e.g. >700 MHz), time drift $t_r(t)$ falls under 0,1 ms with $t$<60 s. Therefore, each client is synchronized with the NTP server at every minute. To ensure that the grabbing processes are executed simultaneously with an acceptable error $t_b(c_i)$ has to be set to zero in all CUs. To accomplish this, each grabber (i.e. Grabbing Process) has been equipped with a UDP client that broadcasts/receives time stamped UDP packets. The following steps are therefore performed at boot time:

1. grabbing processes are started in each CU synchronously and then they wait for UDP packets before grabbing;
2. the UDP server multicasts a packet to all clients;
3. all clients receiving the packet start grabbing processes.

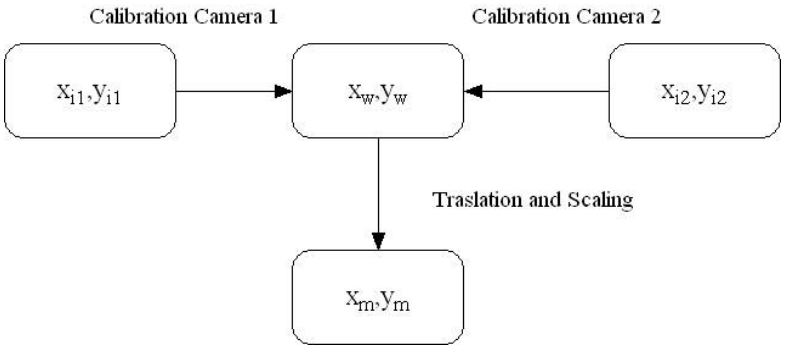



**Figure 12.** Joint calibration strategy.

Considering that the frame rate for a video clip does not exceed 25 fps, an error of 0,1 ms between two CUs ensures synchrony between frames for a time interval of 400 s; after that period, one frame is lost. Higher rates of synchronization ensure better performances.

Spatial alignment is achieved through Camera Calibration. Camera calibration is the process by which optical and geometric features of cameras can be determined. Generally, these features are addressed as intrinsic and extrinsic parameters and they allow for the estimation of a correspondence between coordinates in the Image Plane and in Real Word 3-D space. Calibration of sensors is generally used to deduce 3-D information from 2-D data, but in some cases can be applied to get 2-D image coordinates from 3-D data. Various methods have been proposed to perform calibration: some use non-linear optimisation techniques [31], others use systems of linear equations. The Camera Calibration we use is based on the classic Tsai method [32]. In the presented system, all video sensors have been calibrated with a common calibration strategy. In Figure 12, the chosen approach is outlined. First of all, cameras are calibrated with reference images on a unique map. Then a common reference point has to be found in order for the system to be able to switch between the different reference systems. The World coordinates origin represents a good choice because it is common to all the cameras.

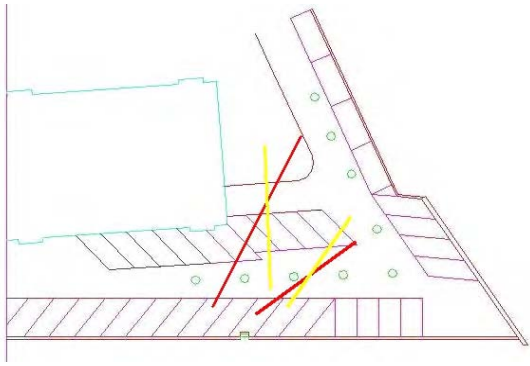
The algorithm can be decomposed in the following steps:

– image coordinates for i-th Cam ($x_{i1}$,$y_{i1}$) are converted to World Coordinates ($x_w$,$y_w$);
– ($x_w$,$y_w$) are converted to a 2-D map ($x_m$,$y_m$) with translation and scaling transformations.

### *4.2.2. Candidate Selection (CS)*

Once objects can be placed synchronously in a common space, a situation such as the one depicted in Figure 13 a–b must be contended with; objects (targets) appear on different parts of the map and have to be associated to existing tracks (i.e. trajectories of objects). To achieve this, fields of view (FOV) of the cameras are plotted on the map using the method described in [15].

(a) (b)



(c)

**Figure 13.** (a) and (b) FOVs for the two cameras; (c) FOVs projected on the map.

The result is visible on Fig. 13b where FOVs for two views (fig. 13a–b) have been projected on the ground plane map. In figure 13c, different parts are highlighted. In particular, a preliminary distinction is made on:

– OFOV: Overlapping Field of View, the region is monitored by more than one sensor;
– NOFOV: Non Overlapping Field of View, only one sensor is active in the region.

The importance of FOV estimation for correct Data Association is evident. Due to the nature of the considered sensors, FOV estimation can be improved with the intro-

duction of an a priori scene model [33] in which different sensors' FOVs are automatically modified in order to overcome problems of low reliability (e.g.: static occlusions etc.). Candidate Selection is then performed to select all DR present in OFOVs.

Given the total number $I_k$ of DR belonging to $R_k$, the Candidate Selection module has to parse this set of reports in relation to the sensors' FOVs, which can be defined with two linear functions in the map plane:

$$\begin{aligned} f^1{}_s &= a^1 x_m + b^1 y_m + c^1 = 0 \\ f^2{}_s &= a^2 x_m + b^2 y_m + c^2 = 0 \end{aligned} \qquad \text{with sensor index } s \in [0..., S].$$

A Candidate DR set can be defined in turn as follows:

$$R^s{}_k = \{\bar{r}^i_{k,m} : f^1{}_s \le \bar{p}^i_{k,m} \le f^2{}_s, \forall s \in [0..., S]\} \quad \text{With } R^s_k \subset R_k .$$

*4.2.3. Data Associations*

Data Association (DA) is the process that associates each DR already parsed by the CS step with an existing/novel trajectory. Relevant features for DA can be found at different levels:

- signal (pixel) level: color histograms;
- object (blob) level: shape, corners, position, other;
- event level: dynamics.

Before Data Association is performed, each feature must be aligned. In particular, histograms have to be normalized in relation to illumination conditions, corners scaled according to dimensions and positions of detected objects in the two cameras, and parameters which regulate corner extraction have to be calibrated in relation to illumination. Given a DR $\bar{r}^i_{k,m}$, the problem of Data Association consists in the selection of the correct track $R^t_{K,n}$ to which the report belongs. In order to perform this step, different association metrics are used; in particular, report features used for association are *speed*, *position* and *color*.

An Association Metric $M_h$ is built for each feature in order to give a binary output representing the match between the observed reports $\bar{r}^i_{k,m}$ and existing tracks $R^t_{K,n}$ with $0<n<N$. The Metric used to test color correspondence between observed DR and tracks is represented by the evaluation of the Bhattacharrya coefficient [34]:

$$M_0(\bar{r}^i_{k,m}, R^t_{K,n}) = \sum_{R,G,B} \sqrt{\bar{h}^i_m(k)\bar{h}^f_n(K)}$$

which measures the distance between the two color histograms $\bar{h}^i_m$ $\bar{h}^f_n$ with respect to R, G and B channels. A high Bhattacharrya coefficient indicates that the object is similar to the given track and that it can be associated to it.

Another metric used within the scope of association is a simple Euclidean metric:

$$M_1(\bar{r}^{i}_{k,m}, R^{t}_{K,n}) = \sqrt{\left(\bar{p}^{i,x}_{m}(k) - \bar{p}^{i,x}_{n}(K)\right)^2 + \left(\bar{p}^{i,y}_{m}(k) - \bar{p}^{i,y}_{n}(K)\right)^2}$$

which gives the absolute distance between center of mass of the DR and the n-th track $R^{t}_{K,n}$ in Image Coordinates. To take into account the dynamics of the DRs, speed vectors are used:

$$M_2(\bar{r}^{i}_{k,m}, R^{t}_{K,n}) = \sqrt{\left(\bar{v}^{i,x}_{m}(k) - \bar{v}^{i,x}_{n}(K)\right)^2 + \left(\bar{v}^{i,y}_{m}(k) - \bar{v}^{i,y}_{n}(K)\right)^2}$$

they turn out to be good features when faced with situations such as the one sketched in Fig. 6: two people have relative small distance but colliding trajectories, therefore in this case, a position based metric will fail whereas a metric based on speed can lead to the correct result. Once all metrics are evaluated, a threshold step is performed in order to extract binary results from the matching procedure:

$$o_n(M_h(\bar{r}^{i}_{k,m}, R^{t}_{K,n})) = \begin{cases} 1 & if\,(M_h(\bar{r}^{i}_{k,m}, R^{t}_{K,n})) \le th_h \\ 0 & if\,(M_h(\bar{r}^{i}_{k,m}, R^{t}_{K,n})) > th_h \end{cases}$$

The threshold step is repeated for all metrics and results are stored in a Decision Vector:

$$\bar{O}_{m_f} = [o_1, ...., o_H]$$

Association Rules are therefore applied to the Decision vector in order to establish whether or not the DR belongs to the given track. A set *A* of Association Rules (AR) is therefore defined:

$$A = \{a_j : j = 0, ....., j\}$$

Commonly used AR are the MAJ rule, and the AND /OR rule [27]. The final output of the Association Module takes the form of a subset of DR which are associated to a single track:

$$R^{A}_{k,m_f} = \left\{ \bar{r}^{i}_{k,m} \in \bar{R}_{k,m_f} : a_j(O_{m_f}) = true \right\}$$

with $R^{A}_{k,m_f} \subset R_k$.

*4.2.4. State Estimation*

The problem of state estimation arises when for a given track $R^{t}_{K,n}$ a set of associated DR is evaluated. Features of the track have to be updated taking into account the new

associated observations. The principal feature to be estimated is nonetheless the position of each DR in map coordinates. Therefore, given the set of available DRs associated with the track, a relatively simple approach has been exploited for determining the position of center of mass of the track in a condition of *non-occlusion*:

$$\bar{p}_n^{i,x}(k) = \frac{1}{M}\sum_m \bar{p}_m^{i,x}(k) \quad , \quad \bar{p}_n^{i,y}(k) = \frac{1}{M}\sum_m \bar{p}_m^{i,y}(k)$$

In the condition of *occlusion* of a DR, a more complex method [35] based on the Generalized Hough Transform (GHT) is used. The GHT is a technique used to search for arbitrary curves in an image without the need for parametric equations. A look-up table called R-table is used to model the template shape of the object. This R-table is used as a transform mechanism.

To build the R-Table, first a reference point and several feature points of the shape are selected. For each feature point the orientation $\alpha$ of the tangential line at that point, the length ***r***, and the orientation $\theta$ of the radial vector that joins the reference point and the feature point can be calculated. If ***n*** is the number of feature points, a 2 x ***n*** indexed table can be created using all ***n*** pairs ($r, \theta$) and using $\alpha$ as an index. This table is the model of the shape and it can be used with a transformation to find occurrences of the same object in other images. The shape is localized by using a voting technique. The high curvature points (e.g.: corners [18]) of each blob detected in the image are extracted, and for every point, the orientation $\alpha$ is computed. Using $\alpha$ as an index for the R-table, the pair ($r, \theta$) is extracted. Using the pair ($r, \theta$), the possible position for the reference point can be computed and an accumulator of its position is incremented. Although some points that do not belong to the desired shape will have similar $\alpha$ and will introduce false reference points, the maximum accumulator value will occur with high probability at the actual reference point. In our approach, a computationally simpler variation than the GHT is used in order to automatically extract the model of the object (R-table) and also to evaluate the position of the object (voting). Corners extracted from the object are used as feature points and a different parameterization is used. employed. The pairs (***dx***, ***dy***) are used instead of ($r, \theta$), (***dx*** and ***dy*** are the differences in ***x*** and ***y*** with respect to the reference point). The presented GHT variation is shown in Figure 14.

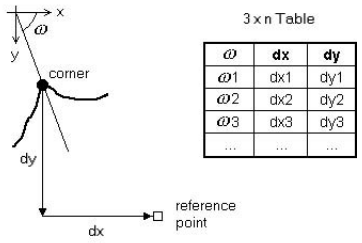



**Figure 14.** R-table structure.

Once voting spaces have been calculated, the fused DR position in terms of the center of mass can be evaluated by setting it to the value that received the highest number of votes in the Hough space.

### *4.3. High Level Fusion Scheme for Scene Interpretation*

The primary objective of the High Level Fusion scheme (HLFA) [36] is to determine the current event taking place in the region of interest by combining heterogeneous data collected by Agents and fused metadata information regarding DRs. The events that HLFA has to discriminate mainly regard the activity of people detected in the monitored area. In particular, seven different events have been taken into account: four of them describe the state of activity of people (i.e.: WHL = high human work, WHF = low human work, WAL = low automatic work and WAF = high automatic work) the other three describe a more general condition (i.e.: ARRIVE = arrival of new person, NULL = system not active and WAIT = nothing relevant). The approach used to implement the HLFA is neural-based, in particular, Self-Organizing Maps (SOM) [37] have been exploited; they perform a spatial organization process of the input features, called *Feature Mapping,* through an unsupervised learning technique (input and output are not mapped by an external supervisor). An input vector $x$ is compared with the parametric reference vector $m_i$ and the best mach is defined as a "response." The input is mapped onto the correct location. To define the *winner* node $c$ to which $x$ belongs, the smallest of the Euclidean distances $\| x - m_i \|$ is evaluated:

$$\|x - m_c\| = \min\left(\|x - m_i\|\right)$$

In our work, all operations can be divided into six steps according to [11]:

1. map initialisation: reference vectors are initialised to random values bounded by the minimum and the maximum of the learning set;
2. map training: a first phase called *ordering phase,* where the reference vectors of the map unit are ordered, and a second phase, in which a fine-tuning is performed.
3. evaluation of quantization error using different feature vectors;

$$err = \|x - m_c\|$$

4. map calibration: map units are calibrated using known input data samples defined by a label;
5. map visualization;
6. real-time work.

The feature vector, $x_{FV}$, used in the previous steps, is defined as below:

$$x_{FV} = [x_P, x_{AP}, x_{PC}]$$

where $x_P, x_{AP}$ and $x_{PC}$ include the output of the sensors processed based on the following considerations.

#### *4.3.1. People Distribution*

People distribution is a feature extracted from the set of fused DR $R^A_{k,m_f} \subset R_k$ and it is evaluated by subdividing the monitored region into g parts (i.e.: with $1 < g \leq G$, G =

total number of regions). Then for each region the number of people in the region is evaluated by exploring $R^{A}_{k,m_f}$.

The result is a feature vector $x_P$ composed by integer numbers, $f_g$, indicating the number n of people present in zone g, weighted by the time of presence in that zone kg:

$$f_g = k_g n_g$$

with $x_P$ defined as follows:

$$x_P = [f_0^1, f_1^1, ..., f_G^1]$$

*4.3.2. Authorized People*

This feature measures the number of authorized people entering and exiting the guarded area as estimated by an electronic badge reader. It should be noted that the sensor is able to send the unique identifier of the person that is trespassing to the associated door, but this additional information is not used by the system.

The feature is composed of two integer numbers: $x_{LI}$ indicating the number of badges read, $x_{\Delta LI}$ indicating the difference between the number of badges read in the previous feature and in the current one, respectively.

$$x_{AP} = [x_{LI}, x_{\Delta LI}]$$

*4.3.3. Logins, PC Usage and Network Load*

The number of people that, upon entering the monitored environment use its computational resources is monitored through dedicated Agents [38] which record all logins into the available PCs. In addition, another set of Agents monitors the network and the processor load for each PC. Two of these are integer numbers, $x_{AL}$ indicates active logins and $x_{\Delta AL}$ indicates the difference between the number of active logins in the previous time frame and in the current one. This feature also contains two float numbers $x^i_{NL}, x^i_{CPU,L}$ to indicate the network load and the percentage of processor usage. Two other float numbers are used for the global network loading $x_{LAN,L}, x_{ON,OFF}$.

$$x_{PC} = [x_{AL}, x_{\Delta AL}, x^i_{NL}, x^i_{CPU,L}, x_{LAN,L}, x_{ON,OFF}]$$

*4.3.4. SOM Training and Calibration Procedures*

The training session was performed by collecting 3769 feature vectors in an offline mode. Data collection was performed by recording information from sensors for a total period of ten hours; this period was divided into three subsets to get a sequence in early morning, midday and evening. This procedure was performed in order to present all

meaningful situations to the AmI system. The whole sequence was reiterated to have 5,000 inputs in the ordering phase and 200,000 for the fine-tuned phase.

In the map calibration, 680 feature vectors derived from some particular situations were collected. In this case only 420 were manually selected and presented to the system. The result of the training and calibration is the map layout shown in Figure 15.

**Figure 15.** Self organizing map layout in which different neuron clusters are highlighted.

## 5. Results

Results shown here mainly regard High Level and Low Level Fusion Schemes. In particular, LLFA has been tested using sequences grabbed from two outdoor video sensors with partially overlapping fields of view. Figure 16 (a, b) reports two people walking in occluding directions in a parking lot; in (c) tracks can be seen belonging to the four non-fused DRs whereas in (d) the result of fusion is provided. It can be noticed how the occluding event has been resolved and tracks are clean with preserved *id*. More qualitative results are available in [39].

The results shown for the HLFA are derived from on-line tests carried out in real conditions.

A ground-truth data set was designed in order to quantitatively test the goodness of the trained map. In particular, plots were written annotating a sequence of actions performed by an actor and the time of the event. Results are summarized in table 1, where a percentage of false alarms $P_{FA}$ and correct decisions $P_D$ is reported for the recognition of the correct event.

**Table 1.** False alarm rate ($P_{\mathrm{FA}}$) and correct detection rate ($P_{\mathrm{D}}$) are shown for each Super State

| super-state | real world | $P_{\mathrm{D}}$ | $P_{\mathrm{FA}}$ |
|---|---|---|---|
| WHF | low human work | 60% | 40% |
| WHL | high human work | 60% | 40% |
| WAF | low machine work | 80% | 30% |
| WAL | high machine work | 80% | 30% |
| ARRIVE | laboratory incomes | 72% | 50% |
| WAIT | Sleeping | 70% | 35% |
| NULL | everything stopped | 98% | 1% |

## 6. Conclusions and Future Work

A complete system for Ambient Intelligence applications characterized by a high level of integration of heterogeneous techniques has been described both at the architectural and the algorithmic level. Particular emphasis has been given to Multisensor Issues on the localization of users and event recognition. The high level of innovation of the combination of video data with contextual data coming from heterogeneous sensors makes possible the high level representation of a given scene of interest with acceptable accuracy.

Future work will mainly concentrate on an enhanced multimode Action and Communication Cluster for superior interaction with users.

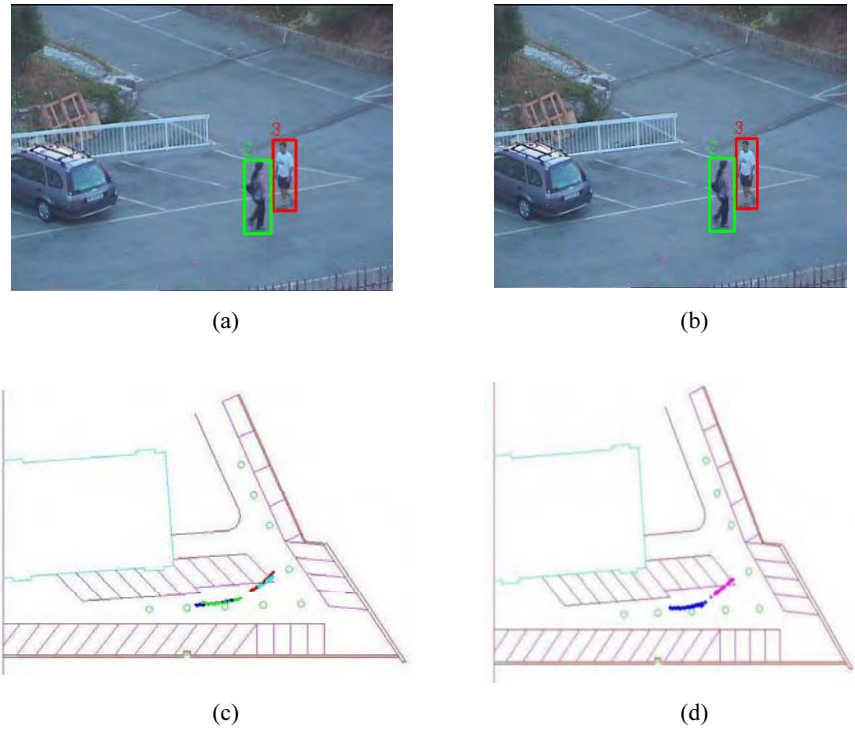


**Figure 16.** (a,b) The two OFOVs (c) non fused trajectories (d) fused trajectories.

## Acknowledgements

This work was partially supported by the University and Scientific Research Ministry (MIUR) of the Italian Government under the National Interest Scientific Research Program and by ELSAG spa.

## References

[1] Mark Weiser, "The Computer for the Twenty-First Century," Scientific American, pp. 94–10, Sept. 1991.

[2] ISTAG; Scenarios for Ambient Intelligence in 2010; Final Report, Feb. 2001, EC 2001: http://www.cordis.lu/ist/istag.htm.
[3] Brooks, R.A. with contributions from M. Coen, D. Dang, J. DeBonet, J. Kramer, T. Lozano-Perez, J. Mellor, P. Pook, C. Stauffer, L. Stein, M. Torrance and M. Wessler, "The Intelligent Room Project". Proceedings of the Second International Cognitive Technology Conference (CT'97), Aizu, Japan, August 1997.
[4] Starner, T. "Human-powered wearable computing", IBM Systems Journal, 35(3 & 4), 618–629, 1996.
[5] ISTAG; Scenarios for Ambient Intelligence in 2010; Final Report, Feb. 2001, EC 2001: http://www.cordis.lu/ist/istag.htm.
[6] http://www.vicom-project.it/.
[7] http://ginevra.dibe.unige.it/ISIP/Projects/miur02/main.html.
[8] http://www-2.cs.cmu.edu/~aura/.
[9] http://oxygen.lcs.mit.edu/.
[10] http://charlotte.acns.nwu.edu/gretchen/design/smart.htm.
[11] Marchesotti, Gandetto, Negroni, Sciutto, Regazzoni, "From multisensor surveillance towards smart interactive spaces", Baltimora, ICME 03, 2003.
[12] E. Stringa, C. Sacchi, C.S. Regazzoni, "A multimedia system for surveillance of unattended railway stations", European Signal Processing Conference, Eusipco 1998, Rhodes, Greece, 1998, pp. 1709–1712.
[13] http://www-2.cs.cmu.edu/~vsam/.
[14] http://www-prima.imag.fr/.
[15] L. Marcenaro, F. Oberti, C.S. Regazzoni, "Multiple objects color-based tracking using Multiple-cameras in complex time-varying outdoor scenes", 2nd IEEE Int. Workshop on Performance evaluation of tracking and surveillance, Kauai, Hawaii, USA, Dec. 2001.
[16] L. Marcenaro, G. Gera and C.S. Regazzoni, "Adaptive Change Detection Approach for Object Detection in Outdoor Scenes Under Variable Speed Illumination Changes", European Signal Processing Conference, Eusipco 2000, Tampere, Finland.
[17] M.J. Swain and D.H. Ballard "Color indexing. International Journal of Computer Vision", Vol. 7 (1): pag. 11–32, 1991.
[18] S.M. Smith and J.M. Brady, "SUSAN – a new approach to low level image processing", Int. Journal of Computer Vision, Vol. 23(1), pag. 45–78, May 1997.
[19] M. Peri, C.S. Regazzoni, A. Tesei, G. Vernazza, "Crowding Estimation in Underground Stations: a Bayesian Probabilistic Approach", ESPRIT Workshop on Data fusion, 1993.
[20] F. Pedersini, A. Sarti, S. Tubaro "Automatic monitoring and 3D reconstruction applied to cultural heritage", Journal of Cultural Heritage, Elsevier Science, Vol. 1, N. 3, pp. 301–313, 2000, L. Marcenaro, F. Oberti, C.S. Regazzoni, "Multiple objects color-based tracking using Multiple-cameras in complex time-varying outdoor scenes", 2nd IEEE Int. Workshop on Performance evaluation of tracking and surveillance, Kauai, Hawaii, USA, Dec. 2001.
[21] J. Black and T. Ellis, "Multicamera Image Tracking", 2nd IEEE Int. Workshop on Performance evaluation of tracking and surveillance, Kauai, Hawaii, USA, Dec. 2001.
[22] Intille, Davis, Bobick, "Real time closed-word tracking", Proc CVPR'97, 1997.
[23] Haritaoglu, Harwood, Davis, "W4: real time surveillance of people and their activities" IEEE PAMI, 22(8): 809–830, 2000.
[24] Bar-Shalom, Y., Fortmann, T.E. (1988) "Tracking and data association" IEEE Trans. On Aerospace and Elect. Systems, vol. 28 no. 2, pp. 560–565.
[25] Hans-W. Gellersen and Michael Beigl, "Ambient Telepresence: Colleague Awareness in Smart Environments", Telecooperation Office (TecO), University of Karlsruhe, Germany.
[26] B. Collins, Lipton, Fujioshi Kanade, "A system for video surveillance and monitoring", Proceedings IEEE Vol. 89, pag. 1456–1477, Oct. 2001.
[27] P.K. Varshney, "Distributed Detection and Data Fusion," N.Y., Springer-Verlag, 1997.
[28] http://inmove.erve.vtt.fi/.
[29] Luca Marchesotti, Alessandro Messina, Lucio Marcenaro and Carlo Regazzoni, "A cooperative multisensor system for face detection in Video Surveillance applications" Int. Journal of Chinese Automation, 2002.
[30] James Llinas (Preface), Edward L. Waltz, Franklin E., Jr. White, "Multisensor Data Fusion", Artech House Radar/Electronic Warfare Library.
[31] J. Renno, J. Orwell, G.A. Jones."Towards Plug-and-Play Visual Surveillance: Learning Tracking Models", International Conference on Image Processing, Sept Rochester, New York, 2002.
[32] Roger Y. Tsai. "A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses". IEEE Journal of Robotics and Automation, RA-3(4), pag. 323–344, August 1987.

[33] T. Ellis and M. Xu, "Object Detection and Tracking in an Open and Dynamic World", 2nd IEEE Int. Workshop on Performance evaluation of tracking and surveillance, Kauai, Hawaii, USA, Dec. 2001.
[34] D. Comamciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using Mean Shift", IEEE Conference Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, volume II, pp. 142–149, June 2000.
[35] F. Oberti, L. Marcenaro, C.S. Regazzoni, "Real-time change detection methods for video-surveillance systems with mobile camera", XI European Signal Processing Conference Sept, Toulouse, France, 2002. J. Black and T. Ellis, "Multicamera Image Tracking", 2nd IEEE Int. Workshop on Performance evaluation of tracking and surveillance, Kauai, Hawaii, USA, Dec. 2001.
[36] L. Marchesotti, M. Gandetto, D. Negroni and C.S. Regazzoni, "From Multisensor Surveillance Towards Smart Interactive Spaces", International Conference and Multimedia Expo, Baltimora, 2–9 july, 2003.
[37] T. Kohonen, "The Self-Organizing Map", Proceedings Of the IEEE, Vol. 78, N° 9, pp. 1464–1480, 1990.
[38] L. Marchesotti, L. Marcenaro, C.S. Regazzoni, "Heterogeneous data collection and representation within a distributed smart space architecture", Advanced Concepts for Intelligent Vision Systems, Ghent University, Belgium, Sept 2002.
[39] http://ginevra.dibe.unige.it/ISIP/sequencesLuca.html.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# An Expert System for Surveillance Picture Understanding

Helman STERN, Uri KARTOUN and Armin SHMILOVICI
*Faculty of Engineering, Ben-Gurion University, P.O. Box 653,*
*Be'er-Sheeva 84105, Israel, Fax: +972-8-6472958; Tel: +972-8-6477550*
*E-mail: (helman, kartoun, armin)@bgumail.bgu.ac.il*

**Abstract.** The last stage of any type of automatic surveillance system is the interpretation of information acquired from sensors. This work focuses on the interpretation of motion pictures taken from a surveillance camera, i.e., image understanding. The expert system presented in this paper can describe simple human activity in the field of view of a surveillance camera in natural language. The system has three different components: a pre-processing module for image segmentation and feature extraction, an object identification expert system (static model), and an action identification expert system (dynamic temporal model). The system was tested on a video segment of a pedestrian passageway taken by surveillance camera.

**Keywords.** Image understanding, picture segmentation, fuzzy expert systems, surveillance video

## 1. Introduction

With the continuous decline in the price of imaging technology, there has been a surge in the use of automatic surveillance systems and closed circuit TV (CCTV). Banks, ATM machines, schools, hospitals, and transport walkways employ automatic video recording of their surrounding environments. There appears to be little human inspection (in real-time or otherwise) of these surveillance videos, and thus the system is relegated to a simple deterrence function (mainly to deter possible felonies). However, in many environments it is necessary to understand the contents of the video, for subsequent event detection, storage and retrieval. Extraction of the desired events requires a high semantic level of human understanding and a prohibitive amount of human processing.

The automatic processing of surveillance videos introduces several practical problems. Recording, storing and managing large volumes of data is expensive, and cost effective solutions are not yet available for cases where most of the data is useless. Also, in the case that someone would want to inspect the contents of the video, there would be a great deal of work involved in watching all the recorded segments; hence there is imperative need for efficient automated retrieval of desirable information.

These problems did not escape the surveillance industry: saving of storage space is introduced by time-laps recording (e.g. a frame a second), and by motion-activated recording. While both of these methods are easy to implement, they do not solve the problem of storing the images by a meaningful key that will later facilitate human re-

trieval of important information (e.g. the face of a felon). Also, the activation sensor might be sensitive to spurious events (e.g. the passage of a pet), which are considered unimportant. The real need is for an effective means by which the content of the data can be automatically characterized, organized, indexed, and retrieved, doing away with the slow, labor-intensive manual search task. Understanding the contents of an image, in the context of its importance to the operator of the surveillance system, is the key to the efficient storage and retrieval of video segments.

The problem of automatic image understanding is a difficult one. The problems of modeling and understanding visual behaviors and their semantics are often regarded as computationally ill-defined [1]. In simple terms, the meaning of a behavior (such as hand-waving) is context dependent. Figuring out the context from the image could be more difficult than identifying the behavior, or the context may not depend on visual information at all.

There are two possible solution paradigms that may be used for that problem [2,3]: the computational feature-based semantic analysis – the detection of features based on elaborate computational models, and the human cognitive perception of high level semantics [4], i.e., the subjective interpretation of the user based on some features in the image. Both approaches, low level feature detection, and interpretation semantics [5–7], can be integrated.

With the computational paradigm, it is technically difficult to identify correctly and in a reasonable amount of time, the contents of an image in all possible circumstances (e.g. identify an object from all possible angles). It would be necessary to develop a model for the features of an image in all possible circumstances.

Human cognitive perception, on the other hand, starts with simple object segmentation, that is to segment projected 2D-plane images (generated from a video sequence) of an arbitrary 3D scene into physically meaningful objects. Image segmentation is potentially simpler than feature identification. Image understanding involves the relations between the objects in the picture and the context in which they appear (e.g. when an object is of importance) and that, in general, is very difficult to formulate. Yet, people, even children, can learn to do it with ease.

The problem of image understanding can be facilitated [8,9] if we restrict the type of objects to be identified, (e.g. humans), the quality of the identification (e.g. contours only), the possible relations between objects (e.g. approaching each other), and the context in which they operate (e.g., a closed passageway).

In this work, a prototype of a fuzzy expert system is presented which can describe, in natural language, simple human activity in the field of view of a surveillance camera. The system has three different components: a pre-processing module for image segmentation and feature extraction, an object identification fuzzy expert system (static model), and an action identification fuzzy expert system (dynamic temporal model). The system was tested on a video segment of a pedestrian passageway taken by a surveillance camera.

The rest of this paper is organized as follows: Section 2 describes the surveillance problem and its simplification. Section 3 describes the construction of the fuzzy expert system. Section 4 explains the static and dynamic fuzzy expert systems for object identification and object behavior, respectively. Section 5 provides the results of applying the system to a pedestrian passageway. Section 6 concludes the paper with a discussion.

## 2. Problem Definition

Figure 1 presents a typical scene that might be observed under a surveillance system. This scene is a semi-covered passageway between two buildings at Ben-Gurion University. The open door on the lower left leads to a cafeteria, while there is a lecture hall on the right side with large windows. The sunlight enters through the right side of the scene during daytime, so the lighting conditions may change drastically throughout the course of the day, causing variable length shadows to be present in this scene. It was decided to use a single camera whose visual axis points down into the scene. This affords the use of prior knowledge regarding a minimalist modeling for future real-time processing and to exploit the complementary qualities of different visual clues, such as the relative sizes and motion of objects.

In the scene below, people are the objects of interest, though occasionally other objects appear, such as, an electric delivery cart to the cafeteria and birds. It would be preferable to describe the activities in this scene in terms of human activity.

**Figure 1.** A typical scene taken from a surveillance camera.

The full range of possible human activities that can be described in natural language is very large indeed. Fortunately, in the context of surveillance, interest is mainly focused on a small subset of these human activities. In this project, *abnormal* incidents were defined as {parcel-near-cafeteria, a running person, a person outside working hours}. Indications of abnormal behavior are, for example, the deployment of a parcel bomb, trespassing, violence and theft. The examples outline the identification of normal gross human behavior. In the context of the scene above, people can either be standing or walking; they can be alone, or in groups of two or more. In either case, the position of the person or group is described relative to the scene.

As it turns out, it is still technically difficult to identify the concept of a “person” in a noisy environment such as that described above, especially with artifacts due to changing lighting conditions, shadows, optical distortions, reflections on large glass windows, and the variability of human motions. Also, there is a limit on the reasonable computational time needed to generate a description. Thus, further simplifications to the problem were made [10,11]:

a) a primitive notion of a "blob" is defined as a set of clustered pixels that have moved between two given images. The blobs in each image were segmented from the background of the image. A blob may or may not be a real object in the scene such as a person. It may instead be background noise that was not removed during the image segmentation operations. This simplification, however, facilitated the necessary image pre-processing operations;
b) a fuzzy inferencing mechanism is used for perceptual integration of simple visual clues. The theory of fuzzy sets, which can use a linguistic variable such as, for example, distance, is employed. This linguistic variable can take on various terms such as: very close, close, far, etc. This replaces the "crisp" mathematical description of distance, such as: 4.15 meters. This facilitated the use of mathematical models that capture and describe activities in the scene in natural language.

The goal of this project was defined as follows: to develop a prototype of a fuzzy expert system that can understand and describe a scene in natural language. Fuzzy rules are based on domain expert knowledge and are used to describe a scene, locations of objects of interests, object descriptions and object behaviors.

Given a set of scenes (images) from a video clip {i.e. a set of images I(t), I(t–1), I(t–2), etc. taken at consecutive times}, describe the scene in terms of the number of people and groups of people in the scene, and actions such as walking toward or away from the camera, standing still, departing from another person, walking with another person, joining a group, etc.

## 3. Fuzzy Expert Systems

Fuzzy set theory [12] and fuzzy expert systems [13] are used to capture expert knowledge that cannot be easily formulated mathematically, or when the mathematics is too complex to solve. In building an expert system, domain experts must first be interrogated (in this case, image processing experts and surveillance staff) and their knowledge must be formulated in the form of linguistic variables and fuzzy rules [12]. Additional domain knowledge can be included. Also, some knowledge can be derived from statistical analysis of historical information. References [14–16] present applications of fuzzy logic to image processing.

The fuzzy system considered in this paper is comprised of four basic elements [12]: a fuzzifier, a fuzzy rule base, a fuzzy inference engine, and a defuzzifier. We consider multi-input single-output fuzzy systems as elaborate mapping functions:

$f$: U $\subset$ Rn $\rightarrow$ V $\subset$ R, where $U = U_1 \times U_2 \times \ldots \times U_n \subset R^n$ is the input space and V $\subset$ R is the output space. A multi-output system can be represented as a group of single-output systems.

A rule is a proposition that implies another proposition. In this paper, the fuzzy rule base consists of a set of linguistic rules in the form of "IF a set of conditions is satisfied THEN a set of consequences is inferred." Assume that there are N rules of the following form:

Ri: IF x1 is Ai1 and x2 is Ai2 and...and xn is Ain THEN y is Ci, i=1,2,...,N

where xj (j=1,2,...,n) are the input variables to the fuzzy system, y is the output variable of the fuzzy system, and the fuzzy sets Aij in Uj and Cj are linguistic terms character-

ized by fuzzy membership functions Aij(xj) and Ci(y), respectively. Each rule Ri can be viewed as a fuzzy implication (relation) Ai = Ai1xAi2x...xAin $\rightarrow$ Ci, which is a fuzzy set in $U \times V = U_1 \times U_2 \times \ldots \times U_n \times V$ with membership function Ri($\overline{x}$,y) = Ai1(x1) * Ai2(x2) * ... * Ain(xn) * Ci(y), and* is the T norm [12], $\overline{x}$ =(x1,x2,...,xn) $\in$ U and y $\in$ V.

## 4. The Static and Dynamic Fuzzy Expert Systems Models

In this section a static and a dynamic (temporal) expert systems are described. Initially, however, a brief discussion of the pre-processing stage is given although this is not the main focus of the paper. In the pre-processing stage, a raw image is pre-processed for the identification of blobs. The main function of the static expert system is for object identification. It uses the geometrical attributes of the blobs to make inferences about the objects in the picture. The main function of the dynamic expert system is for action identification. It uses temporal movement attributes of objects to make inferences about the behaviors of the objects in the picture.

### *4.1. Image Pre-Processing*

The pre-processing stage starts with a gray-scale image of a scene. Various image-processing operations are used to remove noise from the image due to optical distortions of the lens and adapt to ambient and external lighting conditions, and shareholding to segment out blobs from the background. The end result is an image of segmented blobs from which features are extracted. Using the Image Processing Toolbox of MATLAB [17], twelve different geometrical attributes were defined for each blob in the image:

- *area*: the actual number of pixels in a blob;
- *convex area:* the number of pixels in the convex area of a blob;
- *solidity*: the ratio between the above two area measures;
- the *equivalent diameter* of a circle with same area;
- *centroid*: the coordinates of the center of gravity of the blob;
- the coordinates of the *Bounding Box*;
- the *minor axis length*, *major axis length*, and *eccentricity* for a bounding ellipsoid;
- The *orientation* (in degrees) to the horizon;
- *extent*: the proportion of the pixels in the bounding box that are also in the region. Computed as the Area divided by the area of the bounding box;

Figure 2 presents a segmentation of the scene shown in figure 1. Table 3 presents further details of the segmentation.

Different features are associated with each blob. Static features, such as blob centroid and size, will be used to classify each blob in the scene into separate categories such as: one person; two people; more than two people or a noise blob. These are discussed further in the section on the static model. Dynamic features, such as direction of movement relative to the camera, will be used to classify the activities of each blob in the scene into categories such as: blob moving toward camera, blob moving away from

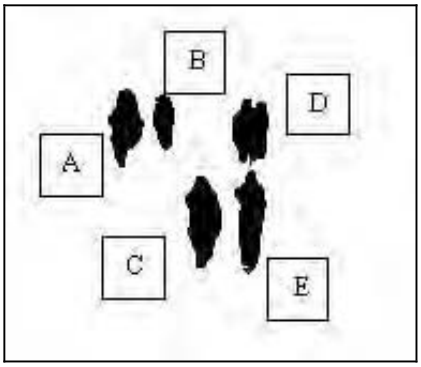



**Figure 2.** Segmentation of Figure 1 into blobs.

camera, blob is stationary. Other possible descriptions, not investigated here, result from the relational positions between different blobs, such as: a blob has merged with another blob; a blob has split into two blobs; a close pair of blobs walk together; two blobs meet and stop; two blobs depart from each other and move in different directions, etc. These are discussed further in the section on the dynamic model.

### *4.2. The Static Expert System Model*

For the object identification stage, a fuzzy expert system was built, with three input variables and one output (conclusion) variable. Linguistic variables were defined for each input.

- the area (in pixels) of the blob, defined on the range [0, 3000] pixels, can take on five terms: Area = {very-small, small, normal, large, very-large};
- the aspect ratio of the bounding box (the height/width ratio of the smallest rectangle that can contain a blob), defined on the range [0, 3.5], can take on five terms: Ratio = {very-small, small, normal, large, very-large};
- the y-coordinates of the center of mass of a blob (a simple estimator for its distance from the camera), defined on the range [0, 250] pixels, can take on four terms: Distance = {very-far, far, close, very-close};
- the conclusion about the identity of the blob, defined on the range [0, 30], can take on five terms: Conclusion = {not-a-person, single, couple, three, many}.

Figures 3(a), 3(b) present the fuzzy sets defined for two of the linguistic variables. These were selected to be Gaussian membership functions from MATLAB's fuzzy logic toolbox [18] after some experimentation with the possible ranges. Table 1 presents the rule-base used for blob identification. The seemingly simple rules are logically derived from prior knowledge of the relative position of the camera and the scene. Also, the rules utilize implicit relational knowledge about the image, such as: "a far object has small area," or "groups have larger area than single person." While it is possible to formulate mathematical functions for these notions, they most likely will be complex because of the stochastic nature of the relationships.

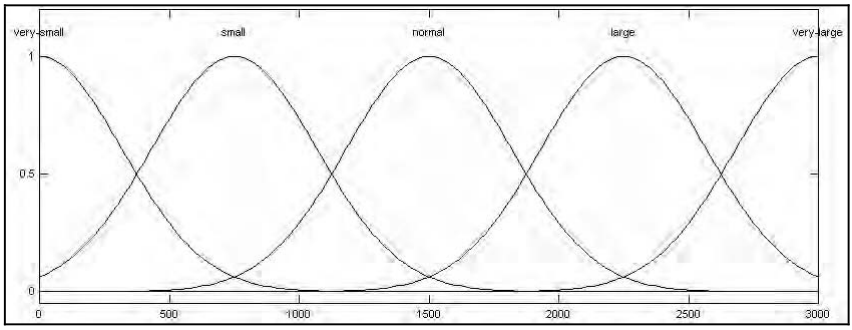



**Figure 3(a).** Membership input function for size of a blob.

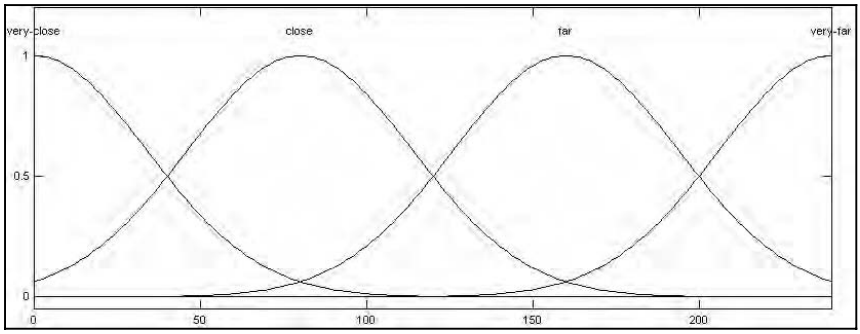



**Figure 3(b).** Membership input function for distance from camera of a blob.

**Table 1.** Fuzzy rules for blob identification

| premise | | | | | | conclusion | |
|---|---|---|---|---|---|---|---|
| if | | area = *very-small* | | | | then | *not-a-person* |
| if distance = *close* | and | area = *small* | and | ratio = *normal* | | then | *single-person* |
| if distance = *close* | and | area = *small* | and | ratio = *very-large* | | then | *single-person* |
| if distance = *close* | and | area = *normal* | and | ratio = *large* | | then | *single-person* |
| if distance = *close* | and | area = *normal* | and | ratio = *very-large* | | then | *single-person* |
| if distance = *close* | and | area = *large* | and | ratio = *normal* | | then | *single-person* |
| if distance = *far* | and | area = *very-small* | and | ratio = *very-large* | | then | *single-person* |
| if distance = *far* | and | area = *small* | and | ratio = *very-large* | | then | *a-couple* |
| if distance = *far* | and | area = *normal* | and | ratio = *large* | | then | *three-people* |

### *4.3. The Dynamic Expert System Model*

For the action identification stage, a second fuzzy expert system is defined with two input variables and two output (conclusion) variables. Linguistic variables are defined to represent the temporal aspects of the blobs.

– the X-movement change – the change of the centroid of a blob in the x-axis, defined on the range [–5, +5] pixels, can take on five terms: X-movement = {dramatically-left, slightly-left, almost-no-change; slightly-right; dramatically-right}.

- the Y-movement change – the change of the centroid of a blob in the y-axis (in relation to the camera), defined on the range [–5, +5] pixels, can take on four terms: Y-movement = {dramatically-away, slightly-away, almost-no-change; slightly-forward, dramatically-forward}.
- the conclusion concerning the object's velocity, defined on the range [–2, +2] units can take on four terms: Velocity = {standing, slow, fast, running}.
- the conclusion concerning the object's direction, defined on the range [0,1] can take on eight terms: Direction = {right, forward, away-left, away-right, forward-right, forward-left, away, left}.

Figure 4 presents an example of the membership functions defined for one of the linguistic variables called x-axis change. Table 2 presents part of the rule base used for blob action identification.

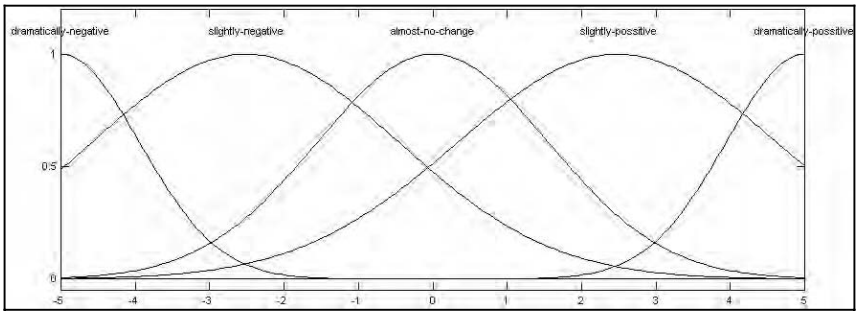



**Figure 4.** Membership functions for the x-axis change conclusion.

**Table 2.** Fuzzy rules for blob action identification

| premise | conclusion |
|---|---|
| If X-movement = *slightly-right* | Then Velocity = *standing* |
| and Y-movement = *almost-no-change* | and Direction = *none* |
| If X-movement = *slightly-right* | Then Velocity = *slow* |
| and Y-movement = *slightly-forward* | and Direction = *forward-right* |
| If X-movement = *slightly-right* | Then Velocity = *fast* |
| and Y-movement = *dramatically-forward* | and Direction = *forward-right* |
| If X-movement = *almost-no-change* | Then Velocity = *standing* |
| and Y-movement = *almost-no-change* | and Direction = *none* |
| If X-movement = *dramatically-left* | Then Velocity = *fast* |
| and Y-movement = *almost-no-change* | and Direction = *right* |

The fuzzy system was implemented with MATLAB's "fuzzy logic tool box" [18]. The operation of the fuzzy system used in this paper was restricted to "Centroid defuzzifier" – that is the center of gravity operator was used to combine the different values resulting from the activation of each fuzzy rule into one crisp result.

Note that this is a prototype system only, so some choices (e.g. the shape of the membership functions) are rather arbitrary. Though further optimization is possible, one of the typical characteristics of fuzzy expert systems is that their predictions are reasonable even with a minimal number of assumptions, and a partial rule-base.

## 5. Application to Understanding Human Activities in a Pedestrian Passageway

A video clip of about 12 seconds (25 frames per second) was taken from the surveillance camera at Ben-Gurion University. The video clip was divided into 299 frames. The pre-processing stage was performed using MATLAB's Image Processing Toolbox. This stage generated all the blobs and their features.

**Figure 5.** Frame no. "180".

Several sets of frames were used to test the performance of the fuzzy systems for different scenarios. In this paper, Frames 170 (figure 1) and 180 (figure 5) are used for demonstrating the behavior of the action identification dynamic fuzzy expert system described in Table 2. Figure 6 is the segmentation of frame 180 (shown in Figure 5) of the video.

Table 3 presents the values of the geometrical attributes; Area, BoundingBox and Centroid for the blobs of frames 170, and 180. The conclusions of the static expert system (of Table 1) regarding the identity of the blobs in figure 2 are presented in bold letters. For example, blob B is correctly identified as a single person. Blob D is correctly identified as a couple.

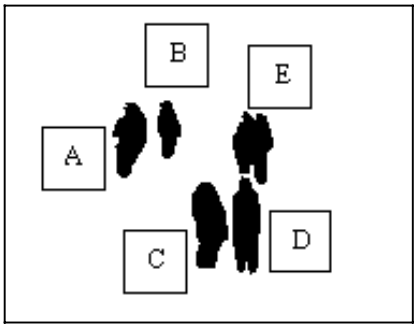



**Figure 6.** Segmentation of frame no. "180".

**Table 3.** Outputs of fuzzy systems regarding blobs in figures 2, 6

| blob identification | A | B | C | D | E |
|---|---|---|---|---|---|
| area (170) (fig. 1) | 378 | 195 | 490 | 401 | 458 |
| boundingbox (170) (fig. 1) | [128.5 36.5 17 37] | [149.5 39.5 10 26] | [164.5 77.5 17 43] | [186.5 41.5 17 31] | [187.5 74.5 15 49] |
| centroid (170) (fig. 1) | [136.6 53.9] | [154.7 51.7] | [173.1 98.1] | [195.2 56.5] | [195.1 97.9] |
| fuzzy blob type conclusion | couple | single | single | couple | single |
| area (180) (fig. 6) | 369 | 204 | 489 | 434 | 474 |
| boundingbox(180) (fig. 6) | [129.5 37.5 16 35] | [150.5 36.5 11 27] | [166.5 74.5 17 40] | [185.5 41.5 19 34] | [185.5 71.5 13 46] |
| centroid (180) (fig. 6) | [137.7 54.1] | [155.6 49.7] | [174.4 94.1] | [195.1 57.5] | [191.9 95.0] |
| fuzzy blob type conclusion | couple | single | single | couple | single |
| X,Y-movements | 1.03 0.19 | 0.92 –1.99 | 1.35 –3.99 | –0.04 1.03 | 3.17 –2.95 |
| fuzzy blob speed conclusion | stand | stand | slow | stand | slow |
| fuzzy blob direction conclusion | none | none | forward-right | none | forward-right |

The last three rows in Table 3 present the input and output variables from the dynamic fuzzy expert system. Based on tracking the outputs of the static expert system the attributes of X,Y movements are determined. The speed and the direction of each blob are identified and shown in the last two rows of the table. For example, blob A was identified as standing. Blob C was identified as moving slowly in the forward-right direction. Thus, the content of the scene could be summarized with five natural language statements such as: "A single person in location (x,y) is moving slowly in the forward-right direction." Furthermore, tracing blobs over time can be used to identify splitting and merging events such as: "one person + two people = three people."

## 6. Conclusions and Discussion

An expert system has been developed for high-level understanding of images from a surveillance video. Two fuzzy inference systems were described – static and dynamic. In the implementation of these two systems the static model is used to identify blobs as clusters of people and provide an estimate of the number of people in each blob. The dynamic model uses temporal information between frames to obtain the movement information of blobs, and with little additional effort can place existing blobs into new categories by identifying merge and split activities. Although correct blob identifications were made for the frames examined, further testing is required. The evaluation of the results of such extended testing requires a performance measure such as the percent of corrected blob types (for the static model), and mean square velocity error (for the dynamic model).

In general, the blob identification system was correct in most cases. Although not implemented here, it is possible to display a "degree of belief" of the expert system regarding the validity of its conclusions. The degree of belief is the value of the membership of the output variables. The larger the membership value, the higher the level of belief. Low degrees of belief will trigger the collection of additional information before presenting a conclusion. Also, further optimization of the fuzzy system will reduce the probability of false classifications.

Content description is often used in video storage for future retrieval [19,20]. In the context of surveillance, it may be interesting to place the linguistic classification in a database. Especially, it may be interesting to record and update a list of people in the scene. This can act as a "pedestrian log." The database can then be used to index the images for archival purposes and subsequent retrieval. For example, each blob in Figure 5 will be indexed by its identified class and its identified actions. It is possible to store only the image information in the bounding box of each blob so as to obtain image compression with minimal information loss. Other expert systems can use this image information to make inferences about other types of activities such as violence, vandalism or theft. It may also be used to study behavioral aspects of crowds or pedestrians. For example, one can query the database for "who gives way to whom," and for "do large blobs act as attractors for passersby"?

## References

[1] Gong S. and Buxton H., Editorial: Understanding visual behavior. *Image and Vision Computing*, Vol. 20, No. 12, pp. 825–826, 2002.
[2] Hanjalic, A. and Li-Qun Xu, Video Analysis and Retrieval at Affective Level, submitted to Special Issue of IEEE Transactions on Multimedia Database, April 2001.
[3] Xu, Li-Qun, Jian Zhu and Fred Stentiford, Video Summarization and Semantics Editing Tools, in SPIE Proceedings – Storage and Retrieval for Media Databases 2001, San Jose, CA, January, 2001.
[4] Gong S., Ng J. and Sherrah J., On the semantics of visual behavior, structured events and trajectories of human action. *Image and Vision Computing*, Vol. 20, No. 12, pp. 873–888, 2002.
[5] Xiang T. and Gong S., Discovering Bayesian causality among visual events in a complex outdoor scene. In Proc. IEEE International Conference on Advanced Video- and Signal-based Surveillance, Miami, USA, July 2003.
[6] Galata A., Learning Spatio-temporal models of object Behavior, 2003.
[7] Xiang T. and Gong S., Scene event recognition without tracking. Special issue on visual surveillance of dynamic scenes, *Acta Automatica Sinica* (Chinese Journal of Automation), Chinese Academy of Sciences, Vol. 29, No. 3, pp. 321–331, 2003.
[8] ICONS: Incident Recognition for Surveillance and Security, DTI/EPSRC LINK Project, July 2000–July 2003, http://www.dcs.qmw.ac.uk/research/vision/projects/ICONS/.
[9] VIGOUR – An Integrated Vision System for Research in Human Recognition, The ISCANIT Project, http://www.dcs.qmw.ac.uk/research/ISCANIT.
[10] Dickinson S.J., Christensen H.I., Tsotsos J.K. and Olofsson G. Active Object Recognition Integrating Attention and Viewpoint Control. Computer Vision and Image Understanding. Vol. 67, No. 3, September, pp. 239–260, 1997.
[11] Lengagne R., Jean-Philippe T. and Olivier M., From 2D Images to 3D Face Geometry. Proceedings of IEEE Second International Conference on Automatic Face and Gesture Recognition, 1996.
[12] Zimmerman H., "Fuzzy Set Theory," Kluwer, 1991.
[13] Cox E., "The Fuzzy Systems Handbook," AP professionals," Boston, 1994.
[14] Shanahan J., Thomas B., Mirmehdi M., Campbell N., Martin T. and Baldwin J., Road Recognition Using Fuzzy Classifiers. Advanced Computing Research Center University of Bristol, 2000.
[15] Balsi M. and Voci F., Fuzzy Reasoning for the Design of CNN – Based Image Processing Systems. Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2000), Geneva, Switzerland, 2000.
[16] Matsakis P., Keller J., Wendling L., Marjamaa J. and Sjahputera O., Linguistic Description of Relative Positions in Images. University of Missouri-Columbia, Columbia, USA, 2000.
[17] Matlab software, Image Processing Toolbox User's Guide. The MathWorks, Inc. www.mathworks.com, 1998.
[18] Matlab software, "fuzzy logic toolbox," The MathWorks Inc., www.mathworks.com, 2000.
[19] Veltkamp, R.C. Burkhardt H., Kriegel H.P. (eds.) State-of-the-Art in Content-Based Image and Video Retrieval, Kluwer, 2001.
[20] CogVis, cogvis.informatik.uni-hamburg.de, "Context-Based Image Retrieval: Performance evaluation and Semantic Scene Understanding," Vogel J., annotated biography.

# 5. TRACKING AND SENSOR FUSION

This page intentionally left blank

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Group Tracking

## *Using Fusion of Simultaneous Track and ID*

Erik BLASCH
*Air Force Research Laboratory, Dayton, Ohio 45433*

**Abstract.** A key issue for situation monitoring, detection, and response management is target tracking, which is a subset of sensor fusion. Tracking includes approaches for single and multitarget tracking, but one area of contemporary research is group tracking. Since users cannot apply attention to hundreds of tracks, they group the moving targets based on categorical features such as allegiance. In this chapter, we show one possible solution to group tracking. The GRoup IMM Tracking (GRIT) algorithm combines the Joint Belief-Probability Data Association (JBPDA) approach with a Variable Structure Interactive Multiple Model (VS-IMM) estimator to track, identify, and group multiple moving targets. Adapting the VS-IMM, a VS-Multiple Validation Gate Model (VS-MVGM) is introduced to capture the behavior of highly maneuvering targets through move-stop-move cycles. Incorporated within the GRIT model are Targets Constrained on Road Networks (TCORN) and High Maneuver Terrain (HMT) features which are assessed in a group tracking scenario to facilitate the representation of multiple targets as groups. The elements of GRIT reduce the amount of tracked information presented on a display. One of the benefits of the GRIT model is the ability to track and identify Moving and Stopping Targets (MST) for groups of targets that are undergoing merging and splitting behavior. Results are presented to demonstrate the incorporation of the MVGM, TCORN, HMT, and MST features to facilitate group tracking.

**Keywords.** Group Tracking, Track and ID, JPBDAF, VS-IMM, VS-MVGM, TCORN

### 1. Introduction

The evolution of target tracking has resulted in many algorithms suited for various multitarget tracking applications [1–7]. In the advancing research on tracking, one issue that plagues the implementation of multitarget tracking applications is that presenting a large number of targets on a display exceeds the human's ability to follow the tracks [13]. Humans naturally group targets into a manageable set for effective comprehension, as shown in Figure 1. In order to facilitate the implementation of tracking algorithms for the user [9,12,13], we seek to group the targets before presenting the track solution to a display in order to minimize clutter and enhance situational awareness.

The Joint Probability Data-Association Filter (JPDAF) algorithm [5] effectively tracks multiple targets and improvements were made to add feature information [9] to identify (ID) targets, eliminate clutter, and locate highly maneuvering targets in the Joint Belief-Probability Data-Association Filter (JBPDAF) algorithm [9,12]. Additionally, using multiple sensors for tracking, detection [20], and identification [13] can sig-

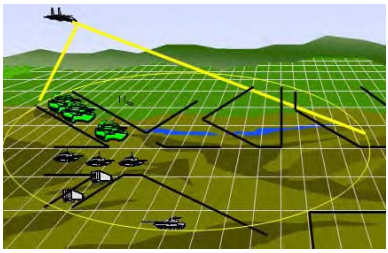


**Figure 1.** Group tracking scenario.

nificantly improve track estimates [17,19]. Examples of track and ID fusion, such as High Range Resolution radar (HRR) [32,33] show improved tracking results [26]. One limitation of the JBPDAF algorithm was capturing moving targets with sharp turns. In an effort to capitalize on detecting moving targets that stop, turn, and move again; it was determined that the Interactive Multiple Model (IMM) algorithm could be useful in tracking high maneuvering targets [29]. The choice of incorporating the IMM with the JBPDAF results from the analysis of the IMMPDAF [23,25] developments in the benchmark problem [8] where the IMM outperformed the Kalman Filter with results comparable to the MHT algorithm [37]. The VS-IMM [3,27,28] is similar to the Multiple Model Estimator (MME) [26] except all motion models need not run at all times. Since the MHT algorithm in a move-stop-move scenario is costly [16], we seek to develop a feasible group tracking algorithm for move-stop-move targets based on the JBPDAF [9,12] and the VS-IMM [3,22].

Group tracking can be performed with measurement information alone; however, utilizing feature information could yield group ID. Blasch [12] presented a group tracking algorithm which utilized track and ID information; however the analysis of the group ID was made for similar type targets. No affiliation of target types was known since the combination of group information was based on similar targets. GRIT includes target-type affiliations using decision logic to determine group structure. Since certain targets with the same allegiance might be closely spaced, additional information is needed to determine how to combine targets for group behavior and uniquely ID targets within a group.

Section 2 overviews the GRIT group tracking methodology which utilizes the VS-IMM [3]. Section 3 develops a multiple model of validation gates to improve the tracking of highly maneuvering targets. Section 4 presents criteria for targets constrained on roads [3] and Section 5 presents a novel approach for tracking moving and stationary targets. Section 6 applies these methods in the GRIT algorithm to facilitate tracking of obscured targets and Section 7 draws conclusions of the work.

## 2. Group Tracking

Group tracking is different than track grouping. Track grouping (e.g. clustering or Centroid Track Grouping (CTG) [18]) first assesses the tracks for each of the targets, clusters, updates all tracks, and then determines which tracks belong to which groups, as shown in Figure 2. Group tracking is a methodology using the group information to update the tracker (e.g. formation group tracking [12]) by assessing a single track for the entire group. In Figure 2, we illustrate the differences in Track Grouping (e.g. clustering) versus Formation Group Tracking (FGT).

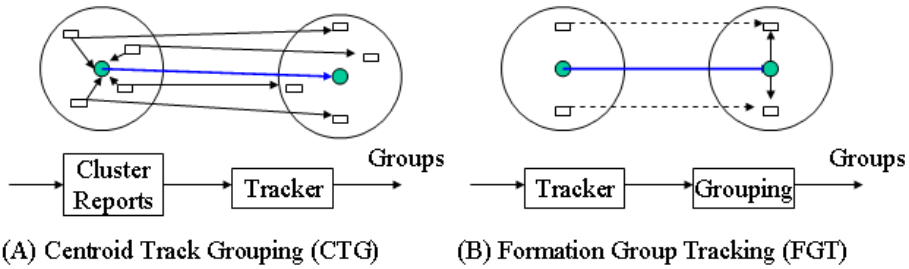



**Figure 2.** Grouping (CTG) versus Group Tracking (FGT).

The difficulty with centroid tracking is that all available track information must be used to determine which targets are assigned to which groups. Thus, grouping information is done *after* the tracks have been formed. While CTG would reduce the amount of track reports presented to a user, it requires lots of overhead and continual observation of all the target tracks. The limitation with formation group tracking is that a formation is constantly changing, as with interacting targets. Thus, the change in formation would result in reducing the group-track accuracy.

Group tracking [12,38,40,30] is a request of operational users. For the group tracking effort, we initiate tracks, categorize targets into groups, and track the group. By only updating a few measurements to the group track, we can reduce computations rather than updating every target. The state and measurement equations for all targets are:

$$\hat{x}(k+1 \mid k) = F(k)\,\hat{x}(k \mid k) \tag{1}$$

$$\hat{z}(k+1 \mid k) = H(k+1)\,\hat{x}(k+1 \mid k) \tag{2}$$

the covariance of the predicted state is,

$$P(k+1 \mid k) = F(k)\,P(k \mid k)\,F^{T}(k) + Q(k) \tag{3}$$

and the covariance associated with the updated state is

$$P(k \mid k) = \beta_0(k)\,P(k \mid k-1) + [1 - \beta_0(k)]\,P^{c}(k \mid k) + \tilde{P}(k) \tag{4}$$

$$\tilde{P}(k) = W(k)\left[\sum_{p=1}^{N_p} \beta_{tp}(k)\,\underline{\nu}_{tp}(k)\,\underline{\nu}_{tp}^{T}(k) - \underline{\nu}(k)\,\underline{\nu}^{T}(k)\right] W^{T}(k)$$

$$P^{c}(k \mid k) = [I - W(k)\,H(k)]\,P(k \mid k-1)\,.$$

where $P$ is the covariance matrix of the state $\hat{x}$; $H$ is the linearized mapping of the state onto the measurement space, $I$ is the identity matrix and $\beta_0$ is the probability that none of the measurements in the validation gate originated from the target in track. The group state and covariance information follows the same procedure (i.e. $\hat{x}_g$, $P_g$). The marginal association probability $\beta_{tp}$ is

$$\beta_{tp} \triangleq \sum_{i} P\{\theta_i \mid Z^k\}\ \hat{\omega}_{tp}(\theta_i) \tag{5}$$

where $\underline{\theta}_i$ is a joint event and $\hat{\omega}_{tp}(\underline{\theta}_i)$ is a binary variable indicating whether joint event $\underline{\theta}_i(k)$ contains the association of track $t$ and measurement $p$. The $i^{th}$ joint event is a hypothesis associating measurements and targets at the $k^{th}$ scan. $\hat{\omega}_{tp}(\underline{\theta}_i)$ is refined to include ID information, move-stop behavior, and group affiliation as discussed in Section 5.

All measurements received from the tracker at time $k$ are compared to the predicted measurements to ascertain whether or not the measurements fall within the kinematic gate. The validation region of the kinematic gate is the elliptical region

$$V(k,\gamma) = \{z : [z - \hat{z}(k \mid k-1)]^{T} S^{-1}(k) [z - \hat{z}(k \mid k-1)] \le \gamma\} \tag{6}$$

where $\gamma$ is the *gate threshold*. These equations introduce the basics for the functionality of the JBPDAF and further information can be gathered from [3,9] as well as equations for the group scenario [12,14].

The JBPDAF simultaneous track and ID algorithm incorporates detection, recognition, ID, and tracking (utilizing a modified JPDAF) [9]. One of the advantages of the approach is that the tracking algorithm can predict the target orientation (or articulation) for ID and pose updates to the tracker. Using an IMM to update target behavior improves results [41]. Thus, the task at hand is to couple the JBPDAF with the IMM to capture group behavior as shown in Figure 3.

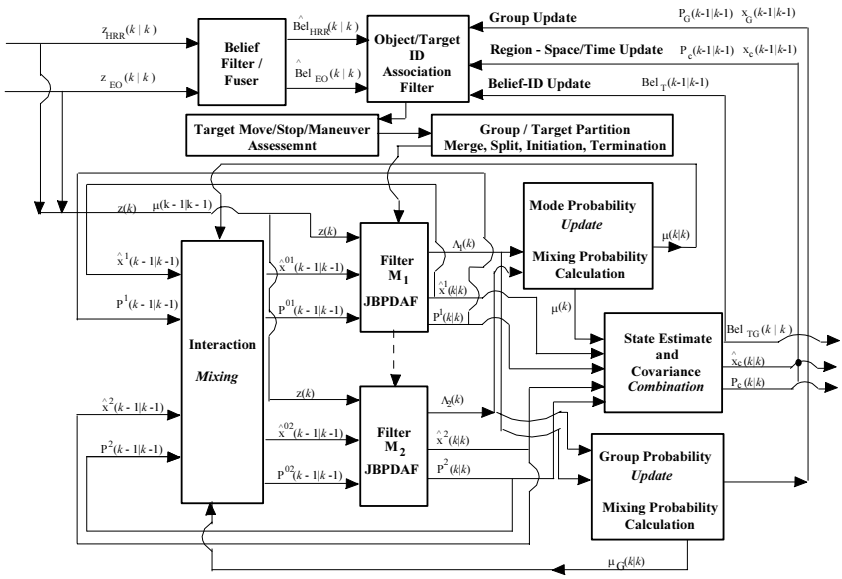



**Figure 3.** Block diagram for implementation of the GRIT.

Adapting the JBPDA filter to accommodate group tracking essentially requires performing the logic of the JPDA [5] twice – once for the measurement-to-track association and once for the track-to-group association. Both association filters are similar to each other with the following exceptions. For measurement-to-track association, the inputs to the JBPDA are the measurements from the detection and the classification sensors and the outputs are the tracks. For track-to-group association, the inputs are the JBPDA track and IDs and the outputs are the group estimates. Another difference is that for track-to-group association, there are more joint association events as a result of having two or more tracks capable of belonging to a single group. For measurement-to-track association, only one measurement could be assigned per track which is based on a mutual exclusivity. But this is no longer the case for track-to-group association since more than one track can be assigned to a group.

The IMM is a high-level algorithm which calls the JBPDA. The IMM can call multiple JBPDA models into effect and uses an optimal combination of these models to determine the final state and covariance estimates. The output of the JBPDA is the state estimate and covariance information to be used by the IMM for the determination of an optimal combination. The JBPDA also outputs to the IMM the innovation and measurement covariance information necessary for the IMM to determine the likelihood functions for each JBPDA model. Figures 4 and 5 show the advantages of coupling IMM (with abrupt motion model) with the JBPDA.

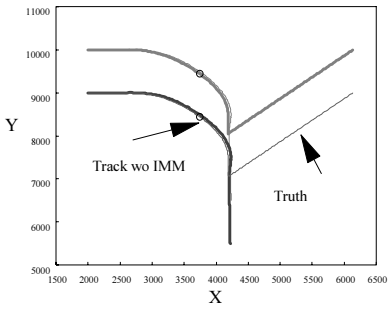



**Figure 4.** Without IMM.

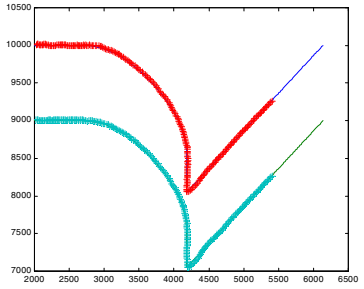


**Figure 5.** With IMM.

Applying the VS-IMM with the JBPDAF, and adding the multiple validation gate model described in this article, we achieve a group tracking scenario capable of tracking multiple groups through highly maneuvering terrain as is demonstrated in Figure 6.

For this scenario, there were 3 groups with 3 tracks each – each group is plotted in a different gray scale. Within each group, there is also a group track plotted in pure black. The thicker, dark black lines represent the road topology. At some points, the tracks are moving side-by-side, and at other points the tracks are moving single file. Also note the 180° tight turn at the right. Here the targets do a 180° turn on a road and proceed in the opposite direction. This tight turn is accomplished by the target initiating a move-stop-turn-move maneuver. The tracker is able to capture this highly maneuvering turn. This scenario was implemented using a Group VS-IMM-JBPDAF methodology employing a VS-multiple validation gate model described in the next section, TCORN, and MST analysis.

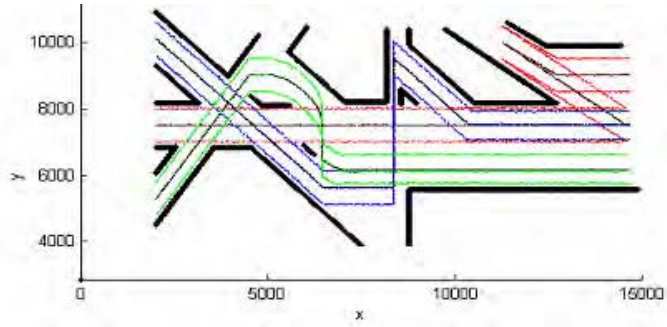



**Figure 6.** Group tracking scenario.

## 3. High Maneuvering Targets with Terrain Information

### *3.1. The Multiple Validation Gate Model (MVGM)*

A significant challenge associated with multi-target tracking as seen in Figure 6 is the capability for the tracking algorithm to positively track targets that are engaged in high maneuver profiles (e.g. tight turns). The MGVM partitions the validation gate to better capture the direction of the group behavior.As can be seen from Figure 7a, the JBPDAF alone is not enough to be able to track targets that undergo a tight turn. This is because the Kalman filter equations predict the location of the next measurement to be along a path based on previous trajectories.

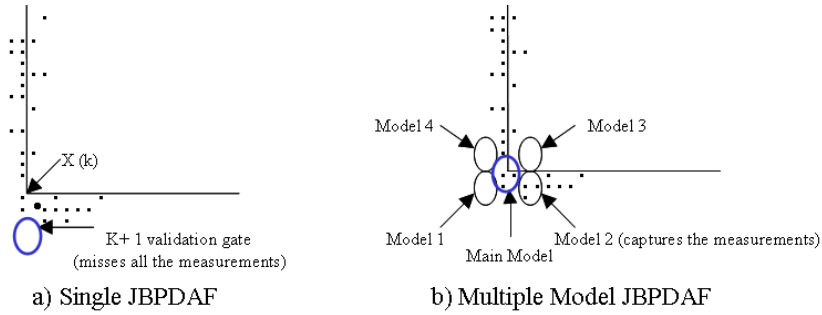



**Figure 7.** Use of a varying gate parameter a) JBPDAF, b) Functionality of applied IMM.

For the traditional JBPDA model, when a tight turn commences, the validation gate of the predicted measurement is nowhere near the actual trajectory of the true measurements. This causes the tracking algorithm to "miss" the track. A typical solution to this is to simply enlarge the size of the validation gate. Doing so, however, will include measurements from other tracks inside the validation gate and would possibly combine two group tracks into one – thereby forcing the tracker off course.

To alleviate this problem, the MVGM uses multiple JBPDA models based on spatially segregated validation regions.Each model predicts the *k+1* measurement to take place in each of four opposite corners from the current target measurement. As with IMM, each model runs in parallel with the other. One model predicts the trajectory to

be in the upper left corner of the current measurement, one model predicts it in the upper right corner, one model predicts the trajectory in the lower left corner, and the last model predicts the trajectory in the lower right corner, as shown in Figure 7b.

When the tracked group commences a tight turn, one of five models is likely to intercept the group. The block diagram for implementation of the MGVM is shown in Figure 2, where the covariance information for a group determines which validation gate is selected. The IMM calls upon up to five JBPDAF models running in parallel to determine the optimum mix of JBPDAF models to accurately track the group. Each model places a spatially segregated validation region in a different location around the predicted measurement. The selection of which models to use is dependent on a known road topology. If a road is known to turn to the left, then only the JBPDAF gate models associated with a left turn are employed by the IMM algorithm. Significant computations and time savings are realized by employing the variable structure. For the MVGM, the format for the $\underline{\boldsymbol{X}}$ state estimate is:

$$\underline{\boldsymbol{X}} = [x \;\; \dot{x} \;\; y \;\; \dot{y}] \tag{7}$$

where $x$ and $y$ are the position estimates of the last time update in the x and y coordinates and $\dot{x}$, $\dot{y}$ are the constant velocities in the x and y coordinates. It is $\dot{x}$, $\dot{y}$ that determines where the location of the validation gate will be predicted for the $k+1$ time step and it is these values that are varied for the different models of the IMM.

Model 1, $x(k+1) = F(k)\, x(k)$ where $x(k) = [x \;\; -\Gamma \;\; y \;\; -\Gamma]$ (8)
– puts validation gate in the lower left hand quadrant
Model 2, $x(k+1) = F(k)\, x(k)$ where $x(k) = [x \;\; \Gamma \;\; y \;\; -\Gamma]$
– puts validation gate in the lower right hand quadrant
Model 3, $x(k+1) = F(k)\, x(k)$ where $x(k) = [x \;\; -\Gamma \;\; y \;\; \Gamma]$
– puts validation gate in the upper left hand quadrant
Model 4, $x(k+1) = F(k)\, x(k)$ where $x(k) = [x \;\; \Gamma \;\; y \;\; \Gamma]$
– puts validation gate in the upper right hand quadrant
Model 5, $x(k+1) = F(k)\, x(k)$ where $x(k) = [x \;\; \beta \;\; y \;\; \theta]$
– validation gate depends on update of main JPDA filter

where $\Gamma$ is a number based on the predicted measurement and validation gate size (innovation covariance) and is desired to be away from the (x, y) coordinate. If the target being tracked is moving quickly, a large $\Gamma$ value should be selected. A slowly moving target dictates a small value of $\Gamma$. For Model 5, $\beta$ and $\theta$ are not fixed values as is $\Gamma$. Instead, $\beta$ and $\theta$ are based on the $\dot{x}$, $\dot{y}$ values as is updated from the JBPDA filter.

Figure 8 shows the move-stop-rotate-move scenario and Figures 9 and 10 are examples demonstrating the effectiveness of the MVGM model. Note that in Figure 9, the JBPDAF tracking algorithm is not able to track through the tight turns. But Figure 10, however, *is* able to capture the tight turns and is able to follow the group track during a 170° turn. Now that we have outlined the methodology for the MVGM, the next section outlines improvements to this methodology by incorporating a variable structure that allows a reduction in the number of JBPDAF models running in parallel. This variable structure MVGM saves significant time and computational burden and is dependent upon knowing *a priori* the topology of a road network.

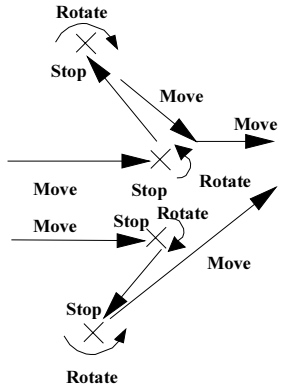



**Figure 8.** Move-stop scenario.

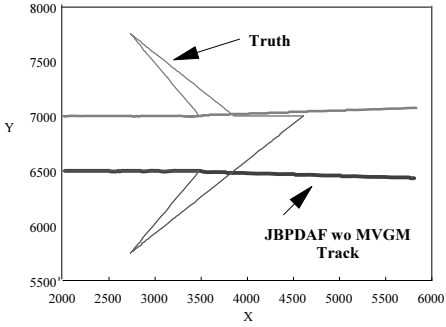



**Figure 9.** JBPDAF without MVGM.

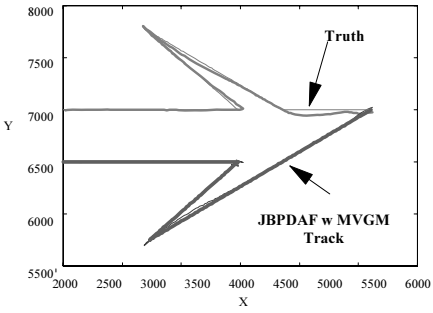



**Figure 10.** JBPDAF with MVGM.

### *3.2. The Variable Structure MVGM*

The incorporation of the VS-MVGM estimator allows the MVGM algorithm to take advantage of a topology of a known road by employing varying mode sets depending on the estimated target position and the corresponding road conditions. If a road has multiple junctions at a single point, then multiple models should be run at that time. If there are only two possible known paths of travel at a particular time instant, then only two models should be run at that point. This differs from the MVGM (without variable structure) from the previous section because in the previous section, all 5 models were being run at the same time. But with VS-MVGM and road knowledge, we can save significant computation time by cutting back from using all 5 models when we don't need them.

In Figure 11, for example, to get from A to B, there is a possibility of taking any one of 3 directions, therefore, we need to run a JBPDAF model for each possible direction. To get from B to C and D to E, there is a possibility of 2 directions for each, therefore, we need to run only 2 JBPDAF models. Lastly, to get from F to G, there is only one possible direction the target could take and so therefore we only need to run 1 JBPDAF model. The result is, by employing this VS-MVGM, significant computational savings can be realized. The VS-MVGM is just one of the advantages realizable

when the topology of a road network is known *a priori*. Another advantage associated with knowing the topology of a road network is that the tracking algorithm can constrain group targets to the proximity of the road network when calculating the contributions of measurements to the track. In essence, we can eliminate those measurements that are found to be located far off the road network. This capability in aiding track performance is called Targets Constrained on Road Networks (TCORN) and is discussed in the next section.

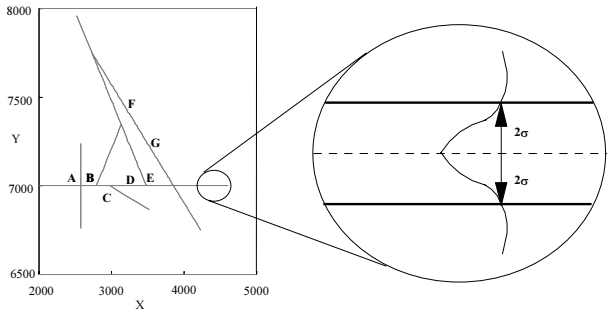



**Figure 11.** A road topology for VS-MVGM and PDF for Road.

## 4. Targets Constrained on Road Networks (TCORN)

Current research [3,34,39] has demonstrated the advantages of using available road information to facilitate the tracking process. TCORN capitalizes on using information assessed from terrain information such as path of travel or acceptable areas of ground target movement of highly traversed areas. One example is that of flat areas in a terrain, which will be referred to as roads, because road networks are inherently flat over a spatial width. Probability density functions (PDFs) for the road network probability are maximum at the center of the road network and tail off outside the road with two standard deviations at the road edge allowing for the possibility of targets transitioning from on-road to off-road, as shown in Figure 11.

Road information can be used in conjunction with a group tracking algorithm to eliminate false targets that are embedded in a target rich, high density environment. A literature review on this topic finds many references to the use of road networks in the aid of target tracking. Ke and Herroro [21] found that tracking techniques with available road information "significantly outperformed the conventional approaches." The implementation of TCORN will demonstrate the ability of the tracking algorithm to eliminate the large number of false targets that lie outside a designated, known road network region. Only those measurements that are "constrained" to the road network will be used toward the estimation of the group track. Another problem of road networks is loss of target ID after groups pass each other on converging roads. When two targets converge to a single point, their affiliation with a group can be confused. A robust group tracking and ID algorithm will correlate the ID of the outbound targets with the ID of the targets *before* converging to the single point. By capitalizing on the belief information of the JBPDA filter, a tracking algorithm can specify which group is associated with which road after passing an intersection. We assume that groups will pass together and that no member of one group will pass in the middle of another group.

The availability of road networks makes the problem of tracking in high density environments an easier task because measurements that are far off- road can essentially be discarded as invalid measurements. For this analysis, we assume that the targets can most likely proceed along the road path. Feature-set classification uses measurements to discriminate between targets both before and after the road crossing. Using this methodology, correct ID of the objects are maintained and correlated before and after the road crossings. With the combined capabilities of both TCORN and group ID, numerous measurements can be disregarded before they ever reach the input of the JBPDA filters – thereby reducing the computational burden of the JBPDAF tracking algorithm. Employing these concepts and using the MVGM provides a robust algorithm for group tracking. The remainder of the chapter will show how the use of group information can facilitate the tracking of both moving and stopping targets and targets whose measurements have been obscured through data dropout.

## 5. Moving and Stopping Targets (MST)

This section proposes a methodology to maintain credible grouping strategies while in the presence of moving and stopping targets. To achieve successful ID of moving targets, a continuous track must be maintained through move-stop-move cycles. We formalize an approach through an assessment of prediction locations of the grouped targets. A "stopped" model was added to the tracker whenever a target fell below some minimum speed to capture groups that are slowing down. This kept targets "alive" even when the measurements were not available. When a group of targets are traveling together and one of the targets has stopped in place, that target should no longer be considered as part of the original group. The GRIT algorithm splits this target off from the original group and maintains the stopped-target as an individual, 1-member group. When the stopped-target resumes its movement and "catches up" to the original group, the tracker must then merge the target back together with the group identity.

As was described earlier, regarding the methodology for performing group tracking, there are two steps required. First is the performance of measurement-to-track association. Secondly is the performance of track-to-group association. For the handling of moving-stopping targets, the process is the same. One difference, however, is that a determination must now be made when a track is in a stationary mode. The typical tracker does not show a return or measurement for a target that is stationary [12]. An imaging mode can be used to assess a stopped target. Only targets in motion are picked up by the tracker. Therefore, a mechanism must be put in place that will accurately determine when a track is moving and when it is stationary. This determination is made during measurement-to-track association. From the JBPDA algorithm: we set up the association matrices as follows:

For a track event, we have:

$$|\hat{\omega}_{jt}(\theta)| \triangleq \begin{cases} 1 & \text{if } \theta_{jt}^{i} \in \theta \quad \text{where mst } [\underline{z}]_{k}^{i} \text{ orginated from track } t \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

For a believable event, which is above a predetermined ID threshold,

$$|\hat{\omega}_{oO}(\theta)| \triangleq \begin{cases} 1 & \text{if } \theta^{i}_{oO} \in \theta \quad \text{where mst } [\underline{Bel}]^{i}_{Ok} \text{ is associated w object } o \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The combined track and classification event (Tracking and Classification Association Matrix) is:

$$|\hat{\omega}_{jot}(\theta)| \triangleq \begin{cases} 1 & \text{if } \theta^{i}_{jot} \in \theta \quad \text{where mst } [\underline{z}]^{i}_{k} \text{ orginated from track } t \text{ with} \\ & \qquad\qquad\qquad\qquad \text{a } [\underline{Bel}]^{i}_{ok} \text{ for a given } O_{ot} \\ 0 & \text{otherwise} \end{cases}$$

where

$$\hat{\omega}_{jot}(\theta) = \hat{\omega}_{jt}(\theta) \oplus \hat{\omega}_{oO}(\theta). \tag{11}$$

The starting point for group tracking lies with the tracking and classification joint association (TCJA) matrix of the JBPDA filter (Figure 12). The upper left number of each square indicates whether or not a measurement is within the kinematic gate – a "one" means the measurement is within the kinematic gate and a "zero" means otherwise. The lower right number of each square indicates whether or not the measurement has an associated ID with a known target type – a "one" means the measurement has a belief ID above a confidence threshold and a "zero" means otherwise.

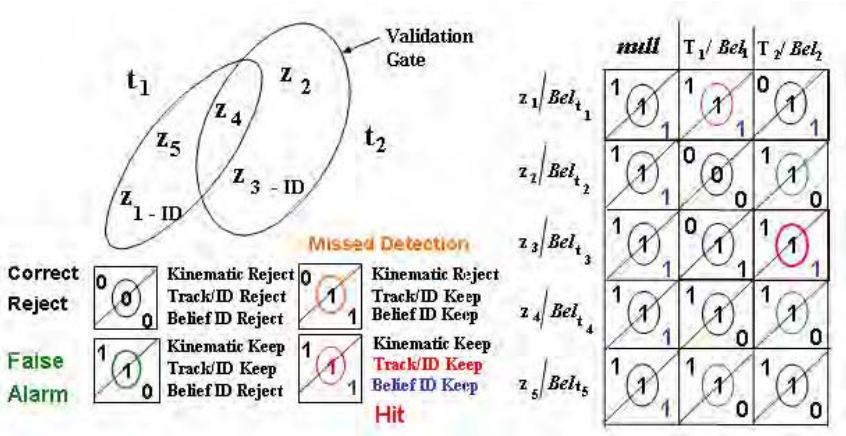



**Figure 12.** Tracking and Classification Joint Association – circles indicate an 'OR'.

The center circle of each square represents the OR function of the previous two events. When an object is stationary, the tracker will not show a return for that object. The TCJA Association matrix will show "zeros" within the track and ID event matrices for the measurement associated with the stationary target. If this is the case, and assuming all other measurements associated with a specific track are "zero," $\sum_{i=1}^{mk} z_i(k) = 0$, the

assumption will be made that that track has stopped in place. The next track update will be shown at the same position as it was at the last time update.

When the target begins moving again, the measurement will once again be registered, and the TCJA matrix will be updated to include a "one" under the specific column of the matrix and the track shall be considered as moving and the JBPDAF logic will update the position of the track estimate accordingly. Note that we assume that there will be no more than 1 measurement within the validation region for a particular track. If you process the TCJA matrix with an 'AND' function, instead of an "OR" function, you will have a more realistic scenario for detecting stopped targets because you will no longer need to assume that there are no other measurements within the validation region for a particular track. Rather, you would only assume that there are no other measurements within a validation region that have an ID that correctly matches that of the known target.

As a result of the measurement-to-track association described above, there are now position estimates for all the tracks involved and we have a record of which tracks are moving and which tracks have stopped.

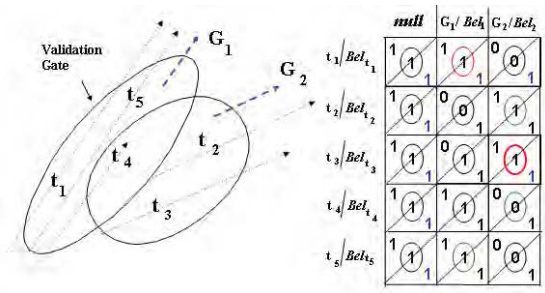



**Figure 13.** Track-Group Joint Association – circles indicate an 'AND'.

The next requirement is to perform the *track-to-group association*. The track-to-group association is a similar analogy to Figure 12, except that we assign tracks to groups. Once the group is identified, we only need to identify a target of the group to update the rest of the members (shown in Figure 13). For track-to-group ID association, we have:

$$|\hat{\omega}_{oG}(\theta)| \triangleq \begin{cases} 1 & \text{if } \theta^{i}_{joG} \in \theta \quad [\underline{Bel}]^{i}_{o_k} \text{ orginated from } G \mid o \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where G is the group and $\hat{\omega}_{oGt}(\theta) = [\hat{\omega}_{jt}(\theta) \oplus \hat{\omega}_{oO}(\theta)] \otimes \hat{\omega}_{oG}(\theta)$, where $\otimes$ denotes an 'AND' function. For the performance of track-to-group association and the handling of moving-stopping targets, we add a new data event that is a result of the previous determination of which tracks are moving and which tracks have stopped. This event is called the group-movement event.

For a group-movement event, we have:

$$|\hat{\omega}_{mt}(\theta)| \triangleq \begin{cases} 1 & \text{if target is moving} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Figure 14 shows a corresponding binary number in the upper right-hand corner of each square to indicate a group-movement event. This number indicates whether or not a track is moving as determined from the previous application of measurement-to-track association, "one" means the measurement is moving and a "zero" means it has stopped.

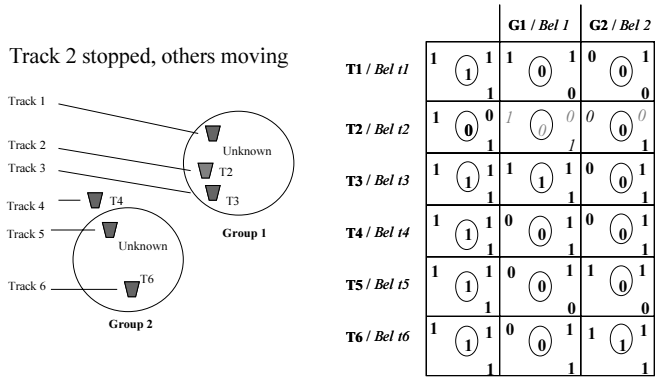



**Figure 14.** Tracking and classification matrix for track to group association.

For track-to-group association, we apply the JBPDA filter adapted for group tracking and assign beta weights [5] to all the measurements for a given track that meet the following condition:

$$\hat{\omega}_{jgt}(\theta) = \hat{\omega}_{jt}(\theta) \oplus \hat{\omega}_{gG}(\theta) \otimes \hat{\omega}_{mt}(\theta) \tag{14}$$

This Boolean algebra has been modified from the measurement-to-track association logic in that we now have a third element, the group-movement event, that we 'AND' (denoted with $\otimes$) with the result of the track and ID events (denoted with $\oplus$ for 'OR'). Pictorially, whenever a track is in a stopped status, the corresponding box of the TCJA matrix will be set to a "0" – regardless of whether or not the track is within the kinematic gate or of the proper ID. This "0" will automatically remove the track that is stopped from being considered as part of the group.

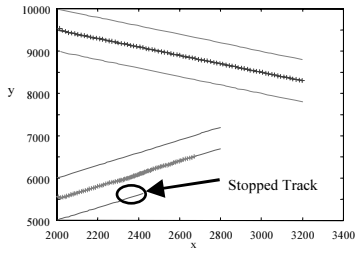



**Figure 15.** JBPDAF without MS Provisions (erroneous group track).

In essence, this action will *split off* the track away from the group track. When a stopped track resumes its movement, the "0" that was set in the TCJA matrix will be

removed and replaced with a "1" thereby allowing the track to be reconsidered as part of the group. The JBPDA logic will then assign a beta weight to that track and use that track as a contributing factor in calculating the next group position update. Rejoining groups are *merged* with the group.

One of the difficult issues in tracking is the ability to split and merge the groups of targets for the assessment of the dynamics of the group of targets being tracked. To perform the merging and splitting, the GRIT algorithm detects that a track has stopped, as discussed above, and splits off the track from the rest of the group, shown in Figure 15 and 16. The track then "falls behind" the rest of the group members. The track won't merge back together with the group until the track begins its movement again and "catches up" to the group identity by falling within the group's validation region. When the track does "catch up" to the group, it will come within the kinematic validation gate and will again be considered as part of the group identity, shown in Figure 17.

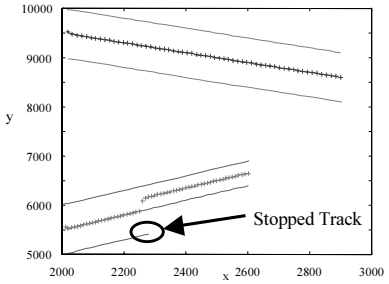



**Figure 16.** JBPDAF with MS Provisions (shows correct group centroid).

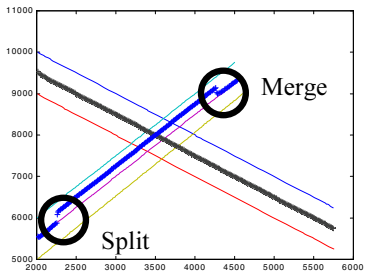



**Figure 17.** Demonstration of tracks splitting off, then re-merging with group track.

In essence, the stopped-track has merged back with the original group. Through this example, it is shown how group information can facilitate the tracking of moving and stopping targets. But moving and stopping targets is not the only area that can benefit from group ID. Another area that can benefit is obscured targets. When targets along a path are blocked or obscured by terrain or other factors, the measurement data needed by the tracking algorithm for the proper track is unavailable. This obscuration produces data dropouts and leads to erroneous track results. Fortunately, group information can also contribute to the correction of the problem of data drop outs. This capability is reviewed in the next section.

## 6. Obscured Targets

Once the group associations have been formed and we know which targets belong to which group, we can use this information to solve the problem of maintaining target track while targets are obscured or overlapping [24]. Obscuration by trees creates data dropouts from the tracker and the tracking algorithm must make assumptions on how to handle such an occurrence, as shown in Figure 18.

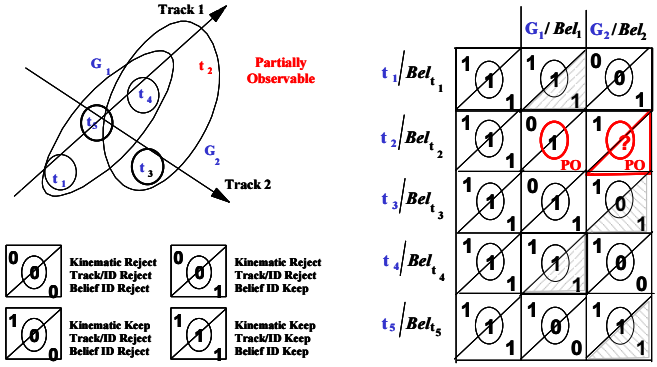



**Figure 18.** The group association matrix for occluded target.

A standard solution to this problem is to assume that a target continues in a straight line from the time that it is obscured by terrain. But this philosophy is readily found to err if the targets are in a nonlinear motion. We assume that an occluded target will continue at the same heading and velocity to keep it in the same relational position as compared to the other members of the group. There are two methodologies to accomplish the group behavior. The first method is a utilization of the group member's track updates. This method uses the state updates from the other members of the group and imposes the average of these state updates onto the track that is *obscured*. This will obtain a track update for the target whose position is unknown. The second method is to measure the distance and bearing information of all the targets within the group as they are related to each other. When one of the targets is *obscured* and has a data dropout, the tracking algorithm will recall from past measurements the distance and bearing information of the missing target from the other members of the group and assume that the same positional relationships are valid at the current time. The algorithm will assume that the correct update of the target whose position is unknown will be based on the predetermined distance and bearing information from the other members of the group. While the tracking algorithm is applying these methodologies to predict the actual position of the track *obscured*, we call this a *coasting* state of the tracker. The group and coasting results are fused to estimate the target location. The tracker will continue in its coasting state until the target emerges from the trees and is re-detected.

Key to the occluded target tracking implementation is the TCJA matrix (reference Figure 18). The functionality of this matrix has been previously described. When a target is obscured, the tracker will not show a target within the kinematic gate (kinematic reject), nor will classification belief be able to show a positive ID (belief ID reject). Therefore, when the tracking algorithm performs an 'OR' operation on these two events, it produces a zero in the TCJA matrix for this particular measurement. The "zero" implies that the tracking algorithm will not use this particular measurement as a factor in calculating the next kinematic update for the track. Otherwise, the presence of a "one" in this matrix would indicate to the tracking algorithm that the measurement *is* a believable event and the algorithm *would* process it as a factor in calculating the *k+1* track update.

If, however, all the measurements within a column of the TCJA matrix are "zero," the algorithm recognizes this scenario for the given track and *coasts* the state of the tracker based on one of two methodologies that follow. Figure 19 presents the imple-

mentation of the first method – utilization of the group member's track updates. The assumption is that the group of targets maintains the same positional orientation amongst themselves. The kinematic update for each track within the group is updated according to the following equation:

$$\hat{\underline{X}}^{t}_{k|k} = \hat{\underline{X}}^{t}_{k-1|k-1} + \mathbf{W}^{t}_{k} \sum_{l=1}^{m^{o}_{k}} \beta^{t}_{lk}\, \nu^{t}_{lk} \tag{15}$$

where $X$ is the target state, $W$ is the filter update, and $\beta$ weighs the innovation $\nu$ for each measurement $m$ within the track t.

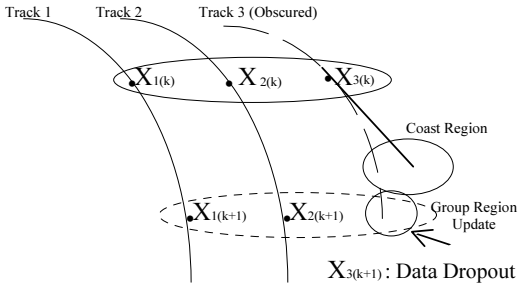



**Figure 19.** Methodology 1 – Update based on group members track updates.

If the tracks within a group maintain the same positional orientation amongst themselves as indicated in Figure 19, then the track update for the missing target should be similar to the track update information of the other tracks within the group. If track 3 was the obscured target as in Figure 19, then the kinematic update for track 3 would be the equation:

$$\hat{\underline{X}}^{3}_{k|k} = \hat{\underline{X}}^{3}_{k-1|k-1} + \frac{1}{2} * \left[ \mathbf{W}^{1}_{k} \sum_{l=1}^{m^{o}_{k}} \beta^{1}_{lk}\, \nu^{1}_{lk} + \mathbf{W}^{2}_{k} \sum_{l=1}^{m^{o}_{k}} \beta^{2}_{lk}\, \nu^{2}_{lk} \right] \tag{16}$$

The second methodology measures the distance and bearing information of all the targets within the group as they are related to each other. When one of the targets is *obscured* and has a data dropout, the tracking algorithm will recall from past measurements the distance and bearing information of the missing target from the other group members and assume that the same positional relationships are valid at the current time. If track 3 is obscured at time *k+1*, the tracking algorithm recalls the distance and bearing that track three was from track 2 as depicted in Figure 20. It also recalls the distance and bearing that track three was from track 1.

The distance between track members within the group is measured by

$$D = \sqrt{(y_2 - y_1)^2} + \sqrt{(x_2 - x_1)^2} \tag{17}$$

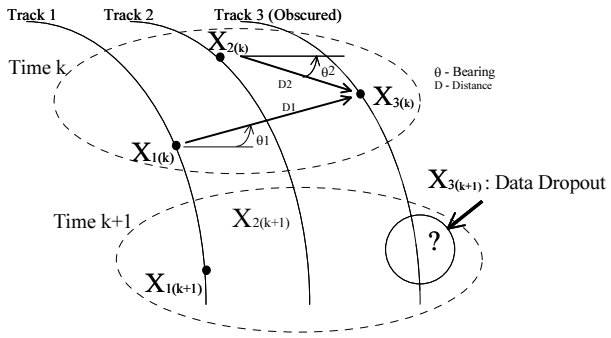



**Figure 20.** Methodology 2 – Track update utilizing bearing and distance info.

where D is the distance between the track members at time *k*. The x, y coordinates of the second track are ($x_2$, $y_2$) and first track are ($x_1$, $y_1$). The bearing information of track members within the group at time *k* is measured by:

$$\theta = \tan^{-1} \frac{(y_2 - y_1)}{(x_2 - x_1)} \tag{18}$$

where θ is the bearing (pose) between track members at time *k*. Using the laws of Sin and Cos, we then can superimpose the heading and distance information onto the time *k* track to arrive at the *k+1* track update.

$$x_3(k+1)(1) = \frac{1}{2} * [(x_1(k+1)(1) + (\text{Cos}\ \theta_1 * d_1) + x_2(k+1)(1) + (\text{Cos}\ \theta_2 * d_2)]$$

$$x_3(k+1)(2) = \frac{1}{2} * [(x_1(k+1)(2) + (\text{Sin}\ \theta_1 * d_1) + x_2(k+1)(2) + (\text{Sin}\ \theta_2 * d_2)] \tag{19}$$

The tracking algorithm will continue to superimpose this heading and distance information on each track update for time *k+1* until the target re-emerges from the occlusion and the tracker detects a new believable measurement.

Figure 21a shows how current JBPDAF implementation does not accommodate the problem of obscured targets. Figure 21 shows track outputs where all the measurements associated with three tracks are located. All three tracks are in a circular movement. Note that the top track has data obscured (missing) about ¾ towards the end of the track. Figure 21b is the resulting track estimate of the measurements based on the JBPDAF algorithm. Note that at the point where the algorithm encounters occlusions, the track estimate for that track goes off in a tangential direction. In fact not only does the track with obscured data go off in the wrong direction, but all three tracks are erroneous at that point. The reason for this is that there is a complete breakdown in the JBPDAF algorithm because targets are assigned to groups. The JBPDAF is based on an estimated known number of tracks from a fixed set of target IDs which form a steady state estimate of the number of targets. When one of the tracks is obscured, the algorithm *sees* only two tracks when it is looking for three.

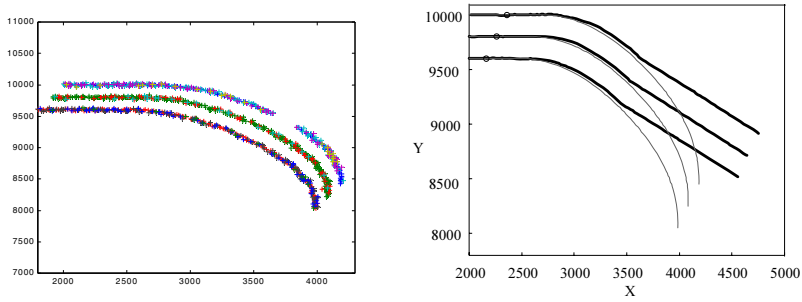



**Figure 21.** (A) Top track has obscured data, and (B) Resulting erroneous track est.

If one of the methodologies previously discussed is used where the missing track estimate is based on the positions and updates from the other group track members, then a satisfactory track estimation can be accomplished. Figure 22a is a plot that shows that the bottom track has spurious measurements and data dropouts. Figure 22b uses methodology 1, *group update*, to accurately predict the truth of the track with the missing data. This shows how group information as determined from the MVGM and TCORN is effective in assisting with the problems associated with target obscuration and data drop out. Like moving and stopping targets, the problem of target obscuration is another area where the benefits of the GRIT can assist with the tracking effort.

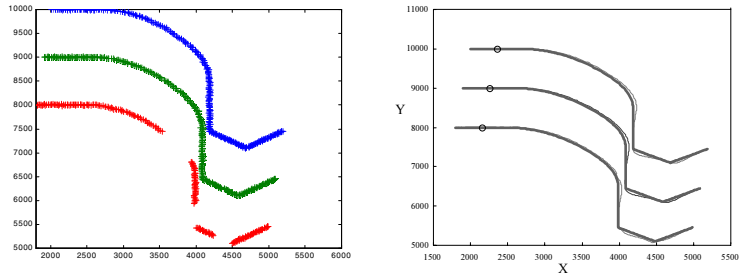



**Figure 22.** (A) Bottom track has obscured data, and (B) Effective track using GRIT.

Figure 23 summarizes the three challenging problems solved by the GRIT algorithm.

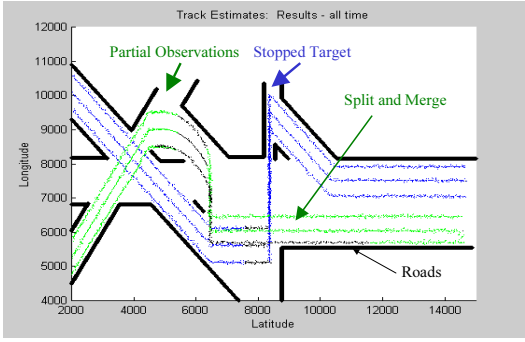



**Figure 23.** Group tracking scenario.

## 7. Conclusions

Recent research has focused on the multi-target, multi-track group tracking problem from user's requests. This article follows current literature and methodologies in adapting the VS-IMM-JBPDAF to include (1) MVGM to capture highly maneuvering targets, (2) TCORN to utilize terrain, and (3) MST to accommodate various target behaviors. An example of GRIT demonstrates how such an algorithm can efficiently track 9 tracks simultaneously and effectively group the tracks. The Group IMM Tracking (GRIT) effectively tracked targets proceeding side-by-side, single file, and in high maneuvering turns. The GRIT algorithm also efficiently demonstrated how information from a network of roads can facilitate the tracking process by eliminating false measurements coming from off-road sources. Benefits of group tracking were also shown to facilitate the problems of data obscuration and moving-stopping targets. In all cases, results showed promising methodologies for handling these challenging tracking scenarios.

## References


[1] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Advanced Applications. Vol. 1*. Dedham, MA: Artech House, 1990. Rep. by YBS Pub. Storrs, CT. 1998.

[2] Y. Bar-Shalom, *Multitarget-Multisensor Tracking. Applications & Advances, Volume II*. Dedham, MA: Artech House, 1992.

[3] Y. Bar-Shalom and T. Kirubarajan, *Ground Target Tracking with Variable Structure IMM Estimator*, IEEE Trans. AES, Vol. 36, No. 1, Jan 2000.

[4] Y. Bar-Shalom and Li, X.R., *Estimation & Tracking: Principles. Techniques & Software*. Dedham. MA: Artech House. 1993. Reprinted by YBS CT. 1998.

[5] Y. Bar-Shalom, and Li, X.R., *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS Publishing, 1995.

[6] S.S. Blackman, S.S. *Multiple-Target Tracking with Radar Applications*. Artech House, Dedham, MA, 1986.

[7] S.S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, Dedham, MA, 1999.

[8] W.D. Blair, G.A. Watson, T. Kirubarajan, and Y. Bar-Shalom, "Benchmark for radar resource allocation and tracking in the presence of ECM," *IEEE Trans. on AES*, 34, 3, pp. 1015–1022, Oct 1998.

[9] E.P. Blasch, *Derivation of a Belief Filter for Simultaneous HRR Tracking and ID,* Ph.D. Thesis, Wright State University, 1999.

[10] E.P. Blasch and T. Connare, "Improving Track Maintenance through Group Tracking," *Workshop on Est., Track, and Fusion*, 2001, 360–371.

[11] E.P. Blasch and T. Connare, "Group Tracking of Occluded Targets", *SPIE* 4365, 2001.

[12] E.P. Blasch and L. Hong, "Group Tracking using Data Association in High Cluttered Environments," *NSSDF,* San Antonio, TX, June 2000.

[13] E.P. Blasch and L. Hong, "Data Association through Fusion of Target track and Identification Sets," *Fusion 2000,* Paris, France, pp. 17–23, 2000.

[14] T.J. Connare, *An IMM-JBPDAF approach to Group Tracking,* Ph.D. Thesis, University of Dayton, 2001.

[15] T. Connare and E.P. Blasch, et al., "Group IMM Tracking utilizing track and Identification Fusion," *Workshop Est., Track, and Fusion*, 2001, pp. 205–220.

[16] S. Coraluppi and C. Carthel, "All-Source Track and Identity Fusion", *National Sensor and Data Fusion Conference,* San Antonio, TX, June 2000.

[17] Z. Ding & L. Hong, "Decoupling probabilistic data association algorithm for multiplatform multisensor tracking," *Optical Eng.*, 37, No. 2, Feb. 1998.

[18] O.E. Drummond, S.S. Blackman, and G.C. Petrisor, "Tracking clusters and extended objects with multiple sensors", *SPIE* 1305, 1990, 362–375.

[19] L. Hong, W. Wang, et al., "Multiplatform multisensor fusion with adaptive rate data communication," *IEEE Trans. AES,*, 33, No. 1, pp. 247–281, 1997.

[20] K. Kastella, Joint multitarget probabilities for detection and tracking, *SPIE AeroSense'97*, pp. 21–25, 1997.
[21] C. Ke and J. Herroro, "Comparative analysis of alternative ground target tracking techniques," *Proc. 3rd Int'l Conf. on Information Fusion,* July 2000.
[22] T. Kirubarajan and Y. Bar-Shalom, *Tracking Evasive Move-Stop-Move Targets with an MTI Radar Using a VS-IMM Estimator,* Univ. of CN, 2000.
[23] T. Kirubarajan, Y. Bar-Shalom, W.D. Blair, and G.A. Watson, "IMMPDA solution to benchmark for radar resource allocation and tracking in the presence of ECM," *IEEE Trans. AES*. 34. 3, pp. 1023–1036, Oct.1998.
[24] T. Kirubarajan, Y. Bar-Shalom, and K.R. Pattipati, "Multi-assignment for tracking a large number of overlapping objects," *SPIE*, 3163 (July 1997).
[25] T. Kirubarajan, M. Yeddanapudi, et al., Comparison of IMMPDA and IMM-Assignment algorithms on real air traffic surveillance data. *SPIE*, 2759, 1996.
[26] J.R. Layne, "Automatic Target Recognition and Tracking Filter," *SPIE AeroSense – Small Targets*, April 1998.
[27] X.R. Li, "Hybrid estimation techniques" In C.T. Leondes (Ed.), *Control and Dynamic Systems*. Vol. 76. San Diego, CA: Academic Press, 1996.
[28] X.R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Trans. on Automatic Control*. 41, pp. 478–493, Apr. 1996.
[29] H. Lin and D.P. Atherton, "An investigation of the SFIMM algorithm for tracking maneuvering targets," *Proc. 32nd IEEE CDC*, pp. 930–935, Dec. 1993.
[30] P.S. Maybeck and K.P. Hentz, "Investigation of moving-bank multiple model adaptive algorithms," *AIAA J. of Guid, Cntrl, & Dyn.*, 10, pp. 90–96, 1987.
[31] C.Y. Mao and S. Li, *An Initialization Method for Group Tracking*, IEEE AES Conf., pp. 303–308. 1995.
[32] R.A. Mitchell, *Robust HRR Target ID using a Statistical Feature Based Classifier with Feature Level Fusion*, PhD Thesis, Univ. of Dayton, 1997.
[33] R.A. Mitchell and J.J. Westerkamp, Robust statistical feature based aircraft identification, *IEEE Trans. AES*, 35, Issue: 3, pp. 1077–1094, July 1999.
[34] B. Noe and N. Collins, "VS-IMM Filter for Tracking Targets with Transportation Network Constraints", *Proceedings of SPIE*, 2000.
[35] K.R Pattipati, S. Deb, Y. Bar-Shalom, and R.B. Washburn, A new relaxation algorithm and passive sensor data association. *IEEE Transactions on Automatic Control*, 37, 2, pp. 198–213, Feb 1992.
[36] A.B. Poore, "Multidimensional assignment formulation of data-association problem arising from multitarget and multisensor tracking," *Computational Optimization and Applications*. 3, 1, 27–57, Mar. 1994.
[37] D.B. Reid, "An algorithm for tracking multiple targets," *IEEE transaction on Automatic Control*, 24, pp. 282–286, 1979.
[38] D. Salmond and N. Gordon, *Group and Extended Object Tracking*, British Crown Copyright, DERA, 1999.
[39] P. Shea and T. Zadra, "Improved state estimation through the use of roads in ground tracking," *Proceedings of SPIE*, 2000.
[40] H. Shyu, Y. Lin, and H. Jinchi, "The Group Tracking of Targets on Sea Surface by 2-D Search Radar," *IEEE Int. Radar Conference,* 1995.
[41] C. Yang and E. Blasch, "Mutual Aided Target Tracking and Identification", *5099 SPIE 03*, April 2003.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Robust Modification of the EM-Algorithm for Parametric Multitrajectory Estimation in Noise and Clutter

Mikhail B. MALYUTOV[a] and M. LU[b]
[a]*Department of Mathematics, Northeastern University, Boston, USA*
[b]*Department of Biostatistics, The University of Iowa, Iowa City, USA*

**Abstract.** A robust family of algorithms generalizing the EM-algorithm for fitting parametric deterministic multi-trajectories observed in Gaussian noise and clutter is proposed. It is based on the M-estimation generalizing the Maximum Likelihood estimation in the M-step of the EM-algorithm. Simulation results of the comparative performance of our algorithm and the traditional EM-algorithm in noise and clutter are described.

**Keywords.** Parametric deterministic multi-trajectory estimation, generalized EM-algorithms, M-estimation, simulation of performance

## 1. Introduction

Traditional methods of multi-trajectory estimation (MTE) are based on preliminary data association (PDA) (measurements and objects are associated at each frame according to some criterion) [1]. The computational complexity of these methods grows exponentially in the number of frames observed, thus requiring supercomputers. We are also unaware of their consistency when the distances between targets are comparable with the standard errors of measurements. Thus it is likely that whatever amount of data and computation is available, the algorithm resolution cannot be made better than certain non-vanishing limits.

The input data is a sequence of frames which are the results of observations of some moving objects in noise and clutter; "clutter" is the component of noise which is correlated in time and space, its popular model is dealt with in [2–4]. The uncorrelated component is the instrumental noise which we assume mutually independent and normally distributed. On the basis of the frames, it is necessary to detect all the objects and to estimate their trajectories. These procedures are based on some prior knowledge about the background, the object's motion equations etc.

Here we deal only with a posteriori *fitting* the motions' parametric trajectories applicable e.g. to the observations of ballistic missiles in noise and clutter. Chronologically speaking, the first efficient method of solving this problem for trajectories described by polynomial equations in *clutter absence*, namely Symmetric Functions of Measurements (SFM), was proposed in the seventies for the former Soviet anti-missile defense, see [5].

We describe another approach to fitting non-random trajectories based on a robust modification of the EM-algorithm defined in the next section, which has a wider applicability range and better accuracy than the SFM-method. A class of measurement errors is generated by the clutter. We assume as in [3], that a clutter generating noise $\varepsilon_t$ is a stationary Gaussian process with zero mean and covariance function $f(t) = 1 - \frac{\lambda_2^2}{2}t^2 + \frac{\lambda_2}{24}t^4 + o(t^4)$ as $t \to 0$ and $f(t) = O(t^{-b})$ as $t \to \infty, 0 < b < 1$. The process $\varepsilon_t$ generates a Poisson process of false targets with intensity rate $\Lambda$, where $\Lambda$ characterizes a clutter intensity on the interval $[A, B]$ and $H_1$ is specified as the lower intensity threshold for a signal to appear on the frame (we observe only presence or absence of signal on the screen). The value $H_1$ is determined according to the intensity of process $\varepsilon_t$ as explained further. A time interval for the presence of a false target on a frame is a positive random variable (RV) with certain density function.

The third class of measurement errors arises when some targets are lost on a frame, which is modeled as follows. Let the actual target signal level be $H_2 + H_1$. A target is lost at a frame, if the value of the clutter process is less than $-H_2 < 0$. The sign symmetry of the clutter Gaussian process implies similar formulas for the rate of missing and false targets with $H_2$ plugged into the formula below instead of $H_1$.

Starting times for targets becoming missed constitute the Poisson process with rate $\Lambda_{tr} = \frac{\lambda}{2\pi} e^{-H_2^2/2}$, and the time interval at which the target is lost has a random length with a density function $\frac{1}{2}\pi a t e^{-\frac{\pi a^2 t^2}{4}}$, where $a = \frac{H_2\lambda}{\sqrt{2\pi}}$ for every target [6]. We keep the E-step of the EM-algorithm as described in [7,8] and replace the weighted Least Squares (LS) estimation of trajectories parameters by its robust weighted M-version on the M-step. The maximal break-down point of this algorithm is attained by the Least Median of Squares (LMS) estimator [9,10] which is very slow. The Huber kernel for the M-estimator outperforms the LS-estimator in the case of strong clutter, and is comparatively fast. We presented our simulation results at the NATO ASI-2003 in Tsakhkadzor, Armenia. The code is available from the second author.

## 2. Generalized EM-Algorithms

### *2.1. Introduction to the EM Algorithm*

The estimation-maximization (EM) iterative algorithm was created in the sixties for mixtures estimation (by several authors independently) and (by L.E. Baum et al.) for the parameter estimation of Hidden Markov Models. Dempster et al. [7]) in 1977 reformulated it as a general technique for finding maximum likelihood estimates from incomplete data and gave an erroneous proof of its convergence under general conditions to a local maximum of the likelihood function starting from a sufficiently close initial parameter point, which was corrected in [11] and in more theoretical papers of I. Csiszar and G. Tusnady. Our application is a considerably simplified version of the

Hidden Markov model with a deterministic evolution of the hidden state. We use a justification of generalized EM-algorithms (GEM) proposed in [7,8] based on representation of the EM-algorithm as an alternating penalized maximization procedure over parameters and posterior distribution of missing values.

The EM algorithm formalizes an old idea for handling missing data starting from a plausible parameters' distributions: 1) replace missing data values by certain posterior means, 2) re-estimate parameters, 3) re-estimate the missing values pretending that the new parameter estimates are correct, 4) re-estimate parameters, and so forth, iterating until convergence. Such methods are simple for models where the complete data log likelihood $l(\theta \mid Y_{obs}, Y_{mis}) = lnL(\theta \mid Y_{obs}, Y_{mis})$ is linear in $Y_{mis}$; more generally, missing sufficient statistics rather than individual observations need to be estimated, and even more generally, the log likelihood $l(\theta \mid Y)$ itself needs to be estimated at each iteration of the algorithm.

Each iteration of the EM consists of an $\mathbf{E}$ step (posterior expectation for missing values) and a $\mathbf{M}$ step (maximizing the posterior likelihood). The M step is simpler to describe: maximize the likelihood for parameter $\theta$ given a posterior distribution of the missing data in the preceding iteration. Thus the **M** step of the EM algorithm uses the weighted ML-estimation. The **E** step fits the conditional expectation of "missing data" given the observed data and current estimated parameters, and then substitutes these expectations for the "missing data." These steps are often easy to construct, to program for calculation, and to fit into computer storage. Also, each step has a direct statistical interpretation.

An additional advantage of the algorithm is its steady convergence: under general conditions, each iteration increases the log likelihood $l(\theta \mid Y_{obs})$, and if $l(\theta \mid Y_{obs})$ is bounded, the sequence $l(\theta \mid Y_{obs})$ generally converges to a stationary value of $l(\theta \mid Y_{obs})$. Usually, if the sequence $\theta^{(m)}$ converges, it converges to a local maximum or a saddle point of $l(\theta \mid Y_{obs})$.

A general description of the EM-algorithm is as follows:

– $E-$ step. Determine $Q(\theta, \theta^{(m)})$, where

$$Q(\theta, \theta^{(m)}) = E\,(\log f(Y_{complete} \mid \theta) \mid Y_{observed}, \theta^{(m)}).$$

– $M-$ step. Determine $\theta^{(m+1)} = arg\max_{\theta} Q(\theta \mid \theta^{(m)})$.

An EM algorithm chooses $\theta^{(m+1)}$ to maximize Q($\theta \mid \theta^{(m)}$) with respect to $\theta$. More generally, a **GEM** (generalized **EM**) chooses $\theta^{(m+1)}$ so that Q($\theta^{(m+1)} \mid \theta^{(m)}$) is greater than Q($\theta^{(m)} \mid \theta^{(m)}$). Say, we replace it by its robust version. The following is the key straightforward result of [7]:

Every GEM algorithm increases $l(\theta \mid Y_{obs})$ at each iteration, that is

$$l(\theta^{(m+1)} \mid Y_{obs}) \geq l(\theta^{(m)} \mid Y_{obs})$$

with equality if and only if

$$l(\theta^{(m+1)} \mid Y_{obs}) = l(\theta^{(m)} \mid Y_{obs})$$

*2.2. LMS Estimation for the MTE*

We model the motion $x_k[t]$ of k-th object as polynomial $\theta_{\mathbf{k}}[\mathbf{t}]$ of order n, of t determined by its coefficients $\theta_{\mathbf{k}}[\mathbf{t}] := (\theta_{\mathbf{k0}}, \ldots, \theta_{\mathbf{kn}}, \mathbf{k} = 1, \ldots, \mathbf{K})$. In the EM algorithm for the MTE, the M-step is reduced to finding $\theta_{j0}, \cdots, \theta_{jn}$ minimizing (under assumption of independent normal instrumental errors):

$$\sum_{t=1}^{T} \sum_{k=1}^{n_t} w_{kj} (x_k[t] - \theta_j[t])^2, \qquad j = 1, 2, \cdots, K;$$

and we use the *sweep* operator described in [12] to find $\theta_{j0}, \theta_{j1}, \cdots, \theta_{jn}$ and $\sigma_j$.

False targets and missing targets influence the M-step similarly to the contamination dealt with in robust regression techniques. In contrast to [13] we prefer to use robust non-parametric versions [3], [14] of the M-step instead of fitting a parametric family of the clutter models for their analysis based on their excessively strong assumption of known parametric family of clutter.

In the LMS estimation for MTE, we find $\theta_{j0}, \theta_{j1}, \cdots, \theta_{jn}$ minimizing

$$\operatorname{med}_{1 \le t \le T} \sum_{k=1}^{n_t} w_{kj} (x_k[t] - \theta_j[t])^2, \qquad j = 1, 2, \cdots, K, \tag{1}$$

To solve (1.2), we use the following scanning algorithm based on consecutive minimizations in the obvious way:

$$\min_{\theta_{jn}} \cdots \min_{\theta_{j0}} \operatorname{med}_{1 \le t \le T} \sum_{k=1}^{n_t} w_{kj} ((x_k[t] - \theta_{jn} t^n) \cdots - \theta_{j0})^2,$$

where $j = 1, 2, \cdots, K$.

For the simplest case n=0, to minimize

$$\operatorname{med}_{1 \le t \le T} \sum_{k=1}^{n_t} w_{kj} (x_k[t] - \theta_{j0})^2 \tag{2}$$

we use a transformation $\theta_{j0} = \tan\alpha$ and search through all angles $\alpha$ from –1.55 rad to 1.55 rad with a step-size of 0.01 rad to approximate the minimal value in (2), and

then we scan with a precision of 0.001 or less rad in the two most promising areas (as determined by the first step).

For n=1, we search in the following way:

$$\min_{\theta_{j1}} \min_{\theta_{j0}} \operatorname*{med}_{1\le t\le T} \sum_{k=1}^{n_t} w_{kj}((x_k[t]-\theta_{j1}t)-\theta_{j0})^2 \tag{3}$$

Indeed, we can treat the parts in (3) separately. The second portion of the minimization can be solved for the n=1 case. Then we have to find $\hat{\theta}_{j1}$ for which

$$m^2(\theta_{j1}) = \min_{\theta_{j0}} \operatorname*{med}_{1\le t\le T} \sum_{k=1}^{n_t} w_{kj}((x_k[t]-\theta_{j1}t)-\theta_{j0})^2$$

is minimal. This is a minimization of a one-dimensional function $m^2(\theta_{j1})$. In order to find $\hat{\theta}_{j1}$, we use the same techniques as in the case where n=1.

For $n > 2$, the scanning would require a tremendous amount of computation.

Finally, we use $\hat{\theta}_{j0}, \hat{\theta}_{j1}, \cdots, \hat{\theta}_{jn}$ to estimate $\hat{\sigma}_j$, $j = 1, 2, \cdots, K$:

$$\hat{\sigma}_j^2 = \operatorname*{med}_{1\le t\le T} \sum_{k=1}^{n_t} w_{kj}(x_k[t]-\theta_{j0}-\theta_{j0}t-\cdots-\theta_{jn}t^n)^2$$

*2.3. M-Estimators*

M-estimators generalize ML-estimators, for the model

$$y_i = g_i(\beta) + \varepsilon_i$$

(smooth parametric function in noise), where $\mathrm{E}\,\varepsilon_i = 0$, $\mathrm{E}\,\varepsilon_i^2 = \sigma^2$, $\beta = (\beta_1, \beta_2, \cdots, \beta_p)$, $\varepsilon'_i s$ are independent if $\varepsilon_i \sim \mathrm{N}(0, \sigma^2)$, then $\hat{\beta}_{ML} = \hat{\beta}_{LS}$.

The ML estimator under a non-degenerate design of observations

$$\hat{\beta} = \operatorname{argmin} \frac{1}{2} \sum_{i=1}^{n} \left( \frac{y_i - g_i(\beta)}{\sigma} \right)^2$$

is the unique solution of the estimating equations

$$\sum_{i=1}^{n} \frac{\partial g_i(\hat{\beta})}{\partial \beta_j} \left( \frac{y_i - g_i(\hat{\beta})}{\sigma} \right) = 0 \qquad \mathrm{j} = 1, \cdots, \mathrm{p}$$

The problem with the ML estimation is that for objective function $\rho(z) = z^2/2$, the least squares minimization heavily weighs large absolute residuals, $|y_i - g_i(\beta)|$. Huber [15,16], proposed a more robust approach using a different $\rho$-function, which puts less weight on extreme residuals. Define $\Psi(z) = d\rho(z)/dz$. The M-estimation

$$\hat{\beta} = \operatorname{argmin} \sum_{i=1}^{n} \rho\left(\frac{y_i - g_i(\beta)}{\sigma}\right)$$

is a solution of estimating equations

$$\sum_{i=1}^{n} \frac{\partial g_i(\hat{\beta})}{\partial \beta_j} \Psi\left(\frac{y_i - g_i(\hat{\beta})}{\sigma}\right) = 0 \qquad j = 1, \cdots, p$$

P. Huber [14] proposed

$$\rho_H(z) = \begin{cases} \frac{1}{2}z^2 & \text{if} \quad |z| \le H \\ H|z| - \frac{H^2}{2} & \text{if} \quad |z| > H \end{cases}$$

and

$$\Psi_H(z) = \frac{d\rho_H(z)}{dz} = \begin{cases} -H & \text{if} \quad z < -H \\ z & \text{if} \quad |z| \le H \\ H & \text{if} \quad z > H \end{cases}$$

Traditionally, H equals 1.5 $\tilde{\sigma}$, $\sigma$ can be estimated as $1.483 med_i \{|y_i - \text{med}_l \{y_l\}|\}$.

Some alternatives to Huber's $\rho$ function were proposed, by D.F. Andrews (see [9,17]):

$$\rho_A(z) = \begin{cases} A^2(1 - \cos(z/A)) & \text{if} \quad |z/A| \le \pi \\ A^2 & \text{if} \quad \text{otherwise} \end{cases}$$

$$\Psi_A(z) = \begin{cases} A\sin(z/A) & \text{if} \quad |z/A| \le \pi \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$$

and J. Tukey's weight $\rho$-function (see [9,17])

$$\rho_B(z)=\begin{cases}\frac{1}{2}B^2[1-(1-(z/B)^2)^3] & \text{if} \quad |z|\le B\\ \frac{1}{2}B^2 & \text{if} \quad \text{otherwise}\end{cases}$$

$$\Psi_B(z)=\begin{cases}z[1-(z/B)^2]^2 & \text{if} \quad |z|\le B\\ 0 & \text{if} \quad \text{otherwise}\end{cases}$$

For MTE, in the M-step we find $\theta_{j0},\cdots,\theta_{jn}$ minimizing

$$\sum_{t=1}^{T}\sum_{k=1}^{n_t} w_{kj}\rho_H(x_k[t]-\theta_j[t]) \quad \mathrm{j}=1,2,\cdots,\mathrm{K}$$

where $\rho_H$ is the Huber's function, $\sigma_j$ j =1 ,2, $\cdots$, K are estimated as

$$\left(\frac{\sum_{t=1}^{T}\sum_{k=1}^{n_t} w_{kj}\rho_H(x_k[t]-\hat{\theta}_j[t])}{T-n-1}\right)^{\frac{1}{2}}$$

### 3. Simulation Results

In our simulation, the number of frames (observations) is 70 and the number of trajectories is 3. We assume the missing rates for each trajectory are either $\lambda_1$ = 0.4, or $\lambda_2$ = 0.3, or $\lambda_3$ = 0.3 and the standard deviation of noise $\sigma$ is 0.1. The true trajectories are defined as

$$x=0.1t, \quad y=0.2t$$
$$x=0.1t, \quad y=2+0.2t+0.001t^2$$
$$x=1+0.1t, \quad y=0.13t+0.001t^2$$

1. Clutter Rate = 0.3

In this case, the number of false measurements is 18, the number of missing data points is 50, and the number of observed data (including the false observations) is 178.

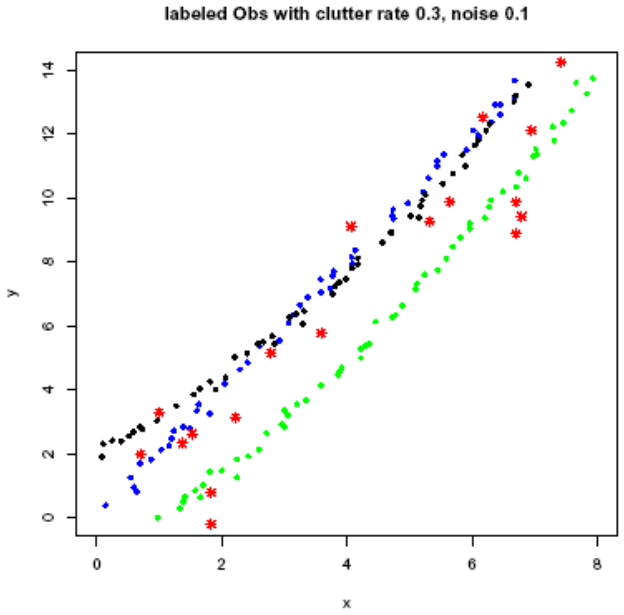



**Figure 1a.** Observations with clutter rate 0.3 and noise 0.1.

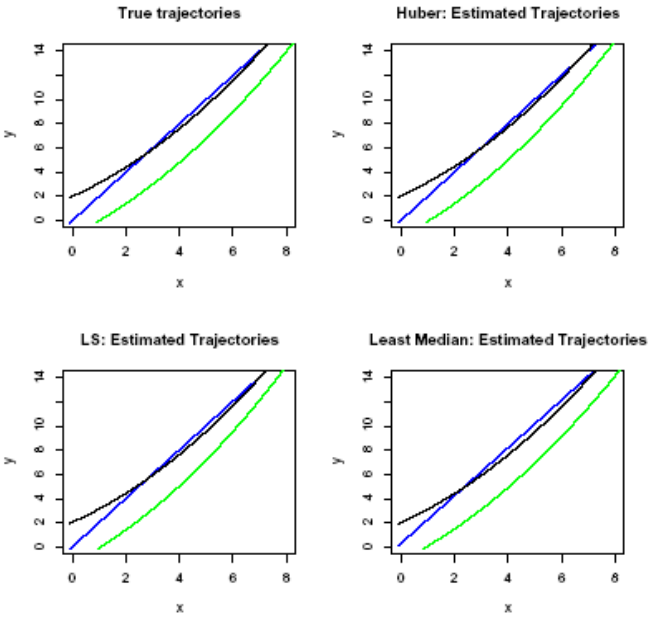



**Figure 1b.** Real and estimated trajectories.

**Table 1a.** First trajectory

| Feature | True Value | LS | LMS | Huber | missing |
|---|---|---|---|---|---|
| $x_0$ | 0 | 0.0017 | 0.0190 | 0.0016 | 19 |
| $x_1$ | 0.1 | 0.0998 | 0.0973 | 0.0998 | 19 |
| $x_2$ | 0 | 0 | 0 | 0 | 19 |
| $\sigma_x$ | 0.1 | 0.1450 | 0.0704 | 0.093 | |
| $y_0$ | 0 | 0.0146 | 0.3247 | 0.0152 | 19 |
| $y_1$ | 0.2 | 0.2002 | 0.1923 | 0.2001 | 19 |
| $y_2$ | 0 | 0 | 0 | 0 | 19 |
| $\sigma_y$ | 0.1 | 0.0830 | 0.1062 | 0.0725 | |

**Table 1b.** Second trajectory

| feature | true value | LS | LMS | Huber | missing |
|---|---|---|---|---|---|
| $x_0$ | 0 | –0.0056 | 0.0260 | –0.0053 | 19 |
| $x_1$ | 0.1 | 0.0984 | 0.0993 | 0.0983 | 19 |
| $x_2$ | 0 | 0.0003 | 0 | 0 | 19 |
| $\sigma_x$ | 0.1 | 0.0876 | 0.0516 | 0.0627 | |
| $y_0$ | 2 | 2.0512 | 2.0554 | 2.051 | 19 |
| $y_1$ | 0.1 | 0.0967 | 0.0993 | 0.0968 | 19 |
| $y_2$ | 0.001 | 0.001 | 0.001 | 0.001 | 19 |
| $\sigma_y$ | 0.1 | 0.1006 | 0.0558 | 0.0719 | |

**Table 1c.** Third trajectory

| feature | true value | LS | LMS | Huber | missing |
|---|---|---|---|---|---|
| $x_0$ | 1 | 1.047 | 0.9543 | 1.0472 | 12 |
| $x_1$ | 0.1 | 0.0955 | 0.1003 | 0.0955 | 12 |
| $x_2$ | 0 | 0 | 0 | 0 | 12 |
| $\sigma_x$ | 0.1 | 0.0774 | 0.0582 | 0.0815 | |
| $y_0$ | 0 | 0.0013 | –0.02 | 0.0012 | 12 |
| $y_1$ | 0.13 | 0.1311 | 0.1307 | 0.1311 | 12 |
| $y_2$ | 0.001 | 0.001 | 0.001 | 0.001 | 12 |
| $\sigma_y$ | 0.1 | 0.0840 | 0.0754 | 0.0719 | |

2. Clutter Rate = 1.1

In this case, the number of false measurements is 79, the number of missing data points is 50, and the number of observed data (including false ones) is 239. The mean-based estimation breaks down if the missing rate is greater than 0.3.

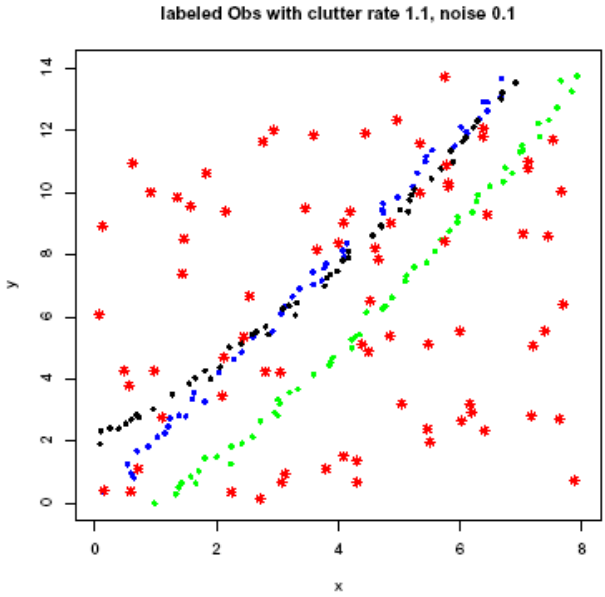



**Figure 2a.** Observations with clutter rate 1.1 and noise 0.1.

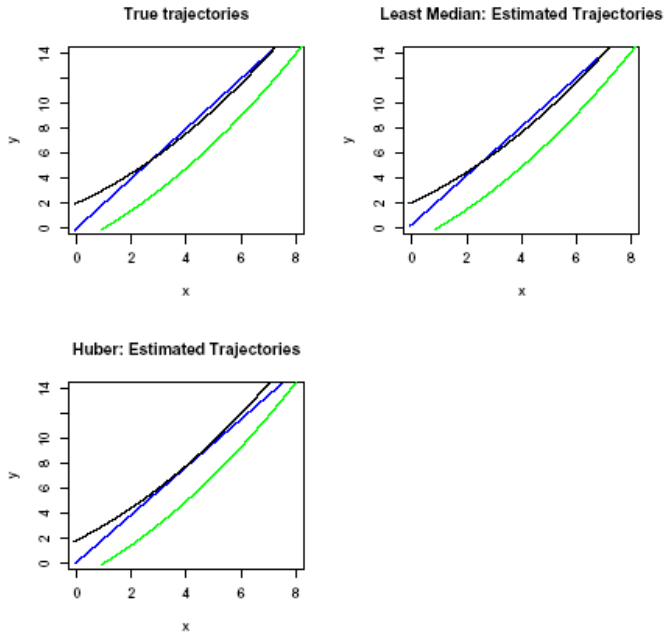



**Figure 2b.** Real and Estimated Trajectories.

**Table 2a.** First trajectory

| feature | true value | LS | LMS | Huber | missing |
|---|---|---|---|---|---|
| $x_0$ | 0 | N/A | –0.007 | 0.048 | 19 |
| $x_1$ | 0.1 | N/A | 0.1003 | 0.098 | 19 |
| $x_2$ | 0 | N/A | 0 | 0 | 19 |
| $\sigma_x$ | 0.1 | N/A | 0.1158 | 0.1901 | |
| $y_0$ | 0 | N/A | 0 | 0.1572 | 19 |
| $y_1$ | 0.2 | N/A | 0.2006 | 0.1914 | 19 |
| $y_2$ | 0 | N/A | 0 | 0 | 19 |
| $\sigma_y$ | 0.1 | N/A | 0.0777 | 0.2897 | |

**Table 2b.** Second trajectory

| feature | true value | LS | LMS | Huber | missing |
|---|---|---|---|---|---|
| $x_0$ | 0 | N/A | –0.010 | –0.03 | 19 |
| $x_1$ | 0.1 | N/A | 0.1003 | 0.1012 | 19 |
| $x_2$ | 0 | N/A | 0 | 0 | 19 |
| $\sigma_x$ | 0.1 | N/A | 0.0737 | 0.1001 | |
| $y_0$ | 2 | N/A | 1.9794 | 1.807 | 19 |
| $y_1$ | 0.1 | N/A | 0.1003 | 0.109 | 19 |
| $y_2$ | 0.001 | N/A | 0.001 | 0.001 | 19 |
| $\sigma_y$ | 0.1 | N/A | 0.0963 | 0.238 | |

**Table 2c.** Third trajectory

| feature | true value | LS | LMS | Huber | missing |
|---|---|---|---|---|---|
| $x_0$ | 1 | N/A | 0.9741 | 1.002 | 12 |
| $x_1$ | 0.1 | N/A | 0.0993 | 0.098 | 12 |
| $x_2$ | 0 | N/A | 0 | 0 | 12 |
| $\sigma_x$ | 0.1 | N/A | 0.0741 | 0.079 | |
| $y_0$ | 0 | N/A | 0.0340 | –0.0298 | 12 |
| $y_1$ | 0.13 | N/A | 0.1287 | 0.1315 | 12 |
| $y_2$ | 0.001 | N/A | 0.001 | 0.001 | 12 |
| $\sigma_y$ | 0.1 | N/A | 0.1042 | 0.096 | |

The following table shows the running time for the three versions of the M-step under different rates of clutter.

| clutter rate | clutter num | LS | M-estimation | LMS |
|---|---|---|---|---|
| 0.3 | 18 | 7s | 8s | 24m 32s |
| 1.1 | 79 | N/A | 12s | 26m 50s |

In the simulation, the LS algorithm breaks down if the rate of clutter is more than 0.3, that is, 18 clutters in the observations. The M-estimation won't be good enough if the rate of clutter is more than 1.1. However, the LMS still works at the rate of 1.3.

## 4. Conclusions

Our simulation results confirmed our theoretical expectation that the median-based estimation and M-estimation are more robust than the mean-based estimation. With a small clutter rate, the mean-based, median-based estimation and M-estimation all work pretty well. However, the mean-based estimation breaks down if a large number of false measurements appear in the data. The median-based estimation and M-estimation continue to work robustly under high clutter rate. The median-based estimation method is very time-consuming, thus to work on-line it must be replaced by a faster implemented M-estimation block, say, the one based on the Huber kernel.

Future work for the median-based GEM will include improvements geared toward reducing the algorithm's running time.

## References

[1] Bar Shalom Y. and X.R. Li. 1995. Multitarget-Multisensor Tracking. Principles and Techniques, YBS, Storrs, Connecticut, 1995.
[2] Streit R. Track initialization sensitivity in clutter, In Proceedings of the Conference on Signal and Data Processing of Small Targets, SPIE International Symposium on Aerospace, San Diego, CA, SPIE, 460–471, 1995.
[3] Malyutov, M.B. and I. Tsitovich. Modeling multi-target estimation in noise and clutter. Proceedings, 12th European Simulation Symposium, ESS 2000 (Simulation in Industry) Sept. 28–30, Hamburg Germany, Soc. for Computer Simulation, Delft, Netherlands, 598–600, 2000.
[4] Malyutov, M.B., M. Lu, A. Nikiforov and R. Protassov. MultiTarget estimation in noise and clutter, Proceedings, International Conference Fusion 2001, August 2001, Montreal, Canada, Fr.C, 17–25.
[5] Bernstein, A.V. Group regression parameter estimation. Izvestia of USSR Academy of Sciences, Technical Cybernetics: 137–141, 1973.
[6] Cramer H. and M.R.Leadbetter. Stationary and related stochastic processes. Sample function properties and their applications. John Wiley, New-York, London, 1973.
[7] Dempster, A.P., N.M. Laird; and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. J. Royal Statist. Soc. Ser. B (methodological) 39: 1–38, 1977.
[8] Neal, R.M. and Hinton, G.E. A view of the EM algorithm that justifies incremental, sparse, and other variants, in M.I. Jordan (editor) Learning in Graphical Models, Kluwer Academic Publishers, Dordrecht, 355–368, 1998.
[9] Rousseeuw, P.J., and Yohai,V. Robust regression by means of S-estimator, edited by J.Franke, W.Hardle, and R.D.Martin, Lecture Notes in Statistics No. 26. Springer-Verlag, New York, pp. 256–272, 1984.
[10] Donoho, D.L. and Huber, P.J. The notion of breakdown point, in Festschrift for Erich Lehmann, edited by P. Bickel, K. Doksum, and J.L. Hodges, Jr., Wadsworth, Belmont, CA, 1983.
[11] Wu, C.F.J. On the convergence properties of the EM algorithm. Ann. Statistics, 95–103, 1983.
[12] Beaton, A.E. The use of special matrix operations in statistical calculus, Educational Testing Service Research Bulletin, RB-64-51, 1964.
[13] Molnar K.I. and J.W. Modestino. Application of the EM algorithm for the Multitarget/Multisensor Tracking Problem. IEEE Transactions on Signal Processing.: 115–129, 1998.
[14] Huber, P.J. Robust methods of estimation of regression coeffcients. Math. Operationsforschung Statist. Ser. Statist 8, 41–53, 1977.
[15] Huber, P.J., Robust statistics: A review. Ann. Math. Statist. 43, 1041–1067, 1972.
[16] Huber, P.J., Robust estimation of a location parameter. Ann. Math. Statist. 35, 73–101, 1964.
[17] Hampel, F.R., E.M. Ronchetti, P.J. Rousseeuw and W.A. Stahel. Robust Statistics The Approach Based on Influence Functions. John Wiley & Sons, 1986.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Non-Parametric Multi-Trajectory Estimation

M.B. MALYUTOV[a], Alexandre B. TSYBAKOV[b] and Ion GRAMA[c]
[a]*Department of Mathematics, Northeastern University, Boston, USA*
[b]*Probability Lab., Paris Universite-6*
[c]*Lab. SABRES, Universite de Bretagne, 5600 Vannes, France*

**Abstract.** The convergence rate for a non-parametric estimation of two distinct smooth trajectories based on pairs of their non-assigned noisy measurements is studied using the kernel estimates for two symmetric functions of observations and the roots of the corresponding parabolic equation. Some simulation results showing the performance of our method are presented.

**Keywords.** Indistinguishable targets, symmetric functions of measurements method, kernel estimates, roots of random polynomials

## 1. Introduction and Outline of the Problem

### *1.1. Simplified Examples*

To explain the novelty of the setting and the ideas proposed, we start with a related simplified problem (of interest in the Reliability theory [1]). Suppose, we toss 2 indistinguishable generally biased coins *simultaneously*, and unknown probabilities of heads in a long sequence of independent trials are $p_1, p_2$ respectively. Suppose that *only the total numbers of heads* $\sigma(i)$ in trials $i = 1, ..., N$, are known. We outline now the Symmetric Functions of Measurements (SFM) method to estimate the set $\{p_1, p_2\}$ of unknown parameters in this and related problems.

Denote $s := p_1 + p_2, \pi := p_1 p_2$ and notice that the set of roots to the quadratic equation $z^2 - sz + \pi = 0$ is exactly $\{p_1, p_2\}$. Next, $s(N) := \sum_{i=1}^{N} \sigma(i)/N$ and $N_2/N$ :=number( outcome $2)/N$ consistently estimate respectively $s$ and $\pi$.

Hence the set of roots to the equation $z^2 - s(N)z + N_2/N = 0$ estimates consistently the set $\{p_1, p_2\}$ as the number of trials $N \to \infty$. The rate of convergence of the SFM estimates for the set of head probabilities is found in [1] for an arbitrary number $n$ of indistinguishable coins flipped simultaneously. If the total number of heads in each of $N$ trials is corrupted by random error, we can generally propose only the methods based on the EM-algorithm (see [2]) to estimate consistently the set of head probabilities.

A next *static* MultiTrajectory estimation example simplifies the one dealt with in [3,4], and [5]. Suppose in each of $N$ independent experiments we observe $2$ real or planar points. Each of them is the noisy measurement $Y_i = a_i + e_i$ of one of fixed centers $a_i, i = 1,2$ on the line (or on the plane) but *no information is available on assigning random measurements to the centers*. How to estimate the set of the centers consistently?

Introduce $s := a_1 + a_2, \pi := a_1 a_2$ and notice that the set of roots to the quadratic equation $z^2 - sz + \pi = 0$ is exactly $\{a_1, a_2\}$ considered as complex numbers. Denote $\sigma(j) := \sum_{i=1}^{2} Y_i(j), \pi(j) := \prod_{i=1}^{2} Y_i(j)$. Under, say, symmetric Gaussian independent errors $e_i(j)$ of $j$-th measurement $s(N) := \sum_{j=1}^{N} \sigma(j)$ and $\Pi_N := \sum_{j=1}^{N} \pi(j)/N$ estimate consistently respectively $s$ and $\pi$. Hence the set of roots to the quadratic equation $z^2 - s(N)z + \Pi(N) = 0$ converges to the set $\{a_1, a_2\}$.

The $\sqrt{N}$ rate of convergence is easy to prove given that the centers are different. A more complicated case of multiple roots is treated in [6]. A more general set up of parametric family of trajectories observations corrupted by noise and clutter is studied using a robust version of the EM-algorithm in [2].

### *1.2. Method Outline*

In this paper we extend the SFM method to the problem of noisy observations of two smooth trajectories. An intuitive initial idea is to approximate the trajectories by piecewise constant ones. Then the values of trajectories on the intervals of constancy can be estimated as before. However, such a method has poor convergence properties. To improve the rate of convergence, the moving window (kernel) estimation replaces this naive piecewise approximation approach. More complicated is the estimation of both trajectories near their crossing points, where the roots of the quadratic equation (similar to that considered above) are close to each other. To keep the convergence rate of the method almost the same as for the case of disjoint trajectories, we can first estimate their derivatives by the kernel method applied to SFM, and then restore the trajectories by integration in small neighborhoods of intersection points.

**Remark.** The SFM method was applied to estimate parameters of MultiTrajectories of fixed or moving (according to a polynomial regression model) targets in [4,5]. The stepwise algorithm of these papers uses asymptotically infinite divergence of polynomial trajectories, and will likely fail if the trajectories stay permanently close to each other. The EM-approach of [2] (enabling in addition to estimate the parameters of the random assignment: targets-to-observations) and the approach of the present paper seem to be free of this deficiency.

## 2. Nonparametric Setting, Non-Intersecting Trajectories

Here we introduce the MSF estimates in a nonparametric MultiTrajectory setting and estimate their rates of convergence.

Consider the model

$$Y_{i1} = f_1(t_i) + \varepsilon_{i1},$$
$$Y_{i2} = f_2(t_i) + \varepsilon_{i2}, \quad i = 1, ..., n. \tag{1}$$

Here $f_1(\cdot) : [0,1] \to \mathbf{R}$, $f_2(\cdot) : [0,1] \to \mathbf{R}$, are unknown smooth functions, $t_i = i/n$, and $\varepsilon_{i1}, \varepsilon_{i2}$ are random variables such that $\varepsilon_{11}, \cdots, \varepsilon_{n1}$ are i.i.d., $\varepsilon_{12}, \cdots, \varepsilon_{n2}$ are i.i.d., and the vectors of random variables $(\varepsilon_{11}, \cdots, \varepsilon_{n1})$ and $(\varepsilon_{12}, \cdots, \varepsilon_{n2})$ are mutually independent.

We are given unordered pairs of observations $(Y_{11}, Y_{12}), \cdots, (Y_{n1}, Y_{n2}),$ such that for each pair of values $(Y_{i1}, Y_{i2})$ we do not know which value is $Y_{i1}$ and which is $Y_{i2}$. More accurate estimation would take into account and estimate the parameters of the random assignment mechanism which we will not pursue in this paper. Our problem is to estimate the functions $f_1(\cdot), f_2(\cdot)$ given these observations. We suppose that $f_1(\cdot), f_2(\cdot)$ are smooth, as stated in the following assumption.

**Assumption 1.** *Let $\beta > 0, L > 0$ and $C_0 > 0$ be finite constants. We assume that the functions $f_1(\cdot), f_2(\cdot)$ belong to $\Sigma(\beta, L, C_0)$, where $\Sigma(\beta, L, C_0)$ is the class of all functions on $[0,1]$ bounded in absolute value by $C_0$ and such that their derivative of order $\lfloor \beta \rfloor$ satisfies $(\beta - \lfloor \beta \rfloor)$-Hölder condition with constant $L$, where $\lfloor \beta \rfloor$ is the maximal integer that is strictly less than $\beta$.*

Denote by $f_{n1}(x), f_{n2}(x)$ the roots of the quadratic equation

$$Z^2 - s_n(x)Z + \pi_n(x) = 0, \tag{2}$$

if these roots are real, and set $f_{n1}(x) = f_{n2}(x) = 0$ otherwise. Here $s_n(x), \pi_n(x)$ are symmetric functions of measurements introduced below such that, under appropriate conditions

$$s_n(x) \xrightarrow{P} f_1(x) + f_2(x), \ \pi_n(x) \xrightarrow{P} f_1(x) f_2(x), \text{ as } n \to \infty$$

Hence the quadratic function in (2) converges in probability to the function $F(Z) = Z^2 - (f_1(x) + f_2(x))Z + f_1(x) f_2(x)$ (uniformly in $Z$ on every bounded

interval). Clearly, the equation $F(Z)=0$ has the roots $f_1(x)$ and $f_2(x)$. The roots of (2) converge to $f_1(x), f_2(x)$ in probability as $n\to\infty$ (cf., e.g. [7]).

Now we define our method. Consider the estimation of $f_1, f_2$ at arbitrary fixed point $x\in(0,1)$. Define the statistics

$$s_n(x)=\sum_{i=1}^{n}(Y_{i1}+Y_{i2})W_{ni}(x),\quad \pi_n(x)=\sum_{i=1}^{n}Y_{i1}Y_{i2}W_{ni}(x), \tag{3}$$

where $W_{ni}(x)$ is a weight function such that $\sum_{i=1}^{n}W_{ni}(x)=1$ or $\sum_{i=1}^{n}W_{ni}(x)=1+o(1)$, as $n\to\infty$. In the following we consider the kernel weights

$$W_{ni}(x)=\frac{1}{nh}K\left(\frac{t_i-x}{h}\right)$$

where $K:\mathbf{R}\to\mathbf{R}$ is a kernel and $h>0$ is a bandwidth, but one can also consider other weight functions $W_{ni}(x)$ used in nonparametric estimation problems (see e.g. [7]). We will need the following assumption.

**Assumption 2.** *The random variables* $\varepsilon_{i1},\varepsilon_{i2}$ *are normal with* $E(\varepsilon_{i1})=E(\varepsilon_{i2})=0$, $E(\varepsilon_{i1}^2)=\sigma_1^2<\infty$, $E(\varepsilon_{i2}^2)=\sigma_2^2<\infty$.

**Theorem.** Let Assumptions 1 and 2 be satisfied, and let $K(\cdot)$ be a compactly supported Lipschitz continuous function such that for $l=\lfloor\beta\rfloor$ we have

$$\int u^m K(u)du=0, m=1,...,l,\quad \int K(u)du=1 \tag{4}$$

(*i.e.* $K$ *is a kernel of order* $l$). *Set* $h=\alpha n^{-\frac{1}{2\beta+1}}$ *for some* $\alpha>0$. *If* $|f_1(x)-f_2(x)|\ge t_0 n^{-\frac{\beta}{2\beta+1}}\log n$ *for some* $t_0>0$, *then*

$$\sup_{f_1,f_2\in\Sigma(\beta,L,C_0)} E_{f_1,f_2}(f_{nj}(x)-f_j(x))^2\le Cn^{-\frac{2\beta}{2\beta+1}}, j=1,2,$$

where $C>0$ is a finite constant. Here $E_{f_1,f_2}$ denotes the expectation with respect to the joint distribution of $(Y_{i1},Y_{i2}, i=1,...,n)$ in model (1).

**Proof.** Repeating the standard argument of nonparametric regression estimation for $s_n(x)$ (see, e.g., [7]), we obtain

$$\sup_{f_1,f_2\in\Sigma(\beta,L,C_0)} E_{f_1,f_2}((s_n(x)-[f_1(x)+f_2(x)])^2)=O(n^{-\frac{2\beta}{2\beta+1}}),n\to\infty. \tag{5}$$

For $\pi_n(x)$ consider separately the bias and variance term. Using the Taylor expansion, the bias of $\pi_n(x)$ can be evaluated as follows

$$Bias=E_{f_1,f_2}(\pi_n(x))-f_1(x)f_2(x)$$
$$=\frac{1}{nh}\sum_{i=1}^{n}f_1(t_i)f_2(t_i)K\left(\frac{t_i-x}{h}\right)-f_1(x)f_2(x)$$

$$=\frac{1}{nh}\sum_{i=1}^{n}[f_1(x)f_2(x)+(f_1f_2)'(x)(t_i-x)+\dots$$
$$+\frac{(t_i-x)^l}{(l-1)!}\int_0^1(f_1f_2)^{(l)}\left(x+\tau(t_i-x)\right)(1-\tau)^{l-1}d\tau]K\left(\frac{t_i-x}{h}\right)-f_1(x)f_2(x)$$
$$=\frac{1}{nh}\sum_{i=1}^{n}\frac{(t_i-x)^l}{(l-1)!}K\left(\frac{t_i-x}{h}\right)\int_0^1(f_1f_2)^{(l)}(x+\tau(t_i-x))(1-\tau)^{l-1}d\tau+O\left(\frac{1}{nh}\right)$$
$$=O\left(h^\beta+\frac{1}{nh}\right)$$

where we used the fact that the derivatives $(f_1f_2)^{(m)}$, $m\le l$, are uniformly bounded for $f_1,f_2\in\Sigma(\beta,L,C_0)$ and that

$$\left|\frac{1}{nh}\sum_{i=1}^{n}\left(\frac{t_i-x}{h}\right)^m K\left(\frac{t_i-x}{h}\right)-\int u^m K(u)du\right|=O\left(\frac{1}{nh}\right).$$

The variance of $\pi_n(x)$ has the form

$$Var=E_{f_1,f_2}((\pi_n(x)-E_{f_1,f_2}[\pi_n(x)])^2)=$$

$$E([\underbrace{\frac{1}{nh}\sum_{i=1}^{n}\xi_{i1}f_2(t_i)K(\frac{t_i-x}{h})}_{Z_1}+$$

$$\underbrace{\frac{1}{nh}\sum_{i=1}^{n}\xi_{i2}f_1(t_i)K(\frac{t_i-x}{h})}_{Z_2}++\underbrace{\frac{1}{nh}\sum_{i=1}^{n}\xi_{i1}\xi_{i2}K(\frac{t_i-x}{h})}_{Z_3}]^2.$$

The terms $Z_1, Z_2$ are treated in a similar way, in particular,

$$E\left(Z_1^2\right) = \sigma_1^2 \frac{1}{n^2 h^2} \sum_{i=1}^{n} f_2(t_i)^2 K\left(\frac{t_i - x}{h}\right)$$
$$= O\left(\frac{1}{nh}\right), n \to \infty, \text{ since } \sup_x | f_2(x) | \leq C_0.$$

For $Z_3$ we get

$$E\left(Z_3^2\right) = \frac{1}{n^2 h^2} \sum_{i,k=1}^{n} E\left(\xi_{i1}\xi_{i2}\xi_{k1}\xi_{k2}\right) K\left(\frac{t_i - x}{h}\right) K\left(\frac{t_k - x}{h}\right)$$
$$= \frac{1}{n^2 h^2} \sum_{i=1}^{n} \sigma_1^2 \sigma_2^2 K^2\left(\frac{t_i - x}{h}\right)$$
$$= O\left(\frac{1}{nh}\right), \; n \to \infty.$$

Thus for our choice of h:

$$\sup_{f_1, f_2 \in \Sigma(\beta, L, C_0)} E_{f_1, f_2}[(\pi_n(x) - f_1(x) f_2(x))^2]$$
$$= O(h^{2\beta} + \frac{1}{nh})$$
$$= O(n^{-\frac{2\beta}{2\beta+1}}), n \to \infty. \tag{6}$$

It follows from (5), (6) that $s_n(x)$ and $\pi_n(x)$ converge in probability for any $x$ to $s(x) = f_1(x) + f_2(x)$ and $\pi(x) = f_1(x) f_2(x)$ respectively at the rate $n^{-\frac{\beta}{2\beta+1}}$. Hence, the discriminant $(s_n^2(x)/4) - \pi_n(x)$ of the random quadratic equation (2) converges in probability to the discriminant $(f_1(x) - f_2(x))^2/4$ of the equation $F(Z) = 0$ at the same rate. Since we assume that $| f_1(x) - f_2(x) | \geq t_0 n^{-\frac{\beta}{2\beta+1}} \log n$, which is logarithmically larger than the rate of convergence in probability $n^{-\frac{\beta}{2\beta+1}}$, we can guarantee that $(s_n^2(x)/4) - \pi_n(x) > 0$ with probability close to 1 for $n$ large enough. Therefore, with probability close to 1, we have

$$f_{n1}(x) = \frac{s_n(x)}{2} + \sqrt{\frac{s_n^2(x)}{4} - \pi_n(x)}, \tag{7}$$

$$f_{n2}(x)=\frac{s_n(x)}{2}-\sqrt{\frac{s_n^2(x)}{4}-\pi_n(x)}. \tag{8}$$

Using (5), (6) we see that the right hand sides of (7), (8) converge in probability to $f_1(x)$ and $f_2(x)$ respectively at the rate $n^{-\frac{\beta}{2\beta+1}}$. Finally, a uniform integrability argument permits to obtain the convergence of second moments and thus to complete the proof.

## 3. Regularly Intersecting Trajectories

Here we extend the above construction for a more complicated case of two smooth ($\beta \geq 2$) possibly intersecting trajectories. Two trajectories will be called regularly intersecting if for all $\Delta > 0$ small enough the inequality $|f_1(x)-f_2(x)|<\Delta$ is satisfied only for $x$ belonging to an interval of length less than $C_1\Delta$, where $C_1 > 0$ is a constant. Here we assume that the trajectories intersect regularly and we keep all the conditions and notation of the previous section. This work is still in progress. In particular, we have not yet completed the simulation of the performance of the algorithm outlined below which we apply only in the neighborhoods of the intersection points.

First, note that the derivatives $f_1'(x)$ and $f_2'(x)$ can be estimated consistently at points $x$ such that $|f_1(x)-f_2(x)|$ is not too small. In fact, acting as in the previous section, it is not hard to show that the derivative

$$s_n'(x)=\sum_{i=1}^{n}(Y_{i1}+Y_{i2})W_{ni}'(x)$$

estimates the function $f_1'(x)+f_2'(x)$ with mean squared error (MSE) of order $n^{-\frac{2(\beta-1)}{2\beta+1}}$ at any point $x$. Quite similarly,

$$\pi_n'(x)=\sum_{i=1}^{n}Y_{i1}Y_{i2}W_{ni}'(x)$$

estimates $\pi'(x)=(f_1(x)f_2(x))'=f_1'(x)f_2(x)+f_2'(x)f_1(x)$ with MSE of order $n^{-\frac{2(\beta-1)}{2\beta+1}}$. Thus, $f_1'(x)(f_1(x)-f_2(x))$ is estimated with MSE of order $n^{-\frac{2(\beta-1)}{2\beta+1}}$ by

$$v_n(x)=s_n'(x)f_{n1}(x)-\pi_n'(x).$$

Dividing this expression by $f_{n1}(x)-f_{n2}(x)$ from the preceding section, we obtain the following estimate of the derivative $f_1'(x)$:

$$f_{n1}^{(1)}(x)=\frac{s_n'(x)f_{n1}(x)-\pi_n'(x)}{f_{n1}(x)-f_{n2}(x)}$$

Analogously, we define the estimate $f_{n2}^{(1)}(x)$ of the derivative $f_2'(x)$.

Next, we evaluate how close is $f_{nj}^{(1)}(x)$ to $f_j^{'}(x), j=1,2,$ in the situation where $x=x_n$ is such that $|f_1(x_n)-f_2(x_n)|=\Delta_n>0$, where $\Delta_n\to 0$, as $n\to\infty$. We will do this only for $j=1$, since the case $j=2$ is similar. Using the above argument and the result of the previous section we obtain that if $\Delta_n > t_0 n^{-\frac{\beta}{2\beta+1}}\log n$,

$$\left|f_{n1}^{(1)}-f_1'(x_n)\right|=$$

$$\left|\frac{f_1'(x_n)(f_1(x_n)-f_2(x_n))+O_p(n^{-\frac{\beta-1}{2\beta+1}})}{(f_1(x_n)-f_2(x_n))+O_p(n^{-\frac{\beta}{2\beta+1}})}-f_1'(x_n)\right|\le\frac{O_p(n^{-\frac{\beta-1}{2\beta+1}})}{\Delta_n+O_p(n^{-\frac{\beta}{2\beta+1}})}.$$

The last expression tends to 0 in probability only if $\Delta_n >> n^{-\frac{\beta-1}{2\beta+1}}$. For definiteness, take in what follows $\Delta_n=n^{-\frac{(\beta-1)}{2\beta+1}}\log n$. Then we get

$$|f_{nj}^{(1)}(x_n)-f_j^{'}(x_n)|=O_p(\Delta_n^{-1}n^{\frac{\beta-1}{2\beta+1}})=O_p(1/\log n),\ \text{as}\ n\to\infty \tag{9}$$

We are now ready to describe our procedure. First, choose a point $x_n$ in a neighborhood of the intersection, i.e. an $x_n$ satisfying $|f_{n1}(x_n)-f_{n2}(x_n)|\approx\Delta_n$, where $f_{n1}$ and $f_{n2}$ are estimators of $f_1$ and $f_2$ defined in the previous section. In an $O(\Delta_n)$-neighborhood of $x_n$ we define new adjusted estimators of $f_1$ and $f_2$ by the formula

$$\hat{f}_{nj}(x)=f_{nj}(x_n)+f_{nj}^{(1)}(x_n)(x-x_n),\quad j=1,2.$$

From (9) and the result of the previous section, using the Taylor expansion of $f_j$ in a neighborhood of $x_n$ (recall that $\beta \geq 2$), we now deduce that

$$\begin{aligned} |\hat{f}_{nj}(x) - f_j(x)| &\leq |f_{nj}(x_n) - f_j(x_n)| + |f_{nj}^{(1)}(x_n) - f_j(x_n)|\,|x - x_n| \\ &\quad + O\left((x - x_n)^2\right) \\ &\leq O_p\left(n^{-\frac{\beta}{2\beta+1}}\right) + O_p\left(1/\log n\right)\Delta_n + \left(\Delta_n^2\right) \\ &= O_p\left(n^{-\frac{(\beta-1)}{2\beta+1}}\right), \text{ as } n \to \infty \end{aligned}$$

Thus, we see that the estimation of trajectories around the intersection is possible with the rate $n^{-\frac{(\beta-1)}{2\beta+1}}$ which is slightly slower than the rate $n^{-\frac{\beta}{2\beta+1}}$ obtained sufficiently far from the intersection (see the previous section). Such a loss of accuracy seems natural because the problem is more complicated.

## 4. Simulation Results

We use the Java applet prepared by M. Lu to simulate the algorithm described for various parameters of error variance and window size. Here

$$f_1(x) = -\left(x - \frac{1}{2}\right)^2 + \frac{1}{4}$$

$$f_1(x) = -\left(x - \frac{1}{2}\right)^2 + \frac{5}{4}$$

$$K(u) = \frac{3}{4}(1 - u^2)I(u),$$

where $x \in$ [0,1] and $I(u) = 1$ if $u \in [0,1]$, otherwise $I(u) = 0$.

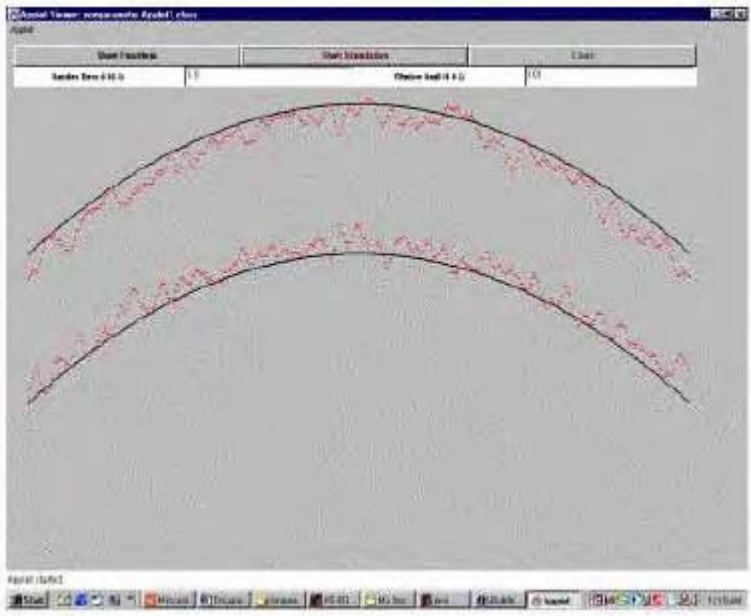


**Figure 1.** Java Applet simulation of the Nonparametric Estimation (M. Lu).

## References

[1] Bernstein, A.V. and Kagan, A.M. Estimation of failure probabilities from multiple observations. Abstracts, 2nd International Conference on Mathematical Methods in Reliability, Bordeaux, July 4–7, 2000, 195–198.

[2] Malyutov, M. and Lu, M. Robust Modification of the EM-algorithm for parametric MultiTrajectory estimation in noise and clutter This volume, 2003.

[3] Malyutov, M. and I. Tsitovich. Modeling multi-target estimation in noise and clutter. Proceedings, 12th European Simulation Symposium, ESS 2000 (Simulation in Industry) Sept. 28–30, Hamburg, Germany, Soc. for Computer Simulation, Delft, Netherlands, 598–600, 2000.

[4] Bernstein, A.V. Statistical analysis of multiple measurements with application to reliability. Abstracts, 2nd International Conference on Mathematical Methods in Reliability, Bordeaux, July 4–7, 2000, 191–194, 2000.

[5] Bernstein, A.V. Group regression parameter estimation. Izvestia of USSR Academy of Sciences, Technical Cybernetics: 137–141, 1973.

[6] Bernstein, A.V. Statistical analysis of random polynomials. Mathematical Methods of Statistics, 274–295, 1998.

[7] Korostelev, A.P. and A.B. Tsybakov. Minimax theory of image reconstruction, Lecture Notes in Statistics, Springer, N.Y., 1993.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Integrated Estimation and Guidance for Target Interception

Dany DIONNE [a] and Hannah MICHALSKA [b]
*Centre for Intelligent Machines, McGill University, 3480 University Street, Montreal, H3A 2A7, Canada,* [a] *ddionne@cim.mcgill.ca,* [b] *michalsk@cim.mcgill.ca*

**Abstract.** A novel integrated estimation and guidance design approach is presented in an attempt to find a suboptimal, but computationally effective procedure for the interception of maneuvering ballistic missiles. The approach incorporates banks of state estimators and guidance laws, a detector for the onset of the evader's maneuver, and a governor for on-line selection and adjustment of both the estimator and the guidance law employed. A new maneuver detector algorithm is introduced: the adaptive-$H_0$ GLR detector. As compared with previous approaches, simulation results show that the adaptive estimation and guidance approach leads to a significant reduction in the miss distance.

**Keywords.** Tracking, guidance, detection

## 1. Introduction

Over the last few years, considerable progress has been made toward a successful solution of the interception problem for non-maneuvering ballistic missiles. The interception of maneuvering ballistic missiles is still, however, an open problem whose difficulty arises from imperfect information about the evader's acceleration and from the fact that the pursuer does not have a significant maneuverability advantage over the evader. As the final miss distance is directly dependent on the accuracy of the evader's estimated acceleration, even small estimation errors can have a devastating effect on homing accuracy. Previous approaches to this joint tracking and guidance problem include: [1], in which the guidance law is designed to exhibit some degree of robustness with respect to the imperfect information structure; [2], which focuses on the design of a better state estimator; and [3], which uses a decision directed adaptive scheme for tracking of the maneuvering target.

The contributions of this paper can be summarized as follows. A novel integrated estimation and guidance approach is developed which draws on some ideas from [1] and [3]. The integration between estimation and guidance is performed using an adaptive incident detector together with a decision rule. The task of the decision rule, which acts as a "governor" in the homing scenario, is to select, on-line, a state estimator *and* a guidance law from pre-defined banks. The criteria used by the governor to select the estimator and the guidance law employ the estimates derived by a detection procedure whose task is to detect the onset of the evader's maneuver and to provide its characteristics. The detector employed here is of the generalized likelihood ratio (GLR) type, but is novel in the way in which it accommodates for the uncertainty in the target's acceleration.

## 2. Problem Statement

The guidance problem is the end-game interception (i.e. the last few seconds of the engagement) between a maneuvering ballistic missile (the evader) and an interceptor (the pursuer). For simplicity, the engagement geometry is assumed to be two dimensional, as shown in Figure 1. The model for the pursuit-evasion problem is originally nonlinear, but is linearized in the neighborhood of the initial line-of-sight resulting in a system description of the form:

$$\begin{aligned} x(k+1) &= Fx(k) + G_1 a_P^c(k) + G_2 z(k) + \omega(k) \\ y(k) &= Hx(k) + \eta(k) \end{aligned} \tag{1}$$

with $x \in \Re^n$, $\omega(k) \in N(0, Q_\omega)$, $a_P^c \in \Re^1$, $z \in \Re^1$, $y \in \Re^m$, and $\eta(k) \in N(0, Q_\eta)$. The state and measurement variables, $x$ and $y$, are random time series which are solutions of the above linear stochastic system. The known input, $a_P^c$, is the pursuer's command acceleration, and $\omega$ and $\eta$ are values of white Gaussian noise processes (in general non-stationary). The evader has no information on the state of the pursuer; its optimal evasive strategy is of the bang-bang type, using the maximum maneuvering potential, and with a single acceleration switch (at a random time instant) occurring over the time interval of the end-game engagement [4]. The jump process, $z$, is a random telegraph wave of an unknown magnitude that models the optimal evasive strategy of the evader. The pursuer has access only to measurements on the relative transversal separation, which is assumed to be the first component, $x_1$, of the state vector $x$, i.e. $H$ = [1 0 ⋯ 0]. It is further assumed that the pursuer and evader have dynamics characterized as a first order lag system, and the magnitude of

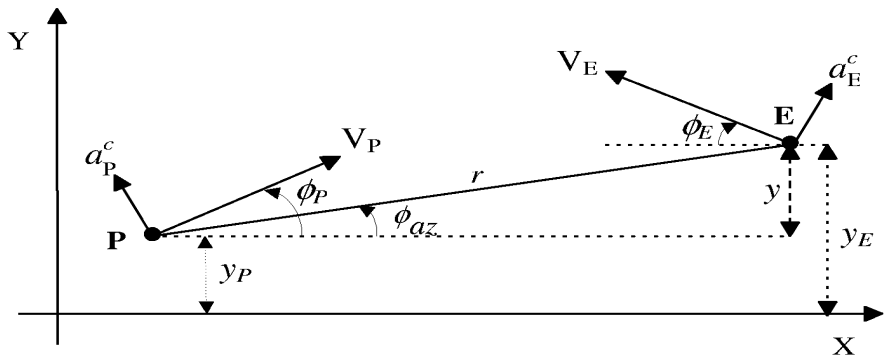



**Figure 1.** Planar engagement geometry. The symbol "P" denotes the pursuer and "E" the evader, $\phi_P$ and $\phi_E$ are the heading angles, and $\phi_{az}$ is the line-of-sight angle. The acceleration $a_P^c$, respectively $a_E^c$, is applied perpendicularly to the pursuer's velocity $V_P$, respectively $V_E$.

their accelerations are bounded by $(a_P^c)_{max}$, and $(a_E^c)_{max}$, respectively. For simplicity, it is assumed that the pursuer and evader travel at constant speeds over the time interval of the end-game engagement.

Example values for the system matrices $F$, $G_1$, $G_2$, and $H$, as well as the non-stationary covariance matrices $Q_\omega$ and $Q_\eta$, are given in many papers and books, e.g. [1] and [5].

The objective of the guidance problem is to steer $x_1$ to zero at a prescribed time instant $k = T_f$, where $T_f$ denotes the engagement horizon. Hence, the cost function, $J$, to be minimized, is:

$$J = E\left\{ \inf_{a_P^c \in A_P^c} \left| x_1(T_f) \right| \right\}, \quad A_P^c \triangleq \left\{ a_P^c \in \Re^1 \,|\, \left| a_P^c \right| \le (a_P^c)_{max} \right\} \tag{2}$$

with respect to all feasible control command acceleration strategies, $a_P^c$, of the pursuer. The quantity $x_1\,(T_f)$ is the miss distance while $A_P^c$ represents the set of feasible strategies.

The guidance problem, as defined by (1) and (2), is first viewed as a stochastic dual control problem for a jump-Gaussian linear system. The optimal solution to this stochastic dual control problem is computationally intractable. The optimal state estimator is infinite dimensional because the jump process $z$ is only *indirectly* observed through the output $y$ of system (1), cf. [6]. Moreover, the optimal closed form guidance laws are obtained based on the assumption of perfect (or delayed) information [7]. As such, they require the Certainty Equivalence Principle (CEP) to hold to maintain their optimality in the full stochastic dual control problem. For the above finite horizon problem, involving a non-Gaussian system with bounded controls and a non-quadratic cost function, the CEP does not hold in general and the optimal controller is a function of the full conditional probability distribution function (p.d.f.) of the state rather than a function of just the expected value of the system's state p.d.f., cf. Theorem 1 in [8].

Hence, an alternative, computationally feasible approach to the solution of (1) and (2) is adopted at the expense of global optimality. The proposed approach, a decision directed adaptive estimation and guidance, uses banks of state estimators and guidance laws interacting with an incident detector through an on-line governor. The details of the adaptive interaction between the detector, the estimator, and the guidance law are presented below. A novel adaptive maneuver detector is also introduced. Extensive simulation results show that the proposed approach is superior to previous attempts aimed at solving the same end-game guidance problem.

## 3. Integrated Adaptive Estimation and Guidance

In an attempt to "approximately separate" the problems of estimation and control, the full conditional p.d.f. of the state for the problem (1) and (2), is assumed to be "partitioned" into two modes. In the first mode, the conditional p.d.f. of the state is assumed to be dominated by a high probability of the evader performing a bang-bang maneuver. Corresponding to this mode, a first pair of an estimator/guidance law is designed. In the

second mode, the probability of an evasive maneuver is assumed to be low, so that the conditional p.d.f. is dominated by the (Gaussian) measurement and process noises. A second pair of estimator/guidance law is designed to serve (address) this situation. The second estimator/guidance law yields an improved performance over the first one, provided no evasive maneuver occurs. A governor, whose task is to identify, on-line, the mode of the system's state conditional p.d.f., and to select the estimator/guidance pair, is then devised in the form of a decision rule. The design of the governor requires a modal description of the system's state conditional p.d.f. as a function of the outputs from a detector. The detector's task is to identify, on-line, a jump in the evader's command acceleration. This integrated estimation and guidance approach is hence adaptive and hierarchical, see Figure 2 which describes its structure.

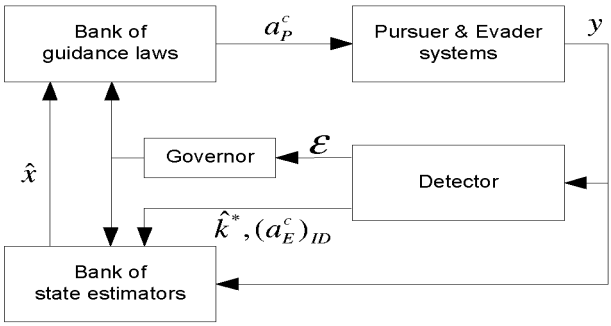



**Figure 2.** Integrated adaptive estimation and guidance.

### *3.1. General Description of the Integrated Procedure*

The principal task of the detector algorithm is to characterize the jump process $z$ by establishing: (a) the estimated time instant, denoted by $\hat{k}^\star$, at which a jump occurs in $z$, and (b) the estimated value of the jump, denoted by $(\hat{a}_E^c)_{ID}$. The detection procedure generates a stopping time, $k_D$, also referred to as the *detection time*, at which an incident is first detected. The relation between $\hat{k}^\star$, $k_D$, and $k$ is clearly given by $\hat{k}^\star < k_D \leq k$. Whenever the detector produces a new value of $k_D$ (i.e. $k_D = k$), the governor is activated to select the most suitable state estimator and guidance law from the provided banks. For further use, it is convenient to define an auxiliary binary random process $E(k)$, $k \geq 0$ that serves as an indicator of a jump in $z$ : i.e. $E(k) = 0$ if no jump is detected, and $E(k) = 1$ if a jump is detected.

The selection procedure for the state estimator consists in the selection of an appropriate value for the process noise covariance of the acceleration model of the evader (see next section). The selection of the guidance law is made in such a way as to decrease the expected worst case miss distance.

### 3.2. The Bank of State Estimators

All state estimators in the bank are assumed to have the same general form which is that of a Kalman filter for the following augmented system:

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{F}\tilde{x}(k) + \tilde{G}_1 a_p^c(k) + \tilde{\omega}(k) \quad && \tilde{x} \in \Re^{n+1}, \tilde{\omega} \in N\left(0, \tilde{Q}_\omega\right) \\ & && a_p^c \in \Re^1 : \text{ a known input} \\ \tilde{y}(k) &= \tilde{H}\tilde{x}(k) + \tilde{\eta}(k) && \tilde{y} \in \Re^m, \tilde{\eta} \in N\left(0, \tilde{Q}_\eta\right) \end{aligned} \tag{3}$$

The augmented state and observation vectors are:

$$\begin{aligned} \tilde{x}(k) &= \left[ x^T(k) \quad z^T(k) \right]^T, \quad x \in \Re^n, \quad z \in \Re^1 \\ \tilde{y}(k) &= y(k), \qquad\qquad y \in \Re^m \end{aligned} \tag{4}$$

The augmentations (3) and (4) consist in introducing a Wiener process model as a finite dimensional approximation to the unknown jump process:

$$\frac{dz}{dt} = dw_a, \;\; dw_a(t) \in N(0, Q_a) \tag{5}$$

The augmented matrices $\tilde{F}, \tilde{G}_1, \tilde{H}, \tilde{Q}_\omega$, and $\tilde{Q}_\eta$ are obtained from the matrices of (1) and the approximation (5), cf. [9], p. 264. Such a suboptimal estimator tracks a jump process if the value of $Q_a$ is sufficiently large but, as a trade-off, the rejection of the Gaussian noises $\tilde{\omega}$ and $\tilde{\eta}$ degrades. To improve the estimation performance, it would hence be desirable to adjust the value of $Q_a$ in accordance with the mode of the conditional p.d.f.

For the sake of simplicity, it is assumed that $Q_a$ is adjusted on a discrete basis and can take only two values: $Q_{a1}$ and $Q_{a2}$. Such an assumption gives rise to a bank of estimators consisting of two members referred to as $E_1$ and $E_2$, respectively.

The covariance $Q_{a1}$ is set to have a larger value than that of $Q_{a2}$. By design, the need to employ estimator $E_1$ arises at times where the uncertainty in the system is believed to be dominated by the unknown jump process. For our application, a suitable value for $Q_{a1}$ is obtained following the formula recommended by [10]:

$$Q_{a1} = 4\frac{(a_E^c)_{max}^2}{T_f} \tag{6}$$

The estimator $E_2$ is used for time instants such that the system uncertainty is dominated by the Gaussian noise processes rather than the jump process. The value of the covariance $Q_{a2}$ is selected as a trade-off between: (a) optimal noise rejection (for which

$Q_{a2}$ should be set to zero), and (b) to provide the filter with a sufficient bandwidth to compensate for possible errors in the estimate of the value of the last jump (for which $Q_{a2}$ must be sufficiently large). A reasonable trade-off is given by:

$$Q_{a2} = \frac{4}{25}\frac{(a_E^c)_{max}^2}{T_f} \tag{7}$$

The switch from $E_1$ to $E_2$ requires an adequate initialization of the newly selected estimator. Let $W(k)$ be the Kalman gain associated with the augmented model (3). At the detection time instant, i.e. at $k = k_D$:

$$\hat{x}(k \mid k)_{E2} = \hat{x}(k \mid k)_{E1} + \delta\hat{x}(k \mid k), \quad k = k_D \tag{8}$$

where the correction term $\delta\hat{x}(k \mid k)$ is calculated recursively according to:

$$\delta\hat{x}(l \mid l) = (I - W(l)H)F\delta\hat{x}(l-1 \mid l-1), \quad l \in [\hat{k}^\star + 1, \cdots, k] \tag{9}$$

with

$$\delta\hat{x}(l \mid l) = \begin{bmatrix} 0 & \cdots & 0 & \delta\hat{a} \end{bmatrix}^T, \quad \delta\hat{a} \triangleq (\hat{a}_E^c)_{ID} - \hat{a}_E^c(l)_{E1} \tag{10}$$

In the above, $\hat{a}_E^c(l)_{E1}$ is the estimated value of the evader commanded acceleration as estimated by the estimator $E_1$ at time instant $l$. Such a correction compensates the current state estimate for the mismatch between the identified past acceleration history, $(\hat{a}_E^c)_{ID}$, and the one estimated by $E_1$. For simplicity, the initialization of the state covariance matrix of $E_2$ is:

$$P_{E2}(k \mid k) = P_{E1}(k \mid k) \tag{11}$$

where $P_{E2}$ and $P_{E1}$ denotes the state covariance matrices of $E_2$ and $E_1$.

The reason for which all the state estimators in the bank incorporate a stochastic model of the unknown jump process despite that the same jump process is identified in parallel by the incident detector procedure deserves some explanation. This is done because: (a) a time delay is present in the detection of an incident, (b) there is an inherent estimation error in the estimates $(\hat{a}_E^c)_{ID}$ and $\hat{k}^\star$, and (c) there is a possibility of a false detection.

### 3.3. The Bank of Guidance Laws

As in the case of the estimators, the bank of guidance laws is limited to two members, referred to as DGL/0 and DGL/1. These laws have been derived by [11] and [12] as optimal solutions to a perfect information zero-sum deterministic pursuit-evasion game:

$$\tilde{J} = \inf_{a_P^c \in A_P^c} \sup_{a_E^c \in A_E^c} \left| x^1(T_f) \right|, \quad \begin{array}{l} A_P^c \triangleq \{a_P^c \in \Re \,|\, |a_P^c| \le (a_P^c)_{max}\} \\ A_E^c \triangleq \{a_E^c \in \Re \,|\, |a_E^c| \le (a_E^c)_{max}\} \end{array} \tag{12}$$

where $A_P^c$ and $A_E^c$ are the feasible sets for the pursuer and evader strategies, respectively. The model of the game is similar to (1) except that the *stochastic* jump process is replaced by the *deterministic* process $a_E^c$ (i.e. $z(k) = a_E^c(k)$) and the Gaussian noises $\omega$ and $\eta$ are removed. The latter system is deterministic and, under the assumption of perfect information on the system state, the optimal game theoretic control laws, DGL/0 and DGL/1, both take the form:

$$a_P^c = (a_P^c)_{max} \text{sign}(\text{ZEM}) \tag{13}$$

where the zero effort miss (ZEM) is the expected miss distance assuming that the pursuer applies a zero acceleration policy after the current time instant. The guidance laws DGL/0 and DGL/1 differ in the way in which the ZEM is computed. The DGL/0 law assumes that the actual target acceleration is unknown while the DGL/1 law assumes the opposite: that it is perfectly known. The DGL/1 law can achieve miss distances much smaller than the DGL/0 provided that there is a sufficient certainty about the current value of the evader's acceleration.

### 3.4. The Governor

Summarizing the above discussion, whenever $E(k) = 0$ the value of the actual evader's command acceleration is uncertain, while the actual evader's command acceleration is considered known whenever $E(k) = 1$. Thus, the governor selects, on-line, the state estimator and the guidance law as follows:

$$(E, DGL) = \begin{cases} (E_1, DGL/0) & \text{for } E(k) = 0 \\ (E_2, DGL/1) & \text{for } E(k) = 1 \end{cases} \tag{14}$$

## 4. The Incident Detector

The incident detector detects the occurrence of a jump in the unknown input of the linear model (1). The selected GLR based incident detector provides also an estimate of the onset time of the jump, $k^\star$, and of the value of the unknown input $z$ after the

jump. The standard GLR incident detector, cf. [13], is briefly described below. Next, the adaptive-$H_0$ GLR detector, a novel modification of the standard GLR incident detector, is presented. The purpose of the adaptive-$H_0$ GLR detector is to provide better robustness against the uncertain value of the evader acceleration *before* the jump.

### *4.1. Standard GLR Detector*

The idea behind the GLR detector is to test a number of pre-specified hypotheses concerning the value of the onset time instant of a possible maneuver:

$$H_0 : k^\star = -\infty \tag{15}$$

$$H_i : k^\star = k_i, \quad i \in \{k-w, \cdots, k-1\} \tag{16}$$

Hypothesis $H_0$ is interpreted as the absence of any jumps in $z$, while hypothesis $H_\mathrm{i}$ corresponds to the occurrence of a jump at time instant $i$. The variable $w$ is the so-called "sliding window width" which is introduced to render the computational scheme finite by limiting the number of possible hypotheses. To evaluate the likelihood of the individual hypotheses, a reference Kalman filter is implemented for system (1) based on the assumption that hypothesis $H_0$ is true:

$$\hat{x}(k+1 \mid k)_{H_0} = F\hat{x}(k \mid k)_{H_0} + G_1 a_k^c + G_2 a^{H_0} \tag{17}$$

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1)_{H_0} + K(k)\gamma(k)_{H_0} \tag{18}$$

where the reference acceleration, $a^{H_0}$, is the assumed value of the evader's command acceleration before the jump and $K(k)$ is the Kalman filter gain.

The measurement residual, $\gamma(k)_{H_0}$, is given by:

$$\gamma(k)_{H_0} = y(k) - H\hat{x}(k \mid k-1)_{H_0} \tag{19}$$

A "signature" develops in the sequence of residuals $\gamma(k)_{H_0}$ whenever a difference exists between the behavior of the evader and the behavior assumed by the hypothesis $H_0$. Each hypothesis $H_\mathrm{i}$ is associated with a corresponding signature. Assuming that the value of $z(k)$ *before* the jump is indeed equal to $a^{H_0}$, the residual $\gamma(k)_{H_0}$ can be conveniently re-expressed in terms of a signature, $\nu\rho(k,k^\star)$, and an additive white noise $\gamma_1(k)$:

$$\gamma(k)_{H_0} = \nu\rho(k,k^\star) + \gamma_1(k), \quad \gamma_1(k) \in N(0, V(k)) \tag{20}$$

where $V(k)$ is the innovation covariance, as calculated by the reference Kalman filter, and where $\nu$ is a scaling factor given by:

$$\nu = \frac{z(j) - a^{H_0}}{a^{\rho}} = \text{const. for all } j \in \left\{k^{\star}, \cdots, k\right\} \tag{21}$$

where $z(j)$ is the value of the jump process after the jump, The expression $\rho(k, k^{\star})$ denotes a *normalized* signature; it corresponds to the value of the bias in the residual $\gamma(k)_{H_0}$ assuming that a jump of a normalized magnitude $a^{\rho}$ occurred at the past time instant $k^{\star}$. After scaling, the actual signature which matches the true value of the jump is given by $\nu\rho(k, k^{\star})$.

The normalized signature of hypothesis $H_i$ corresponds to:

$$\rho(k,i) = H\Gamma(k,i), i \in \left[k - w, k - 1\right] \tag{22}$$

$$\Gamma(k,i) = \hat{x}(k \mid k-1)_{H_0} - \hat{x}(k \mid k-1)_{H_i} \tag{23}$$

where the state prediction $\hat{x}(k \mid k-1)_{H_i}$ is computed similarly as given by (17), but using the acceleration history of hypothesis $H_{\mathrm{i}}$ instead of $H_0$.

The expression (23) requires calculating the outputs of as many filters as there are hypotheses. However, $\Gamma(k,i)$ can also be calculated recursively using only the (single) reference Kalman filter:

$$\Gamma(l,i) = G_2 a^{\rho} + \Phi(l-1)\Gamma(l-1,i), \quad \Gamma(i,i) = 0 \tag{24}$$

$$\Phi(l-1) \triangleq F[I - K(l-1)H], \quad l = i+1, \cdots, k \tag{25}$$

A sequential probability ratio test, known as the GLR test, is next performed on the residual sequence $\gamma(k)_{H_0}$ to establish the validity of each of the hypotheses $H_{\mathrm{i}}$. The GLR test involves the computation of the log-likelihood ratio, $l(k,i)$, of each hypothesis $H_{\mathrm{i}}$. The log-likelihood ratios are recursively computed as follow:

$$l(k,i) = \frac{1}{2}\frac{d^2(k,i)}{J(k,i)}, \quad i \in \{k - w, \cdots, k-1\} \tag{26}$$

where the signature correlation, $d(k,i)$, and the Kullback-Leibler divergence, $J(k,i)$, are calculated for each hypothesis according to:

$$d(l,i) = d(l-1,i) + \rho^T(l,i)V^{-1}(l)\gamma(l)_{H_0}, \quad d(i\,|\,i) = 0 \tag{27}$$

$$J(l,i) = J(l-1,i) + \rho^T(l,i)V^{-1}(l)\rho(k,i), \quad J(i\,|\,i) = 0 \tag{28}$$

$$l = i+1,\cdots,k \tag{29}$$

Assuming that the time instant of the jump is known and that no prior information about the value of the jump is available, the signature correlation (27) can be interpreted as a matched filter, or a least squares estimate of the value of the jump, cf. [13]. The error covariance of the scaling factor estimate $\hat{\nu}$ is calculated as $(a^{\rho})^{-2}J^{-1}(k,i)$.

The verification of the hypotheses requires the selection of a suitable threshold, $h$. The value of this threshold is selected as a trade-off between the probability of a false alarm, $\alpha$, and the probability of a truthful detection.

For a jump-linear Gaussian system such as (1), the statistical distribution of the probability of a false alarm is given by a central $\chi^2$ distribution with one degree of freedom, cf. [13]. The value of $h$ solves the equation:

$$\alpha = \int_h^{\infty} \chi^2(u)\,du \tag{30}$$

The GLR test given below allows to accept one of the hypotheses $H_{\mathrm{i}}$ or $H_0$ according to:

$$\{l(k,i^{\star}(k)) \le h\} \Rightarrow \{H_0 \text{ is true}\} \tag{31}$$

$$\{l(k,i^{\star}(k)) > h\} \Rightarrow \{H_{i^{\star}(k)} \text{ is true}\} \tag{32}$$

where $i^{\star}(k)$ is calculated as the index maximizing the log-likelihood ratios:

$$i^{\star}(k) = \operatorname{argmax}\{l(k,i)\,|\,i \in \{k-w,\cdots,k-1\}\} \tag{33}$$

The maximum likelihood estimates of the jump time instant, $\hat{k}^{\star}$, and of the scaling factor, $\hat{\nu}$, are obtained as follows:

$$\hat{k}^{\star} = k_{i^{\star}(k)} \tag{34}$$

$$\hat{\nu} = \frac{d(k,\hat{k}^*)}{J(k,\hat{k}^*)} \tag{35}$$

The detection time, $k_D$, and the maximum likelihood estimate, $(\hat{a}_E^c)_{ID}$, of the actual value of the jump, are defined as:

$$k_D = \text{argmin}\left\{ j \,|\, j \in \{1,\cdots,k\}, l(j,i^*(j)) > h \right\} \tag{36}$$

$$\left(\hat{a}_E^c\right)_{ID} = \begin{cases} a^{H_0} & \text{if } H_0 \text{ is true} \\ a^{H_0} + \hat{\nu} a^{\rho} & \text{otherwise} \end{cases} \tag{37}$$

Whenever $k_D > 0$, the binary event variable $E(k)$ changes its state from zero to $E(k) = 1$.

As the sliding window determining the hypotheses move forward with the current time, it eventually happens that the hypothesis about an already detected jump is left behind. When this last event occurs, [3] and [13] suggest to reinitialize the detector by updating the reference value $a^{H_0}$ and by resetting to zero all of the signatures and likelihood ratios. The state estimate produced by the reference Kalman filter and its covariance are then reinitialized:

$$\hat{x}(k \,|\, k)_{H_0}^{new} = \hat{x}(k \,|\, k)_{H_0}^{old} + \hat{\nu} \Upsilon(k, k-w) \tag{38}$$

$$P(k \,|\, k)_{H_0}^{new} = P(k \,|\, k)_{H_0}^{old} + \Upsilon(k,k-w)\left[ (a^{\rho})^2 J^{-1}(k,k-w) \right] \Upsilon^T(k,k-w) \tag{39}$$

$$\Upsilon(k,k-w) \triangleq (I - K(k)H)\Gamma(k,k-w) \tag{40}$$

where the superscripts $^{old}$ and $^{new}$ distinguish the variables before and after reinitialization, respectively.

### *4.2. The Adaptive-$H_0$ GLR Algorithm*

The standard GLR detector requires the assumption that the value of the reference acceleration $a^{H_0}$ is indeed the true value of $z$ before the jump. In realistic maneuver detection scenarios, this is, however, seldom the case. To remedy this situation, a new algorithm is developed by *enabling the reference acceleration of hypothesis $H_0$ to be changed on-line*.

To this end, an extended set of hypotheses is introduced by augmenting the original set of hypotheses $\{H_0, H_{k-w}, \cdots, H_{k-1}\}$ of (17) by a hypothesis denoted $H_k$. Also, to avoid the loss of information incurred by the standard GLR detector which occurs whenever the signatures and the likelihood ratios are reset to zero, a second hypothesis, $H_\psi$, is added to the set, in order to keep track of any detected jump which is left out from the current hypotheses window. The two additional hypotheses are defined as follows:

$$H_k : \Delta \triangleq z(j) - a^{H_0} = \text{const.} \neq 0, \text{for all } j \in \{1, \cdots, k\} \tag{41}$$

$$H_\psi : k^\star = k - w - \psi, \text{for some } \psi \in N \tag{42}$$

where $\Delta$ is the mismatch between the reference acceleration $a^{H_0}$ and the true acceleration $z(j)$ before the jump. The adaptive- $H_0$ GLR algorithm also includes a hard constraint on the value of $z(k)$ if such an information is available, see below.

The additional normalized signatures, $\rho(k,k)$ and $\rho(k,\psi)$, are computed as before using (22), (24), and (25), but in which the recursions generating $\Gamma(l,k)$ and $\Gamma(l,\psi)$ are initialized with $l = 0$ and $l = k - w - \psi$ (i.e. $\Gamma(0,k) = 0$ and $\Gamma(k-w-\psi,\psi) = 0$), respectively. Similarly, the log-likelihood ratios, $l(k,k)$ and $l(k,\psi)$, are calculated using (26), (27) and (28), but with the recursions $d(l,k)$ and $d(l,\psi)$ initialized with $l = 0$ and $l = k - w - \psi$ (i.e. $d(0,k) = 0$ and $d(k-w-\psi,\psi) = 0$), and with the recursions $J(l,k)$ and $J(l,\psi)$ initialized similarly (i.e. $J(0,k) = 0$ and $J(k-w-\psi,\psi) = 0$). The threshold $h$ and the GLR test remain unchanged. However, the index $i^\star$ is now obtained from:

$$i^\star(k) = \text{argmax}\left\{ l(k,i) \,|\, i \in S^k \right\} \tag{43}$$

$$S^k \triangleq \left\{ j \mid j \in \{\psi, k-w, \cdots, k\} \cap \hat{z}_j(k) \mid < z_{max} \right\} \tag{44}$$

$$\hat{z}_j(k) \triangleq a^{H_0} + \frac{d(k,j)}{J(k,j)} a^\rho \tag{45}$$

The introduction of the constrained set, $S^k$, allows to include a priori information on the maximum value, $z_{max}$, of the jump process.

The additional hypotheses $H_k$ and $H_\psi$ require modifications in the formulae to calculate the detection time and the maximum likelihood estimates. The maximum likelihood estimates, $\hat{k}^\star$ and $\hat{\nu}$, as given by formulae (34) and (35), are now only

defined when $i^\star(k) \neq k$. When $i^\star(k) = k$, hypothesis $H_k$ is considered true and it is understood that $\hat{k}^\star = 0$ and that the maximum likelihood estimate, $\hat{\Delta}(k)$, of the mismatch is:

$$\hat{\Delta}(k) = \hat{v}a^\rho : \text{ whenever } i^\star(k) = k \tag{46}$$

The detection time and the maximum likelihood estimate of the value of the jump are now obtained as follows:

$$k_D = \operatorname{argmin}\left\{ j \mid j \in \{1,\cdots,k\}, l(j, i^\star(j)) > h, i^\star(j) \neq k \right\} \tag{47}$$

$$\left(\hat{a}_E^c\right)_{ID} = \begin{cases} a^{H_0} & \text{if } H_0 \text{ is true} \\ a^{H_0} + \hat{v}a^\rho & \text{if } H_0 \text{ and } H_k \text{ are false} \end{cases} \tag{48}$$

Whenever $k_D > 0$ and $H_k$ is false, the binary event variable $E(k)$ changes its state from zero to $E(k) = 1$.

When the GLR test indicates that the hypothesis $H_k$ is true, it is understood that the value of $z$ before the jump and the value of the reference acceleration $a^{H_0}$ are mismatched and the latter is adapted according to:

$$a_{new}^{H_0} = a_{old}^{H_0} + \hat{\Delta}(k), \quad \text{for } l = 0,...,k \tag{49}$$

For consistency with the new value of the reference acceleration, the state estimate of the reference filter and its covariance are adapted as follows:

$$\hat{x}(k \mid k)_{H0}^{new} = \hat{x}(k \mid k)_{H0}^{old} + (I - K(k)H)\hat{v}\rho(k \mid k) \tag{50}$$

$$P(k \mid k)_{H0}^{new} = P(k \mid k)_{H0}^{old} + \Upsilon(k,k)\left[(a^\rho)^2 J^{-1}(k,k)\right]\Upsilon^T(k,k) \tag{51}$$

It is also required to adapt the log-likelihood ratios to the new value of $a^{H_0}$. The appropriate correction is performed recursively through the signature correlations, according to:

$$d(k,i)^{new} = d(k,i)^{old} - \Delta_d(k,i)\hat{\Delta}(k) \tag{52}$$

where the normalized correction, $\Delta_d(k,i)$, is given by:

$$\Delta_d(k,i) = \Delta_d(k-1,i) + \rho^T(k,i)V^{-1}(k)\rho(k,k) \tag{53}$$

The above formulae for the correction of the signature correlations are obtained by deriving two sets of equations for the signature correlations. One set of equations is obtained using the mismatched $a^{H_0}$ and the second set is obtained using the corrected value instead of $a^{H_0}$. The correction (53) follows by comparison of the two sets of equations.

## 5. Application to Terminal Guidance

The homing performance of the decision directed adaptive estimation and guidance algorithm, using the adaptive-$H_0$ detector, is assessed through Monte Carlo simulations. One hundred Monte Carlo simulations are performed, each characterized by a different selection of an onset time instant for the evader's bang-bang maneuver. Each Monte Carlo simulation repeats the engagement 2 000 times. Each repetition has a different noise realization. The simulations use the nonlinear dynamical model of a pursuit-evasion engagement from [1].

**Table 1.** Simulation parameters

| | | | |
|---|---|---|---|
| pursuer velocity | $V_P$ = 2300 m/s | evader velocity | $V_E$ = 2700 m/s |
| pursuer max. acc. | $(a_P^C)_{max}$ = 30 g | evader max. acc. | $(a_E^C)_{max}$ = 15 g |
| pursuer f.o.l.t.c. | $\tau_P$ = 0.2 s | evader f.o.l.t.c. | $\tau_E$ = 0.2 s |
| initial range | $X_0$ = 20 000 m | measurement freq. | $f$ = 100 Hz |
| false alarm prob. | $\alpha$ = 0.001 | ang. noise std. dev. | $\sigma$ = 0.1 mrad |
| number of hypoth. | $w$ = 70 | max. jump proc. mag. | $z_{max}$ = 100 g |

The simulation parameters are displayed in Table 1 where "f.o.l.t.c" stands for "first-order lag time constant". The measurement noise covariance is obtained by linearization of the angular measurement noise:

$$Q_\eta(k) = (R(k) \times \sigma)^2 \tag{54}$$

where *R(k)* is the distance between the evader and the pursuer.

The homing performance is evaluated using the Single Shot Kill (SSK) probability, i.e. the probability of a successful interception, and the pursuer's lethal radius, i.e. the maximum acceptable miss distance for a successful interception. It is desirable to minimize the pursuer's lethal radius for a given SSK probability and against all possible onset time instants of the evader's maneuver. It is convenient to define, $t^\star$, the absolute onset time instant of the evader's maneuver as $t^\star \triangleq k^\star / f$.

Figure 3 shows the required lethal radius of the pursuer in order to achieve SSK = 0.95 as a function of the onset time instant of the evader's maneuver.

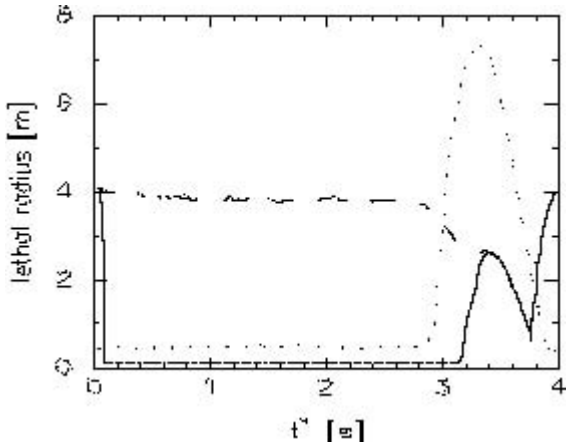



**Figure 3.** Required pursuer's lethal radius to achieve SSK=0.95 versus the onset time instant of the evader's maneuver, $t^{\star}$. Three curves are compared: (1) dotted line – estimator $E_1$ and DGL/1 law, (2) dashed line – estimator $E_1$ and DGL/0 law, and (3) solid line – integrated adaptive estimator and guidance law.

The integrated adaptive estimator and guidance algorithm delivers miss distances which are always smaller or equal than those corresponding to the pair ($E_1$, DGL/0) and, on average, smaller than those corresponding to the pair ($E_1$, DGL/1). At $t^{\star} \approx 0$ s, the relatively large miss distances observe for the adaptive approach can be explained by the fact that the jump fails to be detected due to insufficient information to distinguish between a jump and a mismatch in the value of $a^{H_0}$. As a consequence, the adaptive scheme employs the pair ($E_1$, DGL/0) at all times. In the case of a jump occurring at $t^{\star} \approx [3.2, 4]$ s, the homing performance of the adaptive scheme degrades because there is not sufficient time left in the engagement to either detect the jump or, after detection, to bring the pursuer's trajectory back onto a "perfect" interception course.

## 6. Conclusions

The integrated adaptive estimation and guidance approach, as compared with the conventional approaches, results in a significant improvement in the homing performance against a maneuvering target. This is attributed primarily to the fact that the novel approach does not assume the same p.d.f. to characterize the jump process before and after the jump (the two different p.d.f. are referred to as p.d.f. "modes"). A pair of estimator/guidance law is further optimized for each of the p.d.f. modes which are identified on-line using a detector.

The strength of the approach is that the estimators or the guidance laws in the banks can easily be replaced by more sophisticated procedures which can result in further performance improvement. For example, in the first p.d.f. mode, an IMM filter could be used instead of a single Kalman filter without introducing any modifications to the remaining modules of the approach.

Experiments show that the approach is robust against model-system errors as, in simulations, the original nonlinear model is used while the estimator/guidance laws where design for its linear approximation.

For simplicity of exposition, only a single evader's maneuver was allowed which suggests a simple dual information structure in which the stochastic behavior of the system is either dominated by the jump process, or else by the system Gaussian noise. The approach can be generalized to multiple jumps.

## References

[1] Shinar, J., & Shima, T., "Non-orthodox guidance law development approach for the interception of maneuvering anti-surface missiles," Proc. of the 7 IEEE Mediterranean Conference on Control & Automation, Haifa, Israel, 2000, pp. 651–661.
[2] Oshman, Y., Shinar, J., & Weizman, S.A., "Using a multiple-model adaptive estimator in a random evasion missile/aircraft encounter," Journal of Guidance, Control, and Dynamics, v. 24, n. 6, 2001, pp. 1176–1186.
[3] Dowdle, J.R., Willsky, A.S., & Gully, S.W., "Nonlinear generalized likelihood ratio algorithms for maneuver detection and estimation," Proc. of the ACC, 1982, pp. 985–987.
[4] Shinar, J., & Steinberg, D., "Analysis of optimal evasive maneuvers based on a linearized two-dimensional model," Journal of Aircraft, v. 14, 1977, pp. 795–802.
[5] Zarchan, P., "Tactical and strategic missile guidance," Progress in Astronautics and Aeronautics, vol. 176, AIAA Inc, Washington D.C., ISBN: 1-56347-254-6, 1997.
[6] Kushner, H.J., "Robustness and convergence of approximations to nonlinear filters for jump-diffusions," Computational and Applied Math., v. 16, 1997, pp. 153–183.
[7] Shinar, J., & Glizer, V.Y., "Solution of a delayed information linear pursuit-evasion game with bounded controls," Internat. Game Theory Review, vol. 1, n. 3& 4, 1999, pp. 197–217.
[8] Striebel, C., "Sufficient statistics in the optimum control of stochastic systems," Journal of Mathematical Analysis and Applications, vol. 12, 1965, pp. 576–592.
[9] Bar-Shalom, Y., Li, X.R., & Kirubarajan, T. "Estimation with applications to tracking and navigation," John Wiley & Sons, ISBN: 0-471-41655-X, 2001.
[10] Zarchan, P., "Representation of realistic evasive maneuvers by the use of shaping filters," Journal of Guidance and Control, Vol. 2, 1979, pp. 290–295.
[11] Gutman, S., "On optimal guidance for homing missiles," Journal of Guidance and Control, v. 3, n. 4, pp. 296–300.
[12] Shinar, J., "Solution techniques for realistic pursuit-evasion games," Advances in Control and Dynamic Systems, Ed.: C.T. Leondes, vol. 17, Academic Press, NY, 1981, pp. 63–124.
[13] Willsky, A.S., & Johns, H.L., "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," IEEE transactions on Automatic Control, vol. AC-21, 1976, pp. 108–112.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# An Adaptive, Variable Structure, Multiple Model Estimator

Hannah MICHALSKA [a] and Dany DIONNE [b]
*Centre for Intelligent Machines, McGill University, 3480 University Street, Montreal, H3A 2A7, Canada,* [a] *michalsk@cim.mcgill.ca,* [b] *ddionne@cim.mcgill.ca*

**Abstract.** The paper presents a novel algorithm for faster and more reliable state estimation of maneuvering targets. The system model is assumed linear and the maneuver is modeled as an unknown jump process. Exact solution of the optimal state estimation problem would require the construction of an adaptive procedure which, in the presence of system dynamics, would not lead to a finite, recursive filtering scheme. In quest for computational feasibility, the estimation problem is solved in a sub-optimal way, employing a combination of a generalized likelihood ratio algorithm and a suitably modified multiple model filter.

**Keywords.** Nonlinear filtering, switching linear systems, jump processes

## 1. Introduction

For the purpose of improving aircraft speed and maneuverability, it is of utmost importance that modern airborne surveillance systems be equipped with fast and reliable state estimation algorithms. The algorithms typically used include various versions of the interacting multiple model filters because the models of the target motions are seldom known with satisfactory precision and also often change with time as the targets perform maneuvers. Such maneuvers are predominantly rapid and can often be modeled as jump processes with some partially known characteristics. An example of this type of situation is provided by any missile or aircraft interception scenario. The existing IMM filters are not well suited to these situations and are often known to provide too slow filter responses. On the other hand, there is a body of literature currently emerging which deals with the construction of maneuver detectors. The fastest and most reliable detector algorithms known to date are those based on multi-hypothesis testing via comparison of the Generalized Likelihood Ratios (GLR) for the hypotheses constructed, see [1]. One such GLR detector has been recently developed by Willsky et al. in [2], but requires a priori, exact knowledge of the state of the system before a maneuver. Previous attempts for the design of multiple model estimators for fast maneuvering targets include [3] and [4]; with a comparison of existing approaches provided in [5]. The references mentioned above, all share the same type of model of a maneuvering target which is that of a jump process. An observation window is constructed in which a target is assumed to exhibit a single maneuver. The window is of fixed length in time and slides forward with the time instant at which the state estimate is required. The maneuver (usually identified with a jump in the acceleration of the target occurring at a specific time instant within the window) represents a model of target motion. In [3], the

magnitude of the maneuver (jump) is assumed to be known, or else to be a member of a known finite set of discrete levels. The total number of required number of models in that filter is usually very large as each of the models is assumed to describe not only the magnitude of the jump, but also the time instant at which it occurs. The algorithm in [3] is therefore a standard multiple model filter (without model interaction) with a large number of fixed models.

The filtering algorithm presented in [4] is similar to a GLR type algorithm and reduces the number of models by way of on-line estimation (adaptation) of the jump magnitude. However, it does not permit to account for jumps outside of the sliding observation window. The algorithm presented in [6] is somewhat similar to that of [4], but employs the proper GLR test for hypothesis testing. The GLR test is used to first "identify" a jump instant while an additional adaptation sub-algorithm permits to update its actual magnitude. The main difference between the algorithms presented in [6] and that of [4] is the introduction of a variable hypothesis about the reference level before the jump (hypothesis $H_0$).

In the above context, the novel algorithm presented here, the GLR-MM estimator, has the following advantages:

- it constitutes the first attempt to combine the advantages of the GLR detector with a multiple model (MM) type filter;
- it allows for on-line identification of the jump magnitude;
- it allows for incorporating additional probabilistic information into the transition matrix of the Markov chain which describes model switching (an example of such a case would be a scenario in which the jump process is partly known, and governed by, say, a Poisson p.d.f.).

## 2. Problem Statement

For simplicity of exposition, the new state estimation algorithm will be presented with a reference to the following simplified tracking problem of a target maneuvering in the plane. Specifically, it is assumed that the true system model of the maneuvering target is embedded in the family of the following linear stochastic systems:

$$\begin{aligned} x(k+1) &= Fx(k) + G_1 u(k) + G_2 z(k) + \omega(k) \\ y(k) &= Hx(k) + \eta(k) \end{aligned} \tag{1}$$

with $x \in \Re^n$, $u \in \Re^1$, $z \in \Re^1$, $y \in \Re^m$, $\omega(k) \sim N(0, Q_\omega)$, and $\eta(k) \sim N(0, Q_\eta)$. The state and measurement variables, $x$ and $y$, are random time series which are solutions of the above linear stochastic system, $u$ is a known external input, and $z$ is a stochastic variable subject to abrupt changes. The process $z$ can be interpreted as the unknown evader's command acceleration. The time interval between abrupt changes is assumed to be bounded from below by $w^\star$. For simplicity of the exposition, the process $z$ is further restricted to the special and important class of a jump process of unknown and, possibly, variable magnitude. For the filtering problem to be meaningful, it is assumed that the system (1) is observable.

The objective is to develop a fast, and statistically reliable, multiple model state estimation filter for a system which is a member of the family above. Since the jump process is not known a priori, the general estimation problem, as stated, is that of filtering a hidden jump process. In a predominant number of cases in which the jump process can only be observed through the linear system dynamics, the corresponding optimal filtering problem is infinite dimensional and computationally intractable, cf. [7].

For further use, let $Y^k$ denotes the $\sigma$-field algebra generated by the measurements:

$$Y^k = \sigma\{y_s : 0 < s \leq k\} \tag{2}$$

## 3. The GLR Multiple Model Estimator

The optimal Bayesian estimator for the hybrid system (1) is a NP-complete problem involving an exponentially growing tree of hypotheses, and, as such, it cannot be implemented in real time, cf. [8]. Practical, suboptimal, Bayesian based, estimators have to rely on certain hypothesis management techniques to keep the number of hypotheses within a certain limit. The adaptive, variable structure, GLR multiple model estimator delivers the estimates of the hybrid system (1) by pruning the unlikely models from the hypothesis tree of the optimal Bayesian filter. At each time instant $k$, the pruning involves the following steps:

1. over the time interval $[0,\cdots,k-w-1]$, a single hypothesis is retained which corresponds to the maximum likelihood ratio hypothesis;
2. over the time interval $[k-w,\cdots,k]$, a set, $S^k$, of hypotheses is constructed. Each hypothesis in the set corresponds to a different history for the process $z$ over this time interval $[k-w,\cdots,k]$.

In this paper, the length of the sliding window, $w$, which is used for the construction of the set $S^k$ is restricted to $w < w^\star$. Setting $w < w^\star$ is not essential. However, allowing $w \geq w^\star$ requires considering hypotheses with more than a single jump in $S^k$. The latter requirement about several jumps is numerically cumbersome since the number of hypotheses in the set increases geometrically with the number of allowed jumps within the sliding window. Hence, the set $S^k$ contains only hypotheses with either no jump or a single jump. The set $S^k$ is then defined as:

$$S^k \triangleq \left\{ M_i^k ; i = 0,\cdots,w \right\} \quad w \leq w^* \tag{3}$$

$$M_i^k \triangleq a_0^u(k)\left[1(0) - 1(k-i)\right] + a_i^u(k)1(k-i), \quad k \in N, \quad i = 0,...,w \tag{4}$$

where $1(l)$ signifies the unit step function at time instant $l$, and $a_i^u(k)$ are values to be adapted on-line based on the measurements.

The constant number of models form a M-ary Bayesian tree which renders the computational scheme finite. The model set $S^k$ has a variable structure as the set of hypotheses travels together with the sliding window of hypotheses. As such, the adaptive, variable structure, GLR estimator is a multiple model estimator in which the bank of models is modified on-line at each time instant. The flowchart of the estimation algorithm is displayed in Figure 1.

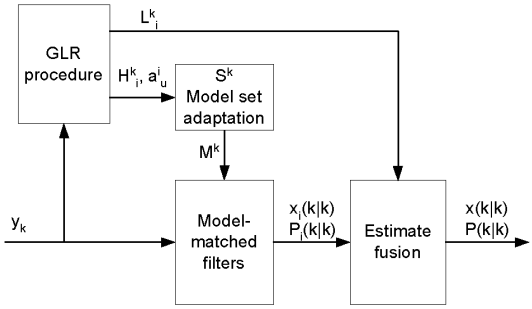



**Figure 1.** Adaptive, variable structure, GLR estimation algorithm.

The four steps of the algorithm are summarized as follows:

- **step 1:** the GLR procedure
  The current value of $a_0^u(k)$ is determined and the maximum likelihood estimate of $a_j^u(k), j = 1, \cdots, w$, are calculated using the likelihood ratios $L_i^k$ associated with each of the hypothesis $H_i^k$;
- **step 2:** model set adaptation
  The model set $S^k$ is constructed in which each of the members $M_i^k$ is associated with a corresponding hypothesis $H_i^k$ using the current estimate of the acceleration $a_i^u(k)$;
- **step 3:** model-matched estimation
  The model-matched state estimates, $\hat{x}_i(k \mid k)$, and their covariance matrices, $P_i(k \mid k)$, are calculated (for the calculation of $\hat{x}_i(k \mid k)$, it is assumed that the true system model is $M_i^k$);
- **step 4:** estimate fusion
  The state estimate $\hat{x}(k \mid k)$ and its covariance $P(k \mid k)$ at time instant $k$ are calculated by fusing the information contained in the $\hat{x}_i(k \mid k)$, $P_i(k \mid k)$, and $L_i^k$.

### *3.1. The GLR Procedure and Model Set Adaptation*

The GLR approach requires stating a number of hypotheses, $H_i^k$, that adequately describe the jump process $z$:

$H_i^k$ : there exist values $a_0^u(k), a_i^u(k)$ such that $M_i^k$ is true (5)

The maximum likelihood estimate of $a_i^u(k)$, assuming that model $M_i^k$ is true, is calculated recursively using a GLR algorithm:

$$a_i^u(k) = a_0^u(k) + \hat{\nu}_i^k a^\rho \qquad i \in \{1, ..., w\} \tag{6}$$

where $a^\rho$ is a hypothetical value of a jump and $\hat{\nu}_i^k$ is a scaling factor given by

$$\hat{\nu}_i^k = \frac{d(k,i)}{J(k,i)}; \tag{7}$$

for the recursive formulae to compute the signature correlation, *d(k,i)*, and the Kull-back-Leibler divergence, *J(k,i)*, see [6]. The value of $a_0^k$ is updated according to the following procedure:

$$a_0^k = a^{H_0} \tag{8}$$

where $a^{H_0}$ denotes the value of the jump process assumed by the reference Kalman filter for the GLR procedure. The way in which the value $a^{H_0}$ is adjusted depends on the particular version of GLR algorithm employed (e.g. standard GLR or adaptive-$H_0$ GLR, cf. [6]).

Also, let $\Lambda_i^k$ be the likelihood of the model $M_i^k$ at time instant $k$, i.e.

$$\Lambda_i^k \triangleq P(Y^k \mid M_i^k) \quad \text{for } k \in N, i \in 0, \cdots, w \tag{9}$$

and let the likelihood ratio, $L_i^k$, with respect to hypothesis $H_0^k$, be

$$L_i^k \triangleq \frac{\Lambda_i^k}{\Lambda_0^k}, \quad L_0^k = 1 \tag{10}$$

Then, the computation of the likelihood ratio $L_i^k$ is performed using:

$$L_i^k = e^{l(k,i)} \tag{11}$$

where the log-likelihood ratio, $l(k,i)$, is recursively calculated using a GLR procedure; see [6] for the details concerning this calculation.

*3.2. Model-Matched Estimates and Estimate Fusion*

With the reference to the hybrid system (1), and assuming that the bank $S^k$ indeed contains the true model, the p.d.f. of the system's state is a Gaussian mixture with $w+1$ terms, i.e.:

$$p\left[x(k)\,|\,Y^k\right]=\sum_{i=0}^{w}P\left(M_i^k\,|\,Y^k\right)N\left[x(k);\hat{x}_i(k\,|\,k),P_i(k\,|\,k)\right],\quad k\in N \tag{12}$$

where the model-matched state estimate $\hat{x}_i(k\,|\,k)$ assumes that $M_i^k$ is true, and $P_i(k\,|\,k)$ is the associated model-matched covariance matrix. The minimum mean square state estimate, $\hat{x}(k\,|\,k)$, is then expressed as a probabilistic mixture using the Bayesian rule:

$$\hat{x}(k\,|\,k)=E\left(x\,|\,Y^k\right)=\sum_{i=0}^{w}E\left(x\,|\,M_i^k,Y^k\right)P\left(M_i^k\,|\,Y^k\right)$$

$$=\sum_{i=0}^{w}\hat{x}_i(k\,|\,k)P\left(M_i^k\,|\,Y^k\right) \tag{13}$$

The corresponding covariance matrix, $P(k\,|\,k)$, of the state estimate is:

$$P(k\,|\,k)=\sum_{i=0}^{w}P\left(M_i^k\,|\,Y^k\right)\times$$

$$\left\{P_i(k\,|\,k)+\left[\hat{x}_i(k\,|\,k)-\hat{x}(k\,|\,k)\right]\left[\hat{x}_i(k\,|\,k)-\hat{x}(k\,|\,k)\right]^T\right\} \tag{14}$$

The conditional a posteriori probability, $P\left(M_i^k\,|\,Y^k\right)$, of model $M_i^k$ is, according to the Bayesian rule:

$$P(M_i^k\,|\,Y^k)=\frac{P(Y^k\,|\,M_i^k)P(M_i^k)}{\sum_{j=0}^{w}P(Y^k\,|\,M_j^k)P(M_j^k)}=\frac{\Lambda_i^k P(M_i^k)}{\sum_{j=0}^{w}\Lambda_j^k P(M_j^k)} \tag{15}$$

where $P(M_i^k)$ is the *known* initial probability of model $M_i^k$. The equation (15) can be re-written as:

$$P(M_i^k \mid Y^k) = \frac{L_i^k P(M_i^k)}{P(M_0^k) + \sum_{j=1}^{w} L_j^k P(M_j^k)} \tag{16}$$

where the value of $L_i^k$ is obtained from (11).

Each of the modal state estimates $\hat{x}_i(k \mid k)$ and its covariance could be computed using a Kalman filter:

$$\hat{x}_i(k+1 \mid k) = F\hat{x}_i(k \mid k) + G_1 u^c(k) + G_2 M_i^k \tag{17}$$

$$\hat{x}_i(k \mid k) = \hat{x}_i(k \mid k-1) + K(k)\gamma_i(k) \tag{18}$$

$$\gamma_i(k) = y(k) - H\hat{x}_i(k \mid k-1) \tag{19}$$

where $K(k)$ is the Kalman gain and $\gamma_i(k)$ is the residual. All the filters in the set use the same value $K(k)$ since the computation of $K(k)$ does not involve the input history. Therefore, the modal state estimate $\hat{x}_i(k \mid k)$ and its covariance $P_i(k \mid k)$ can instead be efficiently computed using the recursion:

$$\hat{x}_j(k \mid k) = \hat{x}_0(k \mid k) + \Gamma(k, j)\hat{\nu}_j^k, \quad j \in \{1, \cdots, w\} \tag{20}$$

$$P_i(k \mid k) = P_0(k \mid k) \tag{21}$$

where $\Gamma(k, j)$ is a quantity already pre-computed by the GLR procedure, cf. [6].

## 4. Comparison of the IMM and GLR-MM Estimators

The estimation performance of the adaptive, variable structure, GLR-MM estimator, described above, is compared with the one obtained when using several classical IMM filters. The version of the GLR algorithm used by the GLR-MM estimator is the adaptive- $H_0$ GLR algorithm, as described in [6].

The GLR-MM estimator uses the following (unnormalized) total probability, $P(M_i^k)$:

$$P(M_i^k) = \begin{cases} 1 & \text{if } \mid \hat{z}_k \mid < z_{max} \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

where $\hat{z}_k$ denotes the estimated target's command acceleration and $z_{max}$ is a hard bound, $z_{max}$ = 30 g, for the maximum magnitude of the target acceleration. The hybrid system implemented is a linear model of a pursuit-evasion engagement between a maneuvering missile (target) and an interceptor with first-order lag dynamics, cf. [3] for the corresponding value of the matrices in equation (1). The only sensor is assumed to be located on-board of the interceptor. The sensor measures the lateral separation between the target and the interceptor. The whole engagement lasts 4 seconds. The target performs a bang-bang maneuver at $t$ = 1.00 s, i.e. the target's command acceleration changes from +15 g to –15 g. The remaining simulation parameters are the same as in [6].

**Table 1.** Simulation parameters

| | | | |
|---|---|---|---|
| pursuer velocity | $V_P = 2\,300$ m/s | evader velocity | $V_E = 2\,700$ m/s |
| pursuer max. acc | $(a_P^c)_{max} = 30$ g | evader max. acc. | $(a_E^c)_{max} = 15$ g |
| pursuer f.o.l.t.c. | $\tau_P = 0.2$ s | evader f.o.l.t.c. | $\tau_E = 0.2$ s |
| initial range | $r(0) = 20\,000$ m | measurement freq | $f = 100$ Hz |
| ang. noise std. dev | $\sigma = 0.1$ mrad | | |

The values of the parameters used in the simulations are provided in Table 1 where "f.o.l.t.c." stands for "first-order lag time constant," and $\sigma$ is the standard deviation of the *angular* measurement noise. The covariance, $Q_\eta$, of the *linearized* measurement noise was made state dependent as follows:

$$Q_\eta(k) = (r(k)\sigma)^2 \tag{23}$$

Five different, fixed structure, IMM estimators were used with regard to this scenario which produced state estimates that were compared with the corresponding estimates achieved by application of the GLR-MM estimator. The five IMM estimators are denoted IMM1, ..., IMM5.

The same noise realizations were used with all the estimators. The estimators were compared using the statistics obtained through Monte Carlo simulation which involved 100 different noise realizations.

### *4.1. Selection of Parameters for the Estimators*

The parameters of the GLR component of the GLR-MM were selected as follows: the probability of false alarm $\alpha = 0.001$, the sliding window width $w = 70$, and the value of the hard bound on the magnitude of $|\hat{z}_i(k)| \le \hat{z}_{max}$ was $\hat{z}_{max} = 100$ [g] for all $k$. The value of $\hat{z}_{\max}$ is much larger than $z_{\max}$ and is employed to discard only the hypotheses with an estimate $\hat{z}$ obviously wrong, even when taking into account the worst possible estimation errors. The following (unnormalized) total probability distribution, $P(M_i^k)$, for the models in the GLR-MM estimator was assumed:

$$P(M_i^k) = e^{\left(|\hat{z}_i(k)|-|\hat{z}_0(k)|\right)^2/\sigma_a^2} \tag{24}$$

where the standard deviation on the adapted values of the models was set to $\sigma_a = 10$ [g].

The estimator IMM1 incorporated 3 Kalman filters. Each filter approximated the unknown target's command acceleration using a Wiener process acceleration model (WPAM), cf. [9]. For the first filter, the covariance, $Q_w$, of the WPAM was set to zero, for the second filter $Q_w = 9$ [ g$^2$ ], and for the third filter $Q_w = 225$ [ g$^2$ ].

The estimators IMM2 and IMM3 incorporated 9 Kalman filters corresponding to 9 different constant acceleration models in a bank common to IMM2 and IMM3. The accelerations for these models were chosen to be: –30 g, –20 g, –10 g, –5 g, 0 g, 5 g, 10 g, 20 g, and 30 g. Note that none of the models in the banks for IMM2 and IMM3 matched the true target's command acceleration. The filters IMM2 and IMM3 differed by only their respective Markovian transition probability matrices, see below.

The estimators IMM4 and IMM5 incorporated 9 Kalman filters corresponding to 9 different constant acceleration models in a bank common to IMM4 and IMM5. The accelerations for these models were chosen to be: –30 g, –20 g, –15 g, –10 g, 0 g, 10 g, 15 g, 20 g, and 30 g. In this case, both IMM4 and IMM5 employed banks which contained (unrealistically) the exact target's command accelerations before and after the jump. These unrealistic but, in a sense, ideal estimators were useful to provide an upper bound on the performance of the IMM algorithm. The filters IMM4 and IMM5 differed by their respective Markovian transition probability matrices.

The elements, $P(M_i^k \mid M_j^{k-1})$, of the Markovian transition probability matrix for the IMM estimators were set to:

$$\begin{aligned}
&\text{IMM1} \quad \Rightarrow \begin{cases} P(M_i^k \mid M_i^{k-1}) = 0.9800 \\ P(M_i^k \mid M_i^{k-1}) = 0.0067 \quad \text{for } i \neq j \end{cases} \\
&\text{IMM2, IMM4} \quad \Rightarrow \begin{cases} P(M_i^k \mid M_i^{k-1}) = 0.9800 \\ P(M_i^k \mid M_i^{k-1}) = 0.0025 \quad \text{for } i \neq j \end{cases} \\
&\text{IMM3, IMM5} \quad \Rightarrow \begin{cases} P(M_i^k \mid M_i^{k-1}) = 0.9990 \\ P(M_i^k \mid M_i^{k-1}) = 0.000125 \quad \text{for } i \neq j \end{cases}
\end{aligned} \tag{25}$$

### 4.2. Simulation Results

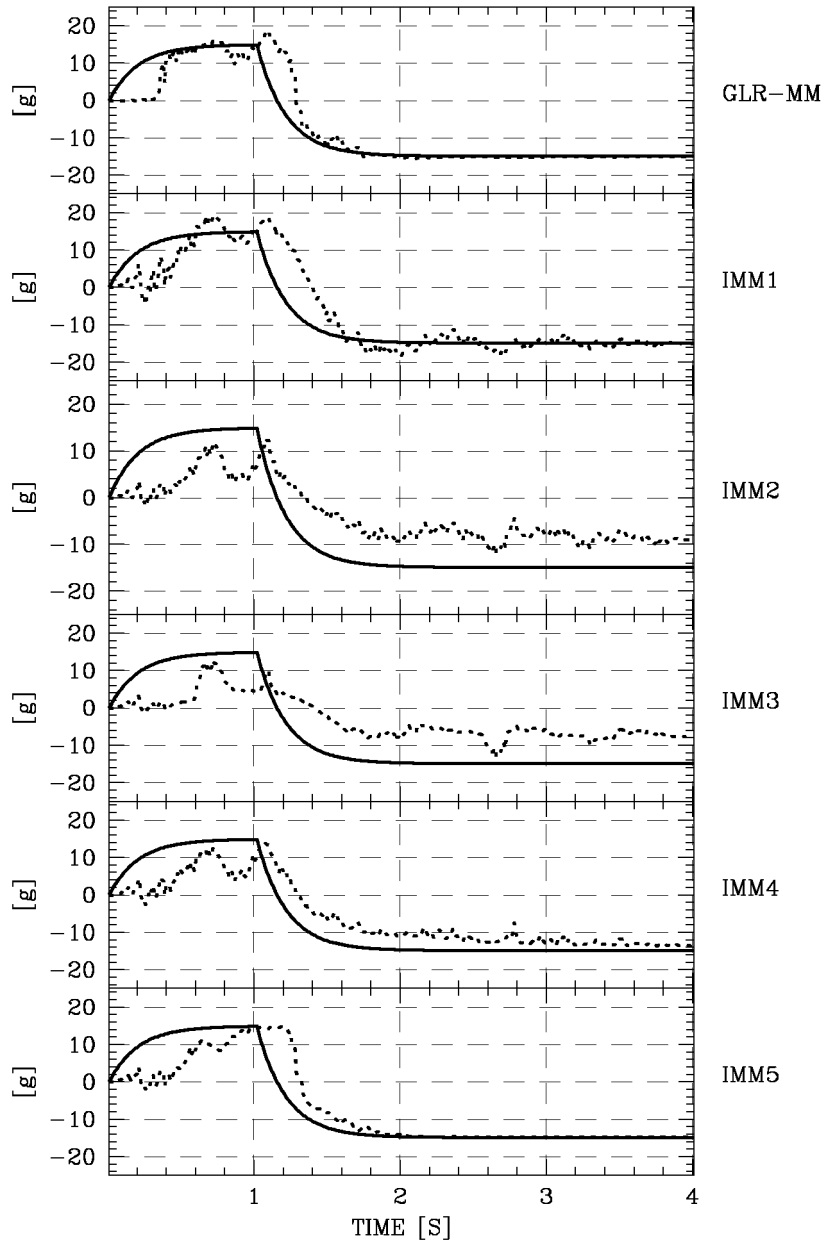



**Figure 2.** Estimation of the target's achieved acceleration. From top panel to lower panel, the estimators used are: GLR-MM, IMM1, IMM2, IMM3, IMM4, and IMM5.

The estimated target's achieved acceleration, when using the estimators GLR-MM and IMM1 through IMM5, is depicted in Figure 2 (using one particular noise realization). The estimates correspond to a particular sample path of the process $z$ (target's command acceleration) which exhibits a single jump at time $k = 100$ ($t = 1$ s) of magnitude 30 g; $z(k) = 15$ g for $k \in [0,100]$ and $z(k) = -15$ g for $k \in [100, 400]$. An initial mismatch is assumed between the command and the achieved acceleration of the target, the latter being equal to zero. As compared to the acceleration estimate produced by the GLR-MM, the estimate obtained from the IMM1 converges to the true achieved acceleration slower and is more sensitive to noise. The estimates from the estimators IMM2 and IMM3 exhibit a significant bias and do not converge towards the true achieved acceleration of the target. This bias is present because none of the models used by the estimators IMM2 and IMM3 match the true target's command acceleration. The biased estimates from the estimators IMM2 and IMM3 are also visibly more sensitive to noise than the estimate from the GLR-MM estimator. The estimates from the IMM4 and IMM5 estimators converge to the true value of $a_E$ slower than the estimate from the GLR-MM estimator. Only, the estimate from the IMM5 estimator has a better noise rejection than the estimate from the GLR-MM estimator. This, seemingly better performance of the IMM estimator, is however produced artificially, by assuming an unrealistically high Markovian probability (0.999) for model $M_i^k$ to remain in effect. Despite the fact that IMM3 employs the same Markovian probability matrix, it is still very sensitive to noise as the true model is absent from the bank.

To summarize the above results, the GLR-MM estimator seems to perform better than the fixed structure IMM estimators in spite that it does not require any prior information about either the time instant of the jump or the values of $z$ before and after the jump. It is only in the case of IMM5 when the performance of the GLR-MM and the IMM are really comparable. However, as pointed out previously, the result achieved through IMM5 requires certainty about the values of the process $z$ before and after the jump. Also, the IMM5 insensitivity to noise, as enforced by a special selection of the Markovian transition probabilities, can be matched by a similar insensitivity of the GLR-MM if the last is indeed allowed to use equally unrealistic parameters.

The results concerning the statistics obtained through Monte Carlo simulation which involved 100 different noise realizations are shown in Figure 3 and Figure 4. As in Figure 2, the plots corresponds to the results obtained while employing the GLR-MM versus IMM1 through IMM5 (the sample errors from which the statistics were computed correspond to the same noise realizations). Again, the GLR-MM estimator is seen to perform better than the IMM1-IMM4 as the average error for IMM1 is larger than that for GLR-MM and the average error of the remaining estimators IMM2-IMM4 have non-zero bias. Even in the case of the unrealistic estimator IMM5, its average error exceeds that of GLR-MM. It is only the standard deviation of the error that is smaller for the IMM5 than that of GLR-MM.

Better performance of the GLR-MM estimator over the IMM estimator is attributed to the two key features of the GLR-MM: its variable structure and capacity for model set adaptation.

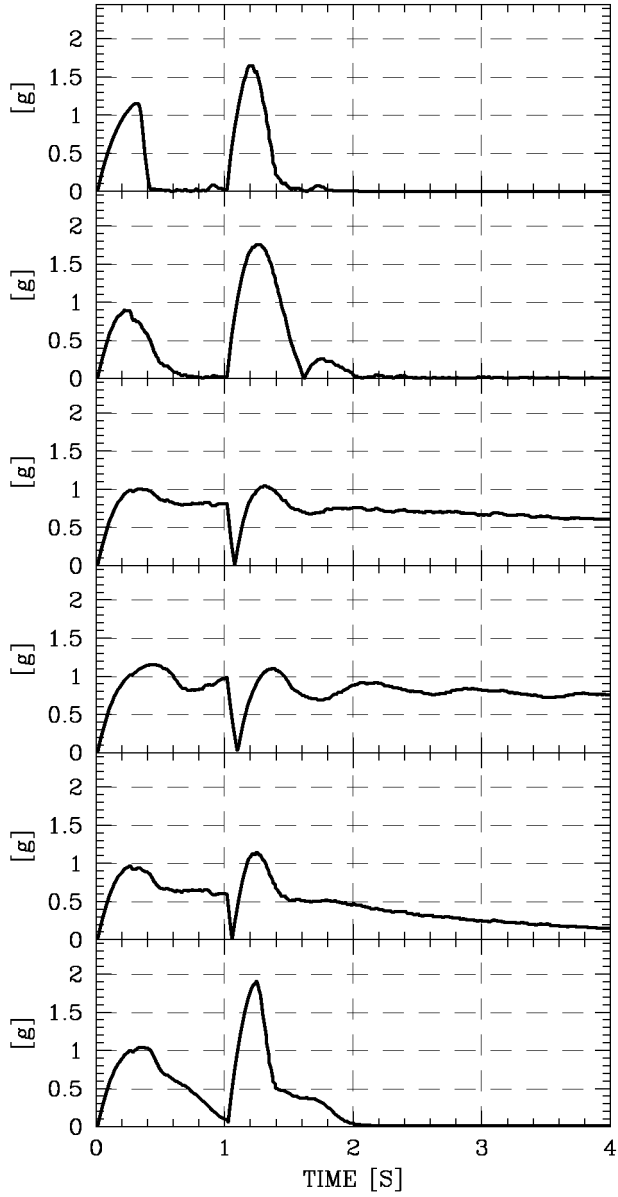



**Figure 3.** Magnitude of the average error in the estimate of the target's achieved acceleration. From top panel to lower panel, the estimators used are: GLR-MM, IMM1, IMM2, IMM3, IMM4, and IMM5.

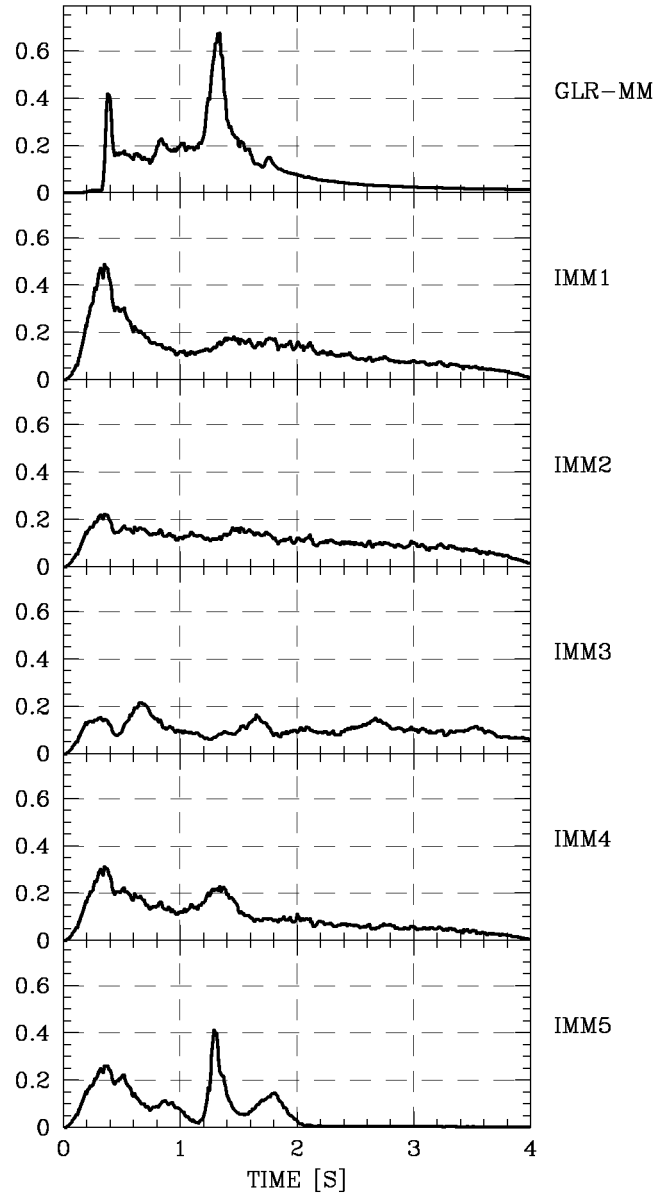



**Figure 4.** Standard deviation of the error in the estimate of the target's achieved acceleration. From top panel to lower panel, the estimators used are: GLR-MM, IMM1, IMM2, IMM3, IMM4, and IMM5.

– the variable structure of the GLR-MM refers to the structure of its bank of models. The structure of this bank is related to the formation of the set of hypotheses which can take account of additional information such as an upper bound on the possible frequency of jumps. Additionally, this bank of models explicitly carries information about the recent history of the process $z$ in

terms of the hypotheses made. In the standard fixed structure IMM estimator (using the assumption about Markovian transitions) it is virtually impossible to take account of information concerning the frequency of jumps;

– the model set adaptation in GLR-MM is very important as it effectively allows for restricting the number of models used. For each model structure considered (for each hypotheses $H_i^k$), a single model is created in the bank $S^k$. For comparison, the IMM estimator which would use a model structure corresponding to that of $H_i^k$ would require several sub-models to allow for variability in the value of process $z$ before and after the jump. Since an exact value of $z$ can never be guessed a priori, the IMM estimator would always be characterized by an error bias.

## 5. Concluding Remarks

To the best of our knowledge, the algorithm presented is the first attempt to incorporate a self-adapting generalized likelihood ratio detector test into a multiple model type estimator. Preliminary experiments show that the GLR-MM algorithm, which employs an extended set of hypotheses, has many advantages over the acclaimed IMM estimator. These advantages are particularly pronounced when the system structure exhibits sudden changes. This is not surprising as variable structure estimators have recently been recognized to perform better over fixed structure ones. Additionally, the algorithm developed leaves room for improvement such as provided by an even better construction of the set of hypotheses. Future work will be directed toward extending the algorithm to handle multiple jumps and to be able to incorporate additional probabilistic information about the jump process itself.

## References

[1] Lai, T.L., & Shan, Z., "Efficient recursive algorithms for detection of abrupt changes in signals and systems," IEEE Transactions on Automatic Control, vol. 44, N. 5, 1999, pp. 952–966.

[2] Willsky, A.S., & Johns, H.L., "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," IEEE Transactions on Automatic Control, vol. AC-21, 1976, pp. 108–112.

[3] Shima, T., Oshman, Y., & Shinar, J., "Efficient multiple model adaptive estimation in ballistic missile interception scenarios," Journal of Guidance, Control, and Dynamics, v. 25, n. 4, 2002, pp. 667–675.

[4] Caglayan, A.K., "Simultaneous failure detection and estimation in linear systems," Proceedings of the Conference on Decision and Control, 1980, pp. 1038–1041.

[5] Schnepper, K., "A comparison of GLR and multiple model filters for a target tracking problem," Proceedings of the Conference on Decision and Control, 1986, pp. 666–670.

[6] Dionne, D., & Michalska, H., "Integrated estimation and guidance for target interception," Data fusion for situation monitoring, incident detection, alert, and response management, NATO ASI series, Kluwer-Dordrecht, 2003.

[7] Kushner, H.J., "Robustness and convergence of approximations to nonlinear filters for jump-diffusions," Computational and Applied Math., v. 16, 1997, pp. 153–183.

[8] Li, X.R., "Hybrid estimation techniques," Control and Dynamic Systems v. 76, Academic Press, pp. 213–287.

[9] Bar-Shalom, Y., Li, X.R., & Kirubarajan, T., "Estimation with applications to tracking and navigation," Wiley-VCH publisher, ISBN: 0-471-41655-X, 2001.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# A Distributed Sensor Network for Real-Time Multisensor Surveillance Against Possible Terrorist Actions

Gian Luca FORESTI and Lauro SNIDARO
*Department of Mathematics and Computer Science, University of Udine,*
*via delle Scienze 206, 33100 Udine, Italy*

**Abstract.** In this paper, a Distributed Sensor Network (DSN) architecture has been designed for fusing multisensor data in a video surveillance system capable of understanding possible terrorist actions. The system uses both physical (i.e., IR and optical cameras) and virtual (i.e. processing units transforming information) sensors. Internal nodes of the DSN fuse data in order to reduce the degree of uncertainty naturally associated with the acquired information, and to produce an interpretation of the environment observed. The performance of the cameras is automatically assessed to regulate the fusion process accordingly. In particular, the system is able to detect, recognize, and track moving people in outdoor environments by analyzing their behavior through trajectory and event analysis. Suspicious behaviors are then signaled to a remote operator supervising the system.

**Keywords.** Surveillance systems, multisensor data fusion, object detection, object tracking, behavior understanding

## 1. Introduction

The events of September 11, 2001, have demonstrated that there is a need for improving surveillance capabilities of public areas (e.g., airports, metro or railway stations, parking lots, tunnels or bridges, etc.) in order to prevent terrorist acts. Advanced multisensor surveillance systems represent a possible answer to prevent terrorist attacks by enhancing monitoring and control capabilities of remote human operators in large environments. Such systems can perform real-time intrusion detection and/or suspicious event detection in complex environments.

New generation surveillance systems [1–3] require managing large amounts of visual data (optical, infrared, etc.). Recently, the development of sensor technology and computer networks has contributed to increasing the interest in Distributed Sensor Networks (DSNs) for real-time information fusion [4–6].

DSNs are basically systems composed of a set of sensors, a set of processing elements (PEs), and a communication network interconnecting the PEs. In this paper, a DSN architecture for a video surveillance system is proposed where sensors may be either physical (i.e., to generate information by translating observed physical quantities into electrical signals) or virtual (i.e. to produce new information from existing information). PEs fuse data acquired by different physical and/or virtual sensors, or preprocessed by other PEs at lower levels, in order to reduce the degree of uncertainty

naturally associated with the acquired information, and to produce an interpretation of the environment observed. The dimensionality of information is reduced, as only significant information is propagated through the network.

The proposed DSN architecture integrates optical and infrared sensors to support a 24h/day real-time visual-based surveillance system for outdoor environments. IR and optical sensors are at the first level of the proposed architecture. Video signals of each physical sensor are first processed to extract moving image regions, called *blobs* [7], and features are computed for the target tracking, classification, and data fusion procedures. This integration allows to improve at higher levels the accuracy of object localization, which is based on the ground plane hypothesis [7] and object recognition [8]. At the first level, specialized PEs track each detected blob on the image plane and transform 2D blob positions (in the sensor coordinates system) into 3D object positions (in the coordinates of the monitored environment's map). Each first level PE is committed to the surveillance of a sub-area of the monitored environment.

The trajectory of each blob, extracted by a given sensor, is first approximated with cubic splines [7]. Splines are used to save bandwidth as they can represent a trajectory with a very limited number of points. In this way only spline parameters need to be sent through the network thus avoiding the transmission of target positions at every time instant.

At the higher level of the architecture a trajectory fusion of the local object trajectories is performed to compute the trajectory of the objects with respect to the whole map of the monitored site. Information about object trajectories and blob features can be used to teach a neural network (e.g. a neural tree [8,9]) to recognize suspicious events in the observed scene [10,11].

A real example of the proposed DSN architecture employing infrared and optical sensors will be presented in the context of the video surveillance of an outdoor parking area. The automation of the surveillance of such areas is considered of particular importance as it is one of the goals of the NATO task group "Advanced Multisensor Surveillance Systems for Combating Terrorism."

The developed system is able to detect and recognise basic events (i.e. vehicle or person moving, stopping, entering, exiting, etc.) and complex ones as well (i.e. interactions between persons and vehicles). The operator can have the system recognise a number of events as suspicious (i.e. non linear trajectories) and signal them in real-time as they are occurring.

The paper is organized as follows. The next Section describes the general architecture of the system, while Section 3 presents the processing steps involved. Experimental results are given in Section 4.

## 2. DSN Architecture

DSN architectures have been a topic of debate since the early 80s [12]. A review of recent advancements can be found in [13] and [21]. Although general discussions on the advantages and disadvantages of the various network topologies are vastly present in the literature, little work has been done on DSNs for video surveillance systems. These are characterized by sources generating great amounts of data at high frequency (typically 25 fps). Therefore, some of the parameters to be considered in choosing the most appropriate network structure are the following:

- data to be sent;
- network bandwidth;
- data fusion technique;
- efficiency;
- cost.

The overall system architecture is shown in Figure 1. The network has a tree structure where the lowest level (leaves) is constituted by heterogeneous (optical, b/w, IR, etc.) multi-resolution sensors. Internal nodes represent processing elements.

Internal nodes (PEs) are subdivided into First Level nodes and Higher level nodes. The former are directly connected to the sensors, forming the so called *clusters* [13], and are in charge of the monitoring of sub areas of the environment under surveillance. The latter receive data from the connected First Level nodes for broader area coverage. The tree topology has been chosen as it offers a hierarchical flow of data and it is easier to extend than the general Anarchic Committee (AC) structure [12] where no hierarchy is defined. Intrinsically more vulnerable then a fully interconnected scheme, the tree topology can be made more fault tolerant adding a certain degree of redundancy to the inner nodes (PEs).

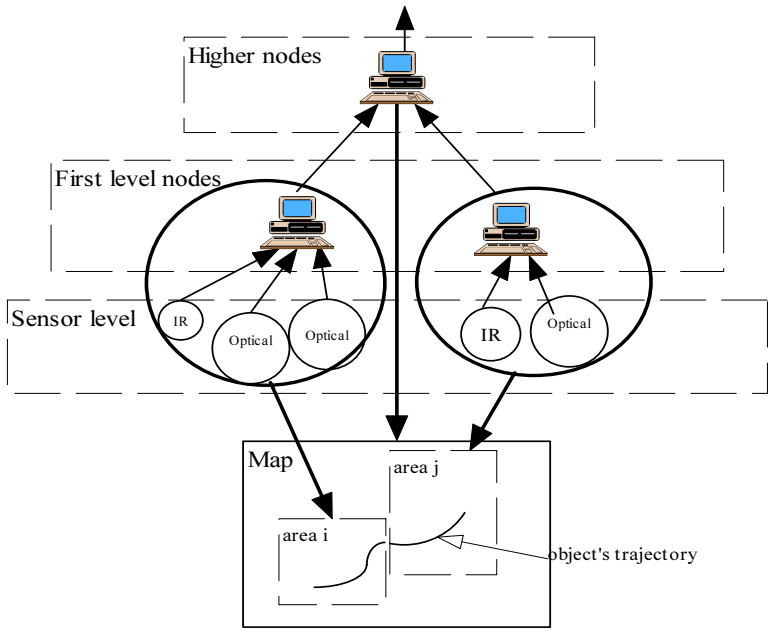



**Figure 1.** The proposed DSN architecture.

Table 1 summarizes some of the characteristics of the possible architectures along with their key strengths and weaknesses.

Hierarchical in nature, the chosen topology allows an intuitive partitioning of the sensors according to the different sub-areas of the environment to be monitored.

**Table 1.** Architecture comparison

| | centralized | hierarchical | distributed |
|---|---|---|---|
| Characteristics | single fusion site<br>single track database | hierarchical structure | no central fusion site<br>no fixed structure |
| Advantages | theoretically optimal<br>simple structure | sensors naturally partitioned<br>simple fusion algorithm | highly fault tolerant<br>processing load is distributed |
| Disadvantages | high processing load<br>high bandwidth requirements<br>low reliability | suboptimal | complicated fusion and communication algorithms<br>non-optimal |

More sophisticated architectures such as the flat tree [14] or the deBruijin graph [15] could be more difficult and expensive to implement, and may not even be useful since with the proposed approach there is no need to exchange data between nodes of the same level. A fully distributed scheme is the most robust but also the most complicated. Sophisticated fusion and communication algorithms have to be developed and the architecture is also non-optimal in nature. An in-depth comparison and discussion can be found in [21].

The sensor level of the proposed architecture is constituted by clusters of sensors. Heterogeneous sensing technologies (optical, infrared, etc.) may be employed to assure the surveillance system's operational state during day and night and in presence of changing weather conditions. Active sensors can also be employed in a cooperative fashion as in [16,20]. Each cluster sends data to the connected first level Processing Element (PE).

Every PE is committed to the surveillance of a sub-area of the monitored environment. These nodes perform multi-sensor multi-target tracking [24,25] and send information about the detected objects to higher level nodes that are responsible for trajectory analysis [39] throughout the controlled sub-areas, that is, they can examine a target's path as a whole (Figure 1). Event detection [10,11] and data fusion can also be performed at this level for decision making based on hypothesis [22,23].

A detailed description of the processing steps performed at each level is given in the following sections.

## 3. Processing

This section describes the processing steps that take place in each component of the proposed architecture. The flow-chart in Figure 2 shows the flow of data through the nodes and the procedures involved; the configuration considered consists of two heterogeneous sensors (optical and infrared) connected to a first level node which in turn feeds a higher level node.

### *3.1. Blob Extraction*

This processing step occurs at sensor level and pinpoints moving regions in the image through change detection algorithms [1,2,7,17–19]. Motion detection and blob extraction is exploited following a layered background subtraction approach [16]. Change

detection is performed using an algorithm for automatic threshold computation based on Euler numbers [19]. The background is updated using a Kalman filter [8]. Frame by frame subtraction is also applied in conjunction to improve detection results [16]. Morphological filters are also applied to improve the quality of the extracted blobs by removing spurious pixels due to noise, and by enhancing the connectivity of regions [16].

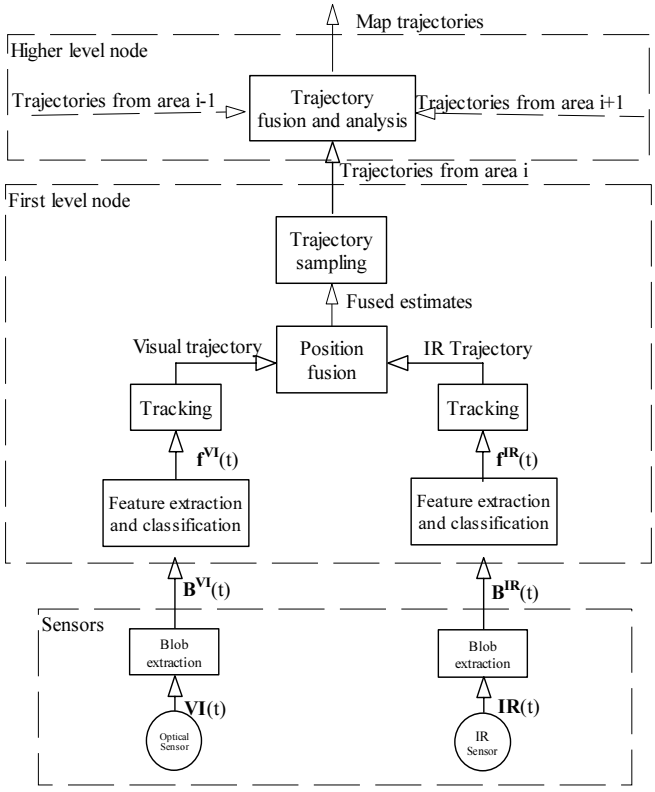



**Figure 2.** Processing steps.

Following the scheme in Figure 2, at time *t*, images **VI**(t) and **IR**(t) are produced respectively by the optical and infrared sensor. Each sensor applies the blob extraction procedure thus obtaining the arrays $\boldsymbol{B}^{VI}(t)$ and $\boldsymbol{B}^{IR}(t)$ containing the blobs extracted on the current frame at time *t* respectively by the optical and the infrared sensor.

The blobs extracted from the video signals obtained from a color and an IR camera respectively are shown in Figure 3.

**Figure 3.** Optical and IR signals and extracted blobs.

### 3.2. Feature Extraction and Classification

Blobs collected from sensors undertake further processing at first level nodes. In particular, some features are needed to describe the moving objects. Typical measures are area, perimeter, sparseness (perimeter$^2$/area), and centroid coordinates of the blob, along with measurements regarding the dimensions of the blob's bounding box (base, height, base/height ratio). Additional information may be acquired by computing and analyzing the color histograms of the blobs [31,32].

This information (in the form of the feature vector $\mathbf{f}^s(t)$, where $s$ indicates the sensor) may be used by the module responsible for object recognition [8,16]. A classifier (e.g. a neural tree [9]) may be employed to identify the objects hence adding precious data to the feature vector characterizing the object. The more information is filled in the feature vector, the more robust will be the subsequent target tracking procedure.

### 3.3. Target Tracking

The system needs to maintain tracks for all the objects that exist in the scene simultaneously. Hence, this is a typical multi-sensor multi-target tracking problem: measurements should be correctly assigned to their associated target tracks and a target's associated measurements from different sensors should be fused to obtain better estimation of the target state.

A first tracking process may occur locally to each image plane. For each sensor, the system can perform an association algorithm to match the current detected blobs with those extracted in the previous frame. A number of techniques is available, spanning from template matching, to features matching [16], to more sophisticated approaches [32].

Generally, a 2D top view map of the monitored environment is taken as a common coordinates system [7,8], but even the GPS may be employed to globally pinpoint the targets [16]. The former approach is obviously more straightforward to implement, as a well-known result from projective geometry states that the correspondence between an image pixel and a planar surface is given by a planar homography [29,30]. The pixel usually chosen to represent a blob and be transformed into map coordinates is a projection of the blob's centroid on the lower side of the bounding box [7,8,16].

To deal with the multi-target data assignment problem, especially in the presence of persistent interference, there are many matching algorithms available in the literature: Nearest Neighbor (NN), Joint Probabilistic Data Association (JPDA), Multiple Hypothesis Tracking (MHT), and S-D assignment. The choice depends on the particular application; detailed descriptions and examples can be found in [24,33,34].

Recent developments on the subject may be found in [16,35–38].

### 3.4. Position Fusion

Data obtained from the different sensors (extracted features) can be combined together to yield a better estimate. A typical feature to be fused is the target's position on the map. A simple measurement fusion approach through a Kalman filter [26,42] was employed for this purpose. This fusion scheme involves the fusion of the positions of the target (according to the different sensors) obtained right out of the coordinate conversion function, as can be seen in Figure 4(a). This is an optimal algorithm, while the track-to-track fusion scheme, depicted in Figure 4(b), has less computational require-

ments but is sub-optimal in nature since performing data fusion with feedback involves the decorrelation of the local estimates as they are affected by a common process noise and cannot be considered independent [24].

The measurement fusion is not only optimal in theory, it also shows good results in practice [40]. However, when dealing with extremely noisy sensors (i.e. video sensors performing poorly due to low illumination conditions), the track-to-track scheme is generally preferred to running a Kalman filter for each track to obtain a filtered estimate of the target's position, thus smoothing high variations due to segmentation errors. In the following Section a confidence measure is presented to weight sensor data coming from video sensors that allows the use of the measurement fusion scheme.

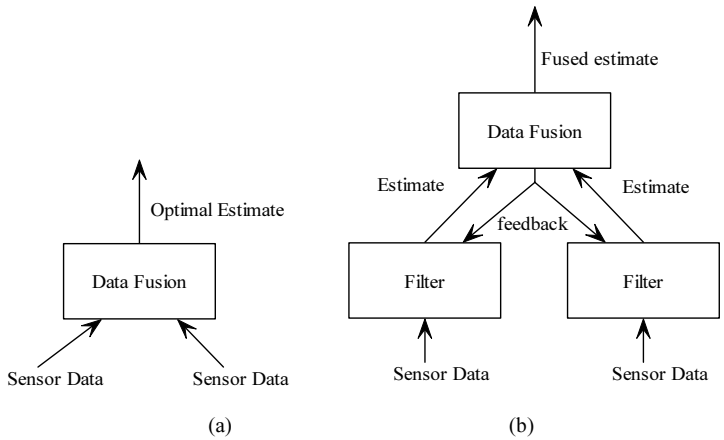



**Figure 4.** (a) Optimal fusion and (b) track fusion.

*3.4.1. The Appearance Ratio (AR)*

The following measure, called Appearance Ratio (AR), gives a value to the degree of confidence associated with the *j*-th blob extracted at time *t* from the sensor *s*:

$$AR(B_{j,t}^{s}) = \frac{\sum_{x,y \in B_{j,t}^{s}} D(x,y)}{\left|B_{j,t}^{s}\right| C} \tag{1}$$

where $D(x,y)$ is the difference map obtained as absolute difference between the current image and the reference image, and $C$ is a normalization constant depending on the number of color tones used in the image. The AR is thus a real number ranging from 0 to 1 that gives an estimation of the level of performance of each sensor for each extracted blob. The AR values for the blobs of Figure 3 are shown in Figure 5. As can be seen, the AR value for the blob extracted from the infrared sensor (right) is considerably higher than the optical one (left).

AR values are then used to regulate the measurement error covariance matrix to weight position data in the fusion process. The following function for the position measurement error has been developed:

$$r(B_{j,t}^{s}) = GD^{2}(1 - AR(B_{j,t}^{s})) \tag{2}$$

where *GD* is the gating distance. The function is therefore used to adjust the measurement position error so that the map positions calculated for blobs with high AR values are trusted more (i.e. the measurement error of the position is close to zero), while blobs poorly detected (low AR value) are trusted less (i.e. the measurement error equals the gating distance).

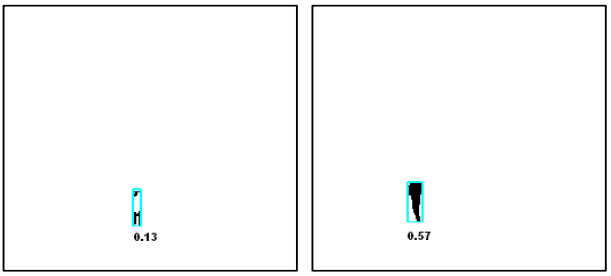



**Figure 5.** AR values for the (left) optical and (right) IR blobs of Figure 3.

### *3.5. Trajectory Sampling*

This module receives fused estimates for each object's map position in the area controlled by the PE and transmits their trajectories to a higher level node. To reduce bandwidth requirements, this module samples objects' positions every $z$ time intervals and calculates the cubic spline passing through the points $M(k)$ and $M(k+z)$ [41].

Only the starting and ending points and tangent vectors are needed to define a spline segment. Each of them may span a predefined number of measurements (track points on the map) or, to improve bandwidth savings, a dynamically computed one. In particular, the sampling procedure may take into account whether the target is maneuvering or not and regulate the sampling frequency accordingly (the trajectory of a maneuvering target requires more spline segments to be described). The spline segments are then sent to the higher level nodes.

### *3.6. Trajectory Fusion and Analysis*

The trajectory fusion occurs at higher level PE. These nodes are in charge of composing the trajectories of the targets by joining together the spline segments sent by the first level PE. The joining itself poses no problem when the segments are provided by a single first level PE, as every spline segment is received along with the target's ID so that the association is easily performed. When a target moves from a sub-area to another controlled by a different first level PE, an association algorithm (Section 3.3) is required to maintain the target's ID at the higher level PE.

The higher level nodes are the ideal processing elements for running trajectory analysis algorithms as they have a broader view of the behavior of the targets. Depending on the specific application, the system can detect and signal to the operator dangerous situations (e.g. a pedestrian walking towards a forbidden area, a vehicle going in a zigzag or in circles, etc.) [10,11].

## 4. Results

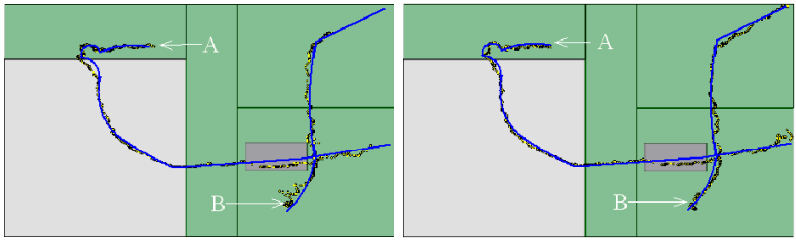
A color camera and a b/w camera with near infrared response have been employed for experiments. Image grabbing was performed at 256x256 pixels resolution. The cameras have been placed to monitor the same area of the parking lot at the Rizzi building of the University of Udine. As already mentioned, parking lots are particularly interesting for surveillance and counter-terrorism purposes. In the following experiment, the two cameras are employed in daylight conditions and two persons are walking in the scene. Here the purpose of the experiment is to verify the accuracy of the trajectories obtained through data fusion against the ones given by the single sensors.

Figure 6 shows the source and processed frames for the b/w and color cameras respectively. In Figure 8 the trajectories obtained by the fusion procedure are more continuous than those produced by the single sensors, especially when compared with the B/W ones in Figure 7(a), even though the coloring of dots for each trajectory would be needed to show that fact. Moreover, the fused trajectories are more similar to the ground truth (curved lines) as confirmed by Table 2 and 3. This is due to the reduction of calibration errors through data fusion.

Note that the color camera is not performing better than the b/w for each frame; as shown in Figure 6, blob "A" has been extracted with better results from the b/w video signal. This is indicative of the effectiveness of the heterogeneous sensor fusion approach.

During the tests the trajectories were performed by an average number of 7 splines segments. The average number of measurements composing each track was roughly 1000. So, only 28 points per track were sent to the higher level node instead of 1000.


**Figure 6.** Source and processed frames for the b/w and color cameras.




(a) (b)

**Figure 7.** Trajectories of the two persons in Figure 6 according to the (a) b/w and (b) color cameras.

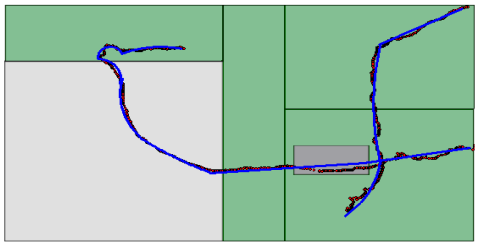


**Figure 8.** Trajectories of the persons in Figure 6 obtained through data fusion.

**Table 2.** Mean and standard deviation (in pixels) of the distance between estimated and ground truth positions for the trajectories of blob "A" in Figures 6 and 7

| | mean | standard deviation |
|---|---|---|
| IR camera | 10.64 | 4.75 |
| color camera | 8.94 | 3.62 |
| Data fusion | 8.21 | 3.02 |

**Table 3.** Mean and standard deviation (in pixels) of the distance between estimated and ground truth positions for the trajectories of blob "B" in Figures 6 and 7

| | mean | standard deviation |
|---|---|---|
| IR camera | 11.2 | 5.23 |
| color camera | 9.83 | 4.67 |
| data fusion | 8.61 | 3.79 |

Achieving better trajectory accuracy is of paramount importance for the successive step of behavior understanding. The system can recognize some basic behaviors like person/vehicle entering, moving, stopping, exiting the area, etc. But it can also take into account compound and more complex situations such as the one shown in Figure 9. In the images, a vehicle enters the area (ID 1), stops, its drivers exits (ID 2), and walks away leaving the area. Then another person walks in the scene (ID 3). The trajectories area is shown in Figure 10. The system can therefore recognize the objects (vehicle, person, group of persons) and some interactions between them. These events are logged into a database.

**Figure 9.** Source images for the parking lot sequence.

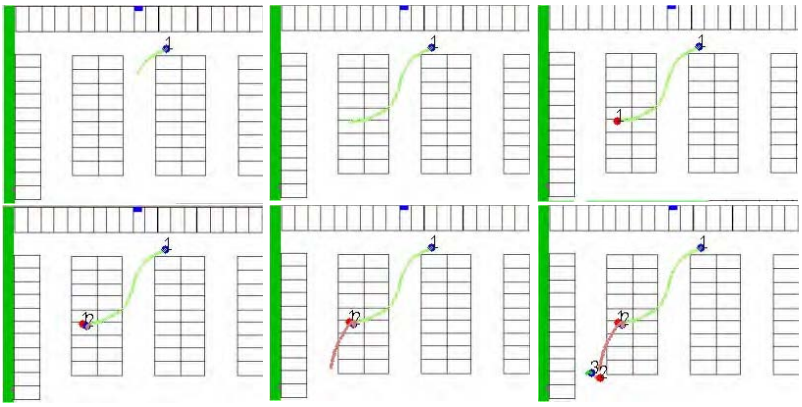


**Figure 10.** Trajectories on the top-view map for the parking lot sequence.

In addition to this logging activity, the system can provide real-time responses to suspicious events. In fact, it can detect unusual patterns (like zigzagging trajectories) in the movements of a pedestrian or a vehicle and signal them to an operator. It can also report events like a vehicle entering the area and stopping and no person getting out of it, or a person getting into a vehicle that doesn't move away. The system is undertaking further development to improve the detection of more complex events with the scope of preventing possible terrorist actions.

## 5. Conclusions

In this paper, a distributed sensor network for video surveillance has been presented. The system which is able of day and night operation even in presence of adverse weather conditions has been designed to detect, track and understand human behaviors in order to individuate suspicious situations, e.g., possible terrorist actions.

The system, which uses heterogeneous multi-resolution sensors and performs data fusion at various levels, is able to track people moving in outdoor scenes more efficiently and accurately than standard surveillance systems. The proposed DSN architecture also reduces networks requirements and is easily scalable and maintainable.

A metric, called Appearance Ratio, has been used to evaluate the performance of the sensors and weight the fusion of the measurements accordingly. The experiments have shown that the trajectories obtained through data fusion are more accurate and continuous than those produced by the single sensors, primarily due to the reduction of the calibration error.

The developed system has been applied to recognize suspicious events and trajectories in parking lots with the purpose of preventing possible terrorist attacks.

## Acknowledgments

This work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project "Distributed systems for multisensor recognition with augmented perception for ambient security and customization" (2002–2004).

## References


[1] G.L. Foresti, C.S. Regazzoni and R. Visvanathan, " Scanning the Issue/Technology – Special Issue on Video Communications, Processing and Understanding for Third Generation Surveillance Systems," Proceedings of IEEE, Vol. 89, no. 10, October 2001, pp. 1355–1367.

[2] R.T Collins, A.J. Lipton, and T. Kanade, "Introduction to the Special Section on video surveillance," IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol. 22, n. 8, pp. 745–746, August 2000.

[3] G.L. Foresti, P. Mahonen, and C.S. Regazzoni, Multimedia Video-Based Surveillance Systems: from User Requirements to Research Solutions, Kluwer Academic Publishers, 2000.

[4] D.N. Jayasimha, S.S. Iyengar, and R.L. Kashyap, "Information integration and synchronization in distributed sensor networks," IEEE Transactions on System, Man, and Cybernetics, Vol. 21, n. 5, pp. 1032–1043, Sept/Oct 1991.

[5] A. Knoll and J. Meinkoehn, "Data fusion using large multi-agent networks: an analysis of network structure and performance," in Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 113–120, Las Vegas, NV, Oct. 2–5 1994.

[6] H. Qi, S. Iyengar, and K. Chakrabarty, "Multiresolution data integration using mobile agents in distributed sensor networks," IEEE Transaction on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 31, No. 3, pp. 383–391, 2001.

[7] G.L. Foresti, "Real-time detection of multiple moving objects in complex image sequences," International Journal of Imaging Systems and Technology, Vol. 10, pp. 305–317, 1999.

[8] G.L. Foresti, "Object recognition and tracking for remote video surveillance," IEEE Transaction on Circuits and Systems for Video Technology, Vol. 9, No. 7, 1999, pp. 1045–1062.

[9] G.L. Foresti, C. Micheloni, and L. Snidaro, "Adaptive high order neural trees for pattern recognition," in Proceedings of the International Conference on Pattern Recognition (ICPR), Quebec City, Canada, pp. 877–880, August 2002.

[10] T. Wada and T. Matsuyama, "Multiobject behavior recognition by event driven selective attention method," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, n. 8, pp. 873–887, August 2000.

[11] G. Medioni, I. Cohen, F. Brémond, S. Hongeng and R. Nevatia, "Event detection and analysis from video streams," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, n. 8, pp. 873–889, August 2001.

[12] R. Wesson, F. Hayes-Roth, J.W. Burge, C. Stasz, and C.A. Sunshine, "Network structures for distributed situation assessment," IEEE Transactions on System, Man, and Cybernetics, Vol. 11, n. 1, pp. 5–23, January 1981.

[13] H. Qi, S.S. Iyengar and K. Chakrabarty, "Distributed Sensor Networks – A review of recent research," Journal of the Franklin Institute, Vol. 338, pp. 729–750, March 2001.

[14] L. Prasad, S.S. Iyengar, R.L. Kashyap, and R.N. Madan, "Functional characterization of sensor integration in distributed sensor networks," IEEE Transactions on System, Man, and Cybernetics, Vol. 21, n. 5, pp. 1082–1087, Sept./Oct. 1991.

[15] S.S. Iyengar, D.N. Jayasimha, and D. Nadig, "A versatile architecture for the distributed sensor integration problem," IEEE Transactions on Computers, Vol. 43, n. 2, pp. 175–185, February 1994.

[16] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade, "A system for video surveillance and monitoring," Proceedings of the IEEE, Vol. 89, pp. 1456–1477, October 2001.

[17] K. Skiestad and R. Jain, "Illumination independent change detection for real world image sequences," Computer Vision Graphics and Image Processing, vol. 46, pp. 387–399, 1989.

[18] P.L. Rosin and T. Ellis, "Image difference threshold strategies and shadow detection," in Proceedings of the 6th British Machine Vision Conference, BMVA Press, pp. 347–356, 1995.

[19] L. Snidaro and G.L. Foresti, "Real-time thresholding with Euler numbers," Pattern Recognition Letters, Vol. 24, n. 9–10, pp. 1533–1544, June 2003.

[20] C. Micheloni, G.L. Foresti, and L. Snidaro, "A cooperative multi-camera system for video-surveillance of parking lots," in Proceedings of the IDSS conference, London, pp. 5/1–5/5, February 2003.

[21] M.E. Liggins, C.Y. Chong, I. Kadar, M.G. Alford, V. Vannicola, S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," Proceedings of the IEEE, Vol. 85, n. 1, pp. 95–107, January 1997.

[22] Z. Chair and P.K. Varshney, "Optimal data fusion in multiple sensors detection systems," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-22, pp. 98–101, January 1986.

[23] P.K. Varshney, "Distributed Detection and Data Fusion," Springer-Verlag, 1997.

[24] Y. Bar-Shalom and X. Li, "Multitarget-Multisensor Tracking: Principles and Techniques," YBS Publishing, 1995.

[25] Y. Bar-Shalom and W.D. Blair (editors), "Multitarget multisensor tracking: applications and advances Volume III," Artech House, 2000.
[26] D. Willner, C.B. Chang, and K.P. Dunn, "Kalman filter algorithms for a multi-sensor system," Proceedings of the IEEE Conference on Decision and Control, pp. 570–574, 1976.
[27] K.C. Chang, R.K. Saha, and Y. Bar-Shalom, "On optimal track-to-track fusion," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-33, n. 4, pp. 1271–1275, October 1997.
[28] K.C. Chang, Zhi Tian, and R.K. Saha, "Performance evaluation of track fusion with information filter," in Proceedings of the International Conference on Multisource-Multisensor Information Fusion, pp. 648–655, July 1998.
[29] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," IEEE Journal of Robotics and Automation, Vol. RA-3, n. 4, pp. 323–344, 1987.
[30] O.D. Faugeras, Q.-T. Luong, S.J. Maybank, "Camera Self-Calibration: Theory and Experiments," Proceedings of European Conference on Computer Vision, pp. 321–334, 1992.
[31] G. Bradski, "Computer Vision Face Tracking for Use in a Perceptual Interface," Intel Technology Journal, 2nd Quarter, 1998.
[32] D. Comanesciu, V. Ramesh and Peter Meer, "Real-time Tracking of Non-Rigid Objects using Mean Shift," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina, 2000.
[33] S.S. Blackman, "Multiple-target tracking with radar applications," Artech House, 1986.
[34] A.B. Poore, "Multi-dimensional assignment formulation of data association problems arising from multi-target and multi-sensor tracking," Computational Optimization and Applications, Vol. 3, pp. 27–57, 1994.
[35] Lipton, "Local Application of Optic Flow to Analyze Rigid vs Non-Rigid Motion," in ICCV Workshop on Frame-Rate Vision, Corfu, Greece, September 1999.
[36] Mikic, S. Santini, and R. Jain, "Tracking objects in 3d using multiple camera views," in Proceedings of ACCV, January 2000.
[37] S.L. Dockstader and A.M. Tekalp, "Multiple Camera Tracking of Interacting and Occluded Human Motion," Proceedings of the IEEE, Vol. 89, n. 10, pp. 1441–1455, 2001.
[38] A. Mittal and L.S. Davis, "M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene," International Journal of Computer Vision, vol. 51, no. 3, Feb/March 2003.
[39] G.L. Foresti and F. Roli, "Real-time recognition of suspicious events for advanced visual-based surveillance" in Multimedia Video-Based Surveillance Systems: Requirements, Issues and Solutions, Kluwer Academic Publishers, Norwell, 2000, pp. 84–93.
[40] J. Roecker and C.D. McGillem, "Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion," IEEE Transactions on Aerospace and Electronic Systems, Vol. 24(4), pp. 447–449, 1988.
[41] D.F. Rogers and J.A. Adams, "Mathematical elements for computer graphics," McGraw Hill, 1990.
[42] J.B. Gao and Chris J. Harris, "Some remarks on Kalman filters for the multisensor fusion," Information Fusion, Vol. 3, n. 3, pp. 191–201, 2002.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Advantages and Drawbacks of Multisite Radar Systems

Victor S. CHERNYAK
*Moscow Aviation Institute (State Technical University), Russia*

**Abstract.** Main advantages and drawbacks of Multisite Radar Systems are considered at the physical level without complex mathematics.

**Keywords.** Multisite (multistatic) radar systems, multiradar systems

## 1. Introduction

The fundamental idea behind Multisite Radar Systems (MSRSs) is to make more effective use of information contained in the spatial characteristics of an electromagnetic field. It is well known that an electromagnetic field scattered by illuminated targets (or radiated by signal sources) propagates through the whole space with the exception of some shielded regions. A monostatic radar extracts information from a single small region of the field corresponding to a receiving antenna aperture. In a MSRS, information is extracted from several spatially separated regions of the field. This allows improved information gathering, interference proofing, and some other important characteristics.

The transition from individual radars to MSRSs is in agreement with the general trend of modern engineering: to integrate individual technical means into systems where fundamental characteristics are enriched due to the cooperative performance and interaction between system elements.

The development of MSRSs is based not only on increasing requirements to radar information but on the significant progress in adjacent engineering fields which enhances MSRS feasibility. The most important achievements for MSRSs are in multichannel antennas with electronic beam steering, high speed digital processors and computers, transmission lines with high capacity and precise synchronization systems.

## 2. Definition and Classification

First of all, it is necessary to define the object of consideration. We define a Multisite Radar System (MSRS) as a radar system including several spatially separated transmitting, receiving and (or) transmitting-receiving facilities where information of each target from all sensors are fused and jointly processed. Thus a MSRS has two principal distinctions: several spatially separated stations and fusion (joint processing) of received target information. It is only the combination of these two particular features that gives rise to the main benefits of MSRSs.

Many different types of MSRSs are known. It is hardly reasonable to construct a unified "tree" for the MSRS classification. Instead it would be better to extract several essential attributes and classify MSRSs in accordance with them (Fig. 1).

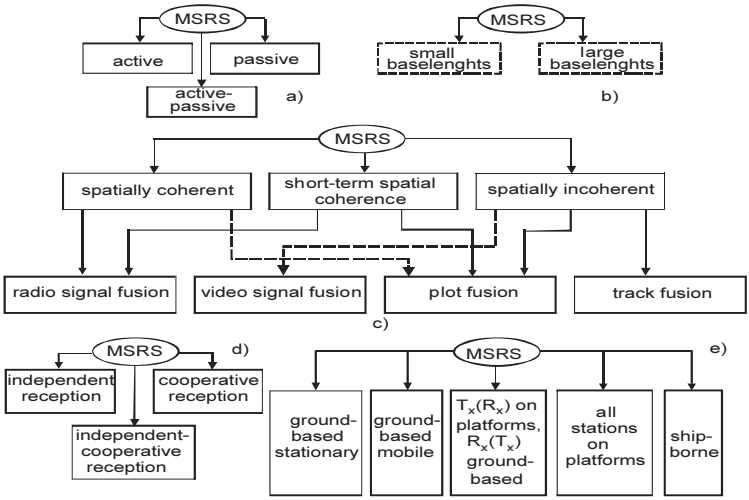



**Figure 1.** Classification of MSRSs.

Classes a), b), and e) in Fig. 1 do not require any explanation. Two of the most important features which determine the capabilities of MSRSs, are **the degree of spatial coherence and the information integration (fusion) level**. These two features are coupled. The *spatial coherence* of a MSRS means its ability to maintain a strong dependence between signal r.f. phases in separated stations and, consequently, to utilize useful information that may be contained in those phase relations. This feature represents, in effect, the phase stability of the equipment.

Depending on **the degree of spatial coherence**, all MSRSs may be placed into three classes. Interstation phase relations of *spatially coherent* MSRSs are known and maintained during time intervals much greater than the duration of signals used (usually during several hours). Equipment phase shifts may be measured periodically with the help of some reference signals. Thus, spatially coherent MSRSs require not only precise time synchronization and frequency control but phase synchronization. Such MSRSs may be considered as sparsely populated phase antenna arrays. Many stations are required for achieving an acceptable form of Spatial Discrimination Pattern – SDP (the three dimensional analogue of the Antenna Directivity Pattern – ADP). For this reason, and because of the difficulties of interstation phasing implementation, spatially coherent MSRSs are the most complicated and expensive. They are used mainly for target recognition.

In MSRSs with *short–term spatial coherence*, the equipment phase stability is maintained within time intervals of the order of used signal duration. Usually these time intervals do not exceed some fractions of a second. It may be assumed that at the beginning of each time interval of signal reception and processing, interstation phase shifts are random and mutually independent. Only time synchronization and frequency control may be necessary. A MSRS with short-term spatial coherence may comprise a few stations only. All these features of MSRSs with short-term spatial coherence de-

crease complexity and cost significantly as compared with spatially coherent MSRSs. True, the resolution and accuracy characteristics no longer depend on carrier frequency but on frequency bandwidth. However, these losses may be compensated in many cases owing to the possibility of lengthening the baselines.

In *spatially incoherent* MSRSs all interstation phase information is eliminated, e.g. by envelope detectors, before signal or data fusion. In connection with this, only time synchronization of separated stations, and maybe frequency control, are necessary. Spatially incoherent MSRSs are much simpler than MSRSs with short-term spatial coherence and, of course, than spatially coherent MSRSs. However, the elimination of phase information leads to certain power and especially information losses. Spatial incoherence of a MSRS does not rule out the possibility of temporal coherence of each station before information fusion. For example, each radar may measure Doppler frequencies and consequently target radial velocities.

According to **the information integration (fusion) level,** MSRSs may be divided into four classes. In each class both analogous and digital Data transmission lines (DTL) may be used. When *the radio signal integration level* is used, all signals, noises and interferences from spatially separated stations are subjected to joint processing. As a rule, wideband DTL (with large handling capacity) are required. If *the video signal integration level* is used, all signals, noises and interferences are also to be transmitted via DTLs, but after phase elimination in each station. This does not reduce required DTL capacity significantly but leads to certain power and especially information losses. Therefore, the video signal fusion is seldom used.

The required DTL handling capacity is reduced drastically when *the plot integration level* is used. "Primary" information processing is accomplished by each station completely including threshold comparison and parameter estimation of detected signals. Final decisions regarding the presence or the absence of a target are made by an Information fusion Centre (IFC) as a result of combining the preliminary decisions coming from all stations. This is the so called decentralized (or distributed) detection.

When *the track (trajectory) integration level* is used, not only "primary" but "secondary" information processing is accomplished in each station which ends with target track formation. Track parameter estimates coming from spatially separated stations are fused, and common tracks are built. "False" tracks are further eliminated and "true" track parameters are estimated more accurately at the fusion process. DTL handling capacity requirements are of the same order as with plot fusion.

In general, the less information lost in each station before fusion, the better the power and information performance characteristics of a MSRS, but a more complicated system and a higher DTL handling capacity is required. Connections between the Spatial coherence degree and information integration level of a MSRS are shown in Fig. 1. Actual MSRS may be of a hybrid type where information can be fused at several different levels.

One more important attribute of MSRSs may be called **the degree of autonomy of signal reception.** If a MSRS is comprised of several monostatic or bistatic radars, each radar may be designed for receiving scattered signals from targets illuminated by the transmitter of the same radar only (a dedicated transmitter). This is a MSRS with *independent (autonomous) signal reception.* The independent signal reception is usually used in spatially incoherent MSRSs with information fusion at the plot or track level. Such MSRSs are often called Netted Radars or Radar Networks. Different radars may operate in different frequency ranges.

Substantially better power and information characteristics are demonstrated by MSRSs with *cooperative signal reception* where each radar or receiving facility can receive and process echoes from targets illuminated by any radar or transmitting facility. A particular case is a MSRS with one transmitting station and several receiving stations.

A MSRS with *the independent-cooperative signal reception* contains both receiving stations exploiting transmissions of their own radars and other receiving stations, which can also exploit transmissions of other radars of the MSRS.

The described classification covers only some main distinctive features of MSRSs and, as with any classification, is not complete. At the same time, it allows to consider features and characteristics of whole classes of MSRSs.

## 3. Main Advantages of MSRS

Owing to information fusion from spatially separated stations, a MSRS presents a series of significant advantages over both monostatic radars and a collection of radars not integrated in a system.

We note here the main advantages in order to give a general idea about them. Obviously, the actual significance of either advantage depends upon the purpose of a MSRS its requirements. On the other hand, the possibilities of exploiting certain advantages are not the same for different types of MSRSs.

### *3.1. Power Advantages*

Evidently, the addition of any number of transmitting or (and) receiving stations to a monostatic radar upgrades the total power or (and) sensitivity of the system. However, MSRSs have some extra power advantages. First of all, the cooperative signal reception where each receiving station (or radar) can exploit the transmission energy of all transmitting stations (radars) enjoys significant power benefit.

If effective baselengths are sufficiently large, scattered signal fluctuations are statistically independent at different receiving stations. Then information fusion may lead to an additional power gain due to fluctuation smoothing, especially if high detection probabilities are required (Fig. 2).

Simultaneous target observation from different directions makes it possible to defeat Stealth technologies.

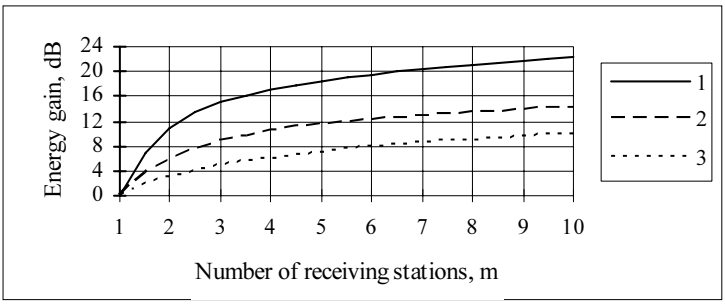



**Figure 2.** Energy gain in a MSRS with one transmitting and $m$ receiving stations. $P_{fa} = 10^{-4}$;1: $P_d = 0.99$, 2: $P_d = 0.9$ (spatially independent fluctuations); 3: coherent summation.

### *3.2. High Accuracy of Target Position Estimation*

Target position determination by usual monostatic radars is much less accurate in the cross range direction than in the down range direction, especially for distant targets. A MSRS allows estimating all three target coordinates through range measurements from several spatially separated monostatic radars or through range-sum measurements relative to several spatially separated transmitting and receiving stations. Fig. 3 shows sections of two error ellipsoids obtained as a result of target position measurement by each of the two radars. Each ellipsoid in 3D space is usually flattened like a "pancake." The intersection of those ellipsoids may represent a resultant error centroid after information fusion (joint processing) from two radars.

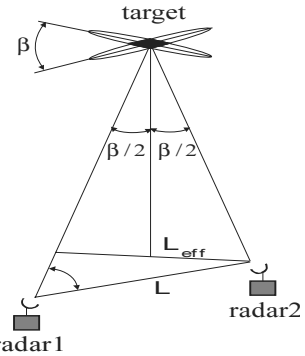



**Figure 3.** Increase of angular accuracy.

It can be seen, that there is a noticeable gain in the target position estimation accuracy mainly due to the range measurements. It may be considered that range measurements increase the angle coordinate estimate accuracy as compared with a monostatic radar. For rough calculations of angle accuracy provided by range measurements in a pair of radars, it is convenient to use the following approximate expression:

$$\sigma(\theta) \approx \sigma(R)\sqrt{2} / L_{eff} . \qquad (1)$$

Here $\sigma(\theta)$ is the r.m.s. error of the angle estimation in the bistatic plane passing through the target and both radars; $\sigma(R)$ is the r.m.s. error of range measurements (assuming these errors to be statistically independent in different radars with equal r.m.s. value); $L_{eff}$ is the effective baselength. Expression (1) has been obtained under the condition of large target range/effective baselength ratio ($R/L_{eff} >> 1$) but may be used if $R/L_{eff} > 3–4$.

It follows from (1) that with range measurements of high accuracy (i.e. if wideband signals are used) and if effective baselengths are sufficiently large, the r.m.s. error may be much less than that of a usual bearing measurement by a monostatic radar.

***Example 1.*** Let $\sigma(R)$ = 5 m, $L_{eff}$ = 30 km. Then (1) yields $\sigma(\theta)$ = 0.8′.

This is a much better angular accuracy than that of a usual monostatic radar.

This feature of MSRSs permits large and expensive radar antennas to be replaced (in certain cases) by small, weakly directional antennas without accuracy losses of target position location.

### *3.3. Possibility of Target Velocity Vector Estimation by the Doppler Method*

Doppler frequency shift measurements at several spatially separated stations allow one to estimate target radial velocities relative to these stations and calculate the velocity vector of a target. This may be of great importance for accurate target tracking, especially along the maneuver portions of target paths, for ballistic target observation when trajectory parameters of a target are to be estimated with high accuracy and in a minimal time interval, etc. In the simplest system comprising two radars with the baselength $L$ between them and with independent signal reception the measured Doppler frequencies (DFs) are $F_{D1} = 2\mathbf{v}\mathbf{r}_1/\lambda$ and $F_{D2} = 2\mathbf{v}\mathbf{r}_2/\lambda$ where $\mathbf{v}$ is the target velocity vector; $\mathbf{r}_1$, $\mathbf{r}_2$ are the unit vectors directed from the target to radar 1 and radar 2, respectively. If $\mathbf{v}$ lies in the plane passing through the target and both radars or if it is a projection of the velocity vector onto this plane, simple approximate equations may be easily obtained from Figure 4 for the r.m.s. errors of the radial and tangential (in the same plane) velocity estimation:

$$\sigma(V_R) \approx (\lambda / 2\sqrt{2})\sigma(F_D);$$
$$\sigma(V_\tau) \approx (\lambda / \sqrt{2})(R / L_{eff})\sigma(F_D). \qquad (2)$$

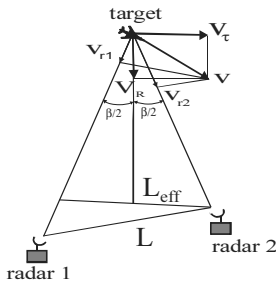



**Figure 4.** Target velocity vector estimation.

Here $\sigma(F_D)$ is the r.m.s. error of DF measurements by each radar (assumed equal); $\lambda$ is the wavelength; $R$ is the target range; $L_{eff}$ is the effective baselength as in (1). The approximate equalities of (2) are valid under the condition $R/L_{eff} >> 1$, when $\cos(\beta/2) \approx 1$, $\sin(\beta/2) \approx L_{eff}/2R$.

***Example 2.*** Let us consider the same pair of radars as in *Example 1* with $L_{eff}$ = 30 km. and $\lambda$ = 0.1 m. Let Doppler frequency be measured with the help of a coherent pair of pulses transmitted with the time interval $T$ = 5 ms. The following equation is valid for the r.m.s. frequency error:

$$\sigma(F_D) = \sigma(\varphi)\sqrt{2} / 2\pi T \qquad (3)$$

where $\sigma(\varphi)$ is the r.m.s. error of each pulse phase measurement. Let $\sigma(\varphi)$ = 7.2° = 0.02·2$\pi$ . Then the radial component according to (2), (3) is $\sigma(V_R)$ ≈ (0.1/2$\sqrt{2}$ )· (0.02·2$\pi\sqrt{2}$ /4$\pi$ *T*) = 0.2 m/s. For the target range *R* = 300 km the tangential velocity $\sigma(V_\tau)$ ≈ 2$\sigma(V_R)$ ·(*R*/$L_{eff}$) = 2·0.2·(300/30) = 4 m/s. It is seen that for the fixed DF measurement accuracy, $\sigma(V_\tau) >> \sigma(V_R)$ by the factor $1/tan(\beta/2)$ ≈ 2*R*/$L_{eff}$. However, such errors are usually by several orders of magnitude smaller than what may be achieved in a monostatic radar by differentiating angle measurements for the same observation time T = 5 ms.

By estimating the speed of Doppler shift variations or by differentiating the velocity vector components one can obtain the target acceleration vector.

The use of Doppler velocity and acceleration estimates for target tracking increases track accuracy and general quality of the tracking process.

### *3.4. Capability to Measure Three Coordinates and Velocity Vector of Radiation Sources*

It is well known, that both monostatic and bistatic radars can only determine Directions Of signal Arrival (DOAs) in a passive mode, i.e. bearings of radiation sources. Unlike those radars, MSRSs can obtain three coordinates and their derivatives. This may be achieved by triangulation, or hyperbolic methods, or their combination. The triangulation method determines the position of a radiation source in 3D space as the intersection of DOAs from spatially separated receiving stations. The hyperbolic method determines the position of a source as the intersection of hyperboloids of revolution which have their foci at receiving stations. A fixed Time Difference Of signal Arrival (TDOA) at a pair of stations (corresponding to the fixed source range difference relative to those stations) determines a hyperboloid of revolution on which surface the source must lie. (A hyperboloid of revolution is a body obtained by revolving a hyperbola about the axis passing through its foci.) The TDOA is estimated by the signal delay in one station which is necessary to maximize the mutual correlation of signals received by the two stations. It is important to note, that when range *R* of a radiation source is several times greater than the effective baselength $L_{eff}$ between stations, then angular errors of both methods are independent of the source range, so that linear cross-range errors are proportional to the range, while range errors are proportional to the squared range. A simple relationship can be written for the approximate comparison of source position fix accuracy attainable by triangulation and hyperbolic method. Under the $R/L_{eff}>>1$ condition a range difference measurement with the r.m.s. error $\sigma(\Delta R)$ (for the hyperbolic method) is approximately equivalent to a bearing measurement (for triangulation) with the r.m.s. error

$$\sigma(\theta)_{eqv} \approx \sigma(\Delta R) / L_{eff} \tag{4}$$

***Example 3.*** A pair of stations where $L_{eff}$ = 30 km and $\sigma(\Delta R)$ = 10 m is approximately equivalent to a direction finder placed in the midpoint of the baseline if its bearing accuracy (r.m.s. value) in the plane passing through the source and both stations is ≈ 3,3×10$^{-4}$ rad ≈ 1.2'. Obviously, such a small r.m.s. error is difficult to obtain by a real direction finder.

The Doppler shift measurements of the mutual correlation function of signals received by a pair of spatially separated stations from a moving radiation source make it possible to obtain an estimate of the source radial velocity difference relative to these stations. A MSRS containing four or more stations can obtain all three coordinates of the source position and all three components of velocity vector by Doppler frequency shift measurements. Using triangulation source velocity can be estimated by differentiating position estimates only.

The capability to determine three space coordinates and the velocity vector of a radiation source is an important feature for tracking such sources.

*3.5. Increase of Resolution Capability*

Let us consider active MSRSs (or an active mode of active-passive MSRSs). Fig. 5 shows a MSRS comprising two monostatic radars. It is seen that two targets are not resolved by the radar 1. Both targets fall into one resolution cell in range and angle. If the range resolution of a radar is much better than cross-range resolution (as it is usually), the angle difference between targets may be quite sufficient to resolve targets in range by radar 2.

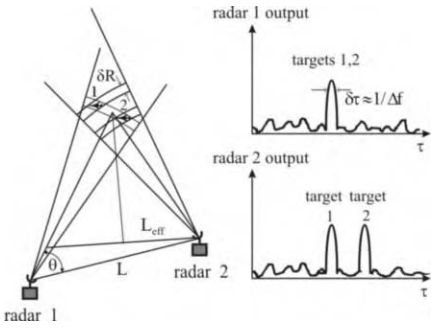



**Figure 5.** The increase of resolution capability.

This effect may be used as a capability of MSRSs to resolve targets in angle within the main beams of receiving antennas. The equivalent angular resolution capability of a system consisting of two radars may be obtained from the range resolution capability of radars $\delta R = c / 2\Delta f_s$ where $c$ is the speed of light, $\Delta f_s$ is the signal bandwidth. It is easy to show that when the target range $R$ is several times greater than the effective baselength $L_{eff}$ between radars, then

$$\delta\theta \approx c / 2\Delta f_s L_{eff} \tag{5}$$

The quantity $\delta\theta$ in (5) may be considered as the beamwidth of a "Resultant Directivity Pattern (RDP)" of a pair of radars in the plane passing through the target and both radars. When the product $\Delta f_s L_{eff}$ is large enough, then the beamwidth of a RDP is much less than the beamwidth of a usual antenna.

***Example 4.*** Let again $L_{eff}$ =30 km and $\Delta f_s$ =10 MHz. Then (5) yields $\delta\theta$ = $0{,}5\times10^{-3}$ rad ≈ 1,7'.

However, when the angle between a baseline and a target direction is small, the effective baselength decreases, which leads to a broadening of RDP beamwidth and consequently to angular resolution deterioration. Such a situation usually takes place when a ground based MSRS has to resolve in angle of elevation several targets appearing near the horizon.

In passive triangulation MSRSs, the resolution cell is determined from the intersection of antenna mainbeams. Unlike monostatic radars, two spatially separated receiving stations have resolution capability in a range which may be expressed approximately as follows (for $R/L_{eff}>>1$):

$$\delta R \approx (2R^2/L_{eff})\,\delta\alpha \tag{6}$$

where $\delta\alpha$ is the antenna beamwidth of receiving stations. It can be seen from (7) that this resolution capability is usually poor. The cross range resolution is of the order $R\,\delta\alpha$.

***Example 5.*** Let again $L_{eff}$ =30 km and $\delta\alpha = 10^{-2}$ rad $\approx 34'$. From (6) we have $\delta R \approx$ 60 km for $R$ = 300 km and $\delta R \approx$ 26,7 km for $R$ = 200 km. The cross range resolution under the same conditions yields 3 km and 2 km respectively.

In passive MSRSs where signals from spatially separated stations have undergone correlation processing (hyperbolic method) the resolution capability is determined by the extent of the envelope's mainlobe of the signal mutual correlation function. This mainlobe width can be expressed in TDOA (delay) $\delta\tau \approx 1/\Delta f_s$ as well as in range difference $\delta\Delta R \approx c/\Delta f_s$. For the RDP of two receiving stations (5) is valid if $R/L_{eff} >> 1$. When the product $L_{eff}\Delta f_s$ is sufficiently large then two sources of mutually uncorrelated radiation located within the mainbeams of receiving station antennas (and consequently not resolved by these antennas individually) can surely be resolved by the system. High "angle" resolution leads to higher range resolution compared to triangulation systems. The estimate for range resolution cell under $R/L_{eff} >> 1$ condition can be derived from (6) and (4):

$$\delta R \approx (R^2 / L_{eff}^2)\,\delta\Delta R \approx (R^2 / L_{eff}^2)(c/\Delta f_s) \tag{7}$$

where $\delta\Delta R$ denotes the resolution cell in range difference.

***Example 6.*** Let $L_{eff}$ = 30 km and the range difference resolution $\delta\Delta R$ = 30 m, which corresponds to the signal bandwidth at the inputs of correlators $\Delta f_s$ = 10 MHz. Then the angle resolution according to (5) will be $\delta\theta \approx 10^{-3}$ rad $\approx 3{,}4'$. Hence the cross-range resolution will be 0.3 km and 0.2 km for the range 300 km and 200 km, respectively. The range resolution we obtain from (7): $\delta R \approx$ 3 km for $R$ = 300 km and $\delta R \approx$ 1.3 km for $R$ = 200 km.

### *3.6. Increase of "Signal Information" Body*

The term "signal information" (unlike "co-ordinate information") usually means information extracted from target echoes and concerning geometrical, physical and other target features including characteristics of the target movement relative to its center of mass. This information is used mainly for target recognition. Since a target can be observed by a MSRS from several different directions nearly simultaneously, the total

signal information body may be substantially larger than that from a monostatic radar. Measuring amplitudes, phases and polarization parameters of signals received by spatially separated stations, one can estimate the size, the form and the relative movement (about its centre of mass) characteristics of a target of interest with higher accuracy or in a shorter time interval. Spatially coherent MSRSs with sufficiently large total antenna aperture of a collection of stations can obtain two dimensional or even three dimensional target radar images. If a MSRS does not belong to a spatially coherent type, one can obtain several range profiles from different directions. Besides, two or three dimensional target images can be obtained by echo phase difference measurements using Doppler shifts "point scatterers" of a target.

### *3.7. Increase of Jamming Resistance*

All antijamming measures (Electronic Counter Counter Measures – ECCM) used in monostatic radars can be employed in MSRS as well. Besides, MSRSs have some extra capabilities to protect against jamming.

It is easy to jam against monostatic radars with the use of highly directional ("smart") jamming with power concentrated towards radars. Jamming power density at the receiver input can be further increased by concentrating the jammers' power within the radar signal bandwidth ("spot noise jamming"). To implement directional jamming against a bistatic radar is often a more difficult problem when the direction from a jammer to a nonradiating receiving station is not known.

A much more difficult problem is to provide highly directional jamming against several stations of a MSRS with sufficient spatial separation. If a MSRS contains transmitting stations operating at several frequency ranges, and cooperative signal reception is used, such a MSRS is virtually immune to narrowband intensive jamming matched to the transmission signal frequency band. Forced spreading of a limited jammer power over a wide solid angle and frequency ranges results in lowering jamming power density against each receiving station. Such MSRSs are much less vulnerable to retrodirective deception (target echo-like) and repetitive pulse jamming. Resistance against these types of jamming can be further gained by taking into account distinctions in the TOA of target echoes and interferences.

It is well known that sidelobe noise-like jamming can be effectively cancelled in a monostatic or a bistatic radar. However, under mainlobe jamming conditions such radars cannot usually detect targets. Mainlobe jamming cancellation is accompanied by target echo suppression.

Spatially coherent MSRSs, and MSRSs with short-term spatial coherence, have a unique and important feature: the capability to cancel noise-like interferences is strongly correlated at the inputs of spaced receiving stations without suppressing target echoes. It permits the detection of targets in the presence of intensive mainlobe jamming when jammers are very close to a target or even coincide with it in space. This mainlobe jamming cancellation is carried out with the help of special Adaptive Mainlobe Jamming Cancellation Algorithms (AMJCAs). We consider a simple example showing the performance of one such algorithm.

***Example 7.*** The system considered for simulation is depicted in Fig. 6.

A monostatic radar and a spatially separated receiving station are located at the azimuths $\beta$ = 0° and $\beta$ = 180°, respectively. An aircraft moves from the range $R(0)$ = 300 km, $\beta$ = 270° at a constant height with the constant velocity $V$ = 0.8 km/s

along a straight line towards the radar system. Phases and amplitudes of target echoes at the receivers are uniformly and Rayleigh distributed, respectively. The baselength $L = L_{eff}$ has been chosen equal to 15 km to provide spatial independence of echo fluctuations at the inputs of receivers, which is a typical situation in practice. At an initial moment the aircraft has a jammer onboard (self-screening condition). Then the jammer leaves the aircraft and moves with the same velocity as the aircraft but in a slightly different direction, which provides the increase of a cross-range distance between the aircraft and the jammer of approximately 1.5 m per second. Such a model, although rather artificial, is used to reveal the algorithm performance characteristics in the most difficult cases where a jammer and a target coincide in space or are in close proximity to each other. Received signals from the receiving station are transmitted to the radar where they are jointly processed with similar signals received by the radar. The output of the jamming cancellation algorithm undergoes matched filtering, envelope detection and thresholding. The signal bandwidth is 5 MHz.

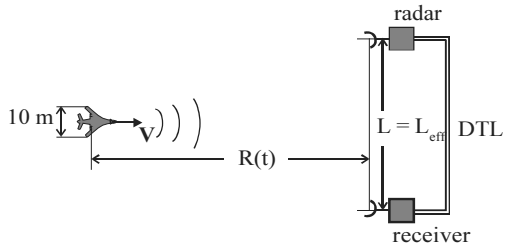



**Figure 6.** Simulation scenario.

Typical plots of the desired signal+jamming+noise versus time (sample number) at the output of the envelope detector are shown in Fig. 7, a) and b). In Fig. 7, a) the AMJCA is switched off. The target echo is perfectly masked by jamming.

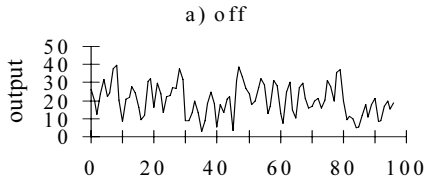



**Figure 7.** Typical plots at the output of the envelope detector.

In Fig. 7, b) the same time interval is shown but the AMJCA is switched on. The jamming cancellation permits detection of the target echo with confidence. The input and output signal-to-interference-plus-noise ratios (SINRs) are approximately −2 dB and +23 dB, respectively.

Fig. 8 demonstrates the input SINR at each receiver and the SINR at the output of the AMJCA before envelope detection as functions of the cross-range distance between the aircraft and the jammer.

It is seen that at the output of the jamming cancellation algorithm the aircraft can be surely detected even when the jammer is onboard, i.e. when the jammer and the aircraft coincide in space (a self-screening situation).

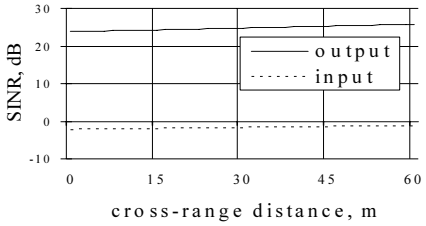



**Figure 8.** Input and output SINR.

Thanks to echo fluctuation spatial independence, jamming subtraction is accompanied by incoherent echo summation. The output averaged SINR maintains a nearly constant value when the jammer moves away from the aircraft. A slight increase of the SINRs in Fig. 8 is caused by the slight decrease of the target range.

Thus, Multisite Radar Systems can solve one of the most difficult radar problems: to detect targets under the condition of intensive mainlobe jamming.

### *3.8. Increase of Clutter Resistance*

When transmitting and receiving stations are spatially separated in a MSRS, the intersection volume of their mainbeams can be much less than the mainbeam volume of a monostatic radar. Under certain conditions it results in the essential reduction of clutter intensity at the inputs of receivers. In such cases, clutter returns from near large objects are smaller than with a monostatic radar, because these objects cannot be at short distances from both transmitting and receiving stations. Highly directional reflectors with a large RCS, such as trihedrals, are ineffective against MSRSs of this type, because they concentrate the reflected energy towards the transmitter.

At the moment when clutter falls into the intersection of the transmitting station's mainbeam and the receiving station's sidelobe or vice versa, the reduction of clutter signals depends only on the one-way antenna sidelobe pattern of the corresponding stations. Hence the low antenna sidelobe level is of great importance.

Signals from volumetric clutter (e.g., chaff) are, as a rule, mutually uncorrelated at the inputs of spaced receiving stations (or radars). Therefore, unlike mutually correlated jamming, coherent subtraction (cancellation) of clutter signals received by different stations is impossible. For enhancing detection characteristics, only target echo integration in a background of spatially uncorrelated clutter signals is possible.

All Moving Target Indication (MTI) techniques used in monostatic radars can be employed in MSRSs. Furthermore, MSRSs are free of some limitations inherent in monostatic radars: the uselessness of MTI techniques for zero and "blind" radial velocities of a target relative to a monostatic radar. A moving target cannot present zero or "blind" radial velocity to several spatially separated stations of a MSRS simultaneously. Besides, if a MSRS contains several radars (or transmitting stations), they can operate with different Pulse Repetition Frequencies (PRF) and even with different carrier frequencies. In the latter case, resonant clutter (e.g., from dipole clouds) while effective against some radars, may be ineffective against others.

A more flexible choice of transmitting signals can be made with MSRSs. In particular, pulsed signals with high PRF can be employed. These signals have broad unambiguity velocity intervals, which is important for target discrimination against clutter. Possible range ambiguities of such signals can be resolved, for example, by trian-

gulation. When transmitting stations are spatially separated from receiving ones, range ambiguities often correspond to target locations outside the mainbeam intersection volume.

### *3.9. Increase of Survivability and Reliability*

The destruction of one or even several stations of a MSRS does not lead to a total loss of information, but only to certain deterioration of performance characteristics. This feature of MSRSs is often called "the graceful degradation." This includes not only the destruction of a station as a result of a hostile physical attack, but also a failure of equipment as well. Therefore, graceful degradation means higher reliability of MSRSs. The possibility of reconfigurating MSRSs, if some stations fail, aids in achieving graceful degradation.

As mentioned above, spatial separation of transmitting and receiving stations makes it difficult to reveal the positions of "silent" receiving stations, especially when they are mobile or can be re-located in a short time interval. As a result, receiving stations of such MSRSs are essentially less vulnerable to a direct physical attack, in particular when Anti-Radiation Missiles (ARMs) are used to home in on a source of radiation. For the protection of transmitting stations, several techniques were suggested, including removal at a far enough distance from a dangerous zone, for example from a border or battle line, the use of Low-Probability-of-Intercept (LPI) probing signals, decoy transmitters, several netted transmitters with irregular switching between them ("winking" transmitter mode).

Survivability of MSRSs can be further enhanced if signal and data processing decentralization is used.

## 4. Main Drawbacks of MSRSs

The main drawbacks of MSRSs are, as a rule, the additional difficulties that must be overcome when creating a MSRS. They may be considered as the price that must be paid for all the advantages described in the previous Section. We will now briefly discuss these drawbacks since they are quite obvious and do not require detailed comments.

### *4.1. Centralized Control of Spatially Separated Stations*

Depending on the type of MSRS, this control can be reduced to target distribution among several groups of radars or can solve more complex tasks: coordinated scanning of space, choice of operational frequencies for different transmitters and receivers, waveforms, processing algorithms etc. If a MSRS contains mobile stations, the stations' position control may be necessary.

### *4.2. Necessity of Data Transmission Lines*

Each MSRS must contain Data transmission lines (DTLs) for signal or data transmission from spatially separated stations to the Information fusion Center (IFC). Both analog and digital DTLs may be employed but for modern systems digital DTLs are preferable. They are also used for command and control information transmission.

When plot or track integration levels are used in a MSRS, then DTLs may be of low handling capacity, not unlike telephone channels. For a radio (and video) signal integration level, wideband DTLs (of high handling capacity) are necessary. For command and control information transmission, DTLs with low handling capacity are usually quite adequate.

DTLs with required characteristics are not at present a difficult engineering problem.

### *4.3. Additional Requirements for Synchronization, Phasing of Spatially Separated Stations, Transmission of Reference Frequencies and Signals*

For information fusion in MSRSs, some kind of synchronization between different stations and an IFC is necessary. Specific requirements depend on the type of a MSRS. Highly accurate target coordinate measurements by elliptic or hyperbolic methods require precise synchronization. Synchronization errors should be no greater than a small fraction of the reciprocal value of the signal bandwidth. Though this is not a trivial problem, it has been solved in operational systems. If target coordinates or tracks measured relative to individual stations are jointly processed at IFCs, precise synchronization is not necessary.

When cooperative signal reception is used in a MSRS, then frequency and signal waveform emitted by any transmitting station (or radars) must be known at receiving stations (or at other radars). It can be achieved either by signal transmission via DTLs from transmitting to receiving stations (or between all radars) or by transmission of special commands providing alignment of the receiving stations (or radars) to correct operational frequency and signal waveform. A common reference frequency at all receiving stations is necessary for the correlation processing of received signals in passive or active-passive MSRSs. The common frequency may be transmitted to each station from a master oscillator (for instance, from an IFC) or may be obtained by using a high stable frequency standard at each station. These standards should be periodically matched to each other or to a reference standard.

In spatially coherent MSRSs, additional phasing of spatially separated stations (phase synchronization) is required, as in the maintaining of phase relationships between all stations within relatively long time intervals between adjacent phase adjustments.

### *4.4. Increased Requirements for Signal and Data Processors and Computer Systems*

This is in fact the consequence of one of the principal advantages of MSRSs: a significant increase of the total information body from a target as compared with monostatic radars.

However, there are certain special procedures that increase computational burden. First, for joint information processing at the IFC, coordinate conversion of radar data from a local coordinate system connected with each station into a common coordinate system of a MSRS is necessary. The second procedure is interstation data association between measurements (plots or tracks) obtained by different stations, on the one hand, and targets, on the other. Besides, most geometrical relationships and tracking algorithms are more complicated than for monostatic radars.

It should be noted that state-of-the-art signal processing and computer technology is quite sufficient to meet all actual requirements.

### *4.5. Necessity of Accurate Station Positioning and Mutual Alignment*

Errors of station position determination and of local coordinate axes orientation directly influence the accuracy of output information of a MSRS. They lead to systematic and (or) slowly varying errors of target location estimated by a MSRS and make the inter-station data association more difficult. Positions of both stationary and mobile stations can be determined with sufficient accuracy by using the GPS "NAVSTAR" and "GLONASS." Several methods and algorithms have been proposed for the alignment of stations (or radars) in MSRSs.

### *4.6. The Need for Target Visibility by Stations of a MSRS*

If a target is not simultaneously visible by at least several transmitting and receiving stations (or by several radars) of a MSRS, no information originating from the same target comes from spatially separated stations (or radars), and information fusion becomes impossible. This may be an important constraint for ground-based MSRS when low altitude targets are to be detected and tracked. Hilly and mountainous terrain may additionally obstruct low altitude targets.

### *4.7. MSRSs*

**MSRSs** containing several spatially separated stations (or radars), data transmission lines (DTLs), and information fusion centers (IFCs) **are usually more complex and expensive than monostatic radars**. However, a comparison of complexity and cost is correct if capabilities and performance characteristics are equal. Certain characteristics of MSRSs are not achievable in monostatic radars while the realization of some others requires a drastic increase of complexity and cost (for example, the employment of phased antenna arrays of enormous size). In some cases, a MSRS containing simple stations of the same type is less expensive than a monostatic radar with similar technical characteristics. Of course, the deployment of MSRSs is reasonable if the usual inexpensive monostatic radars cannot meet imposed requirements. In many cases significant benefits can be obtained at a low cost when a MSRS is created by integrating operational radars or by adding remote receiving stations to operational radars.

## 5. Conclusions

- Multisite Radar Systems have many important advantages in comparison with monostatic or bistatic radars and even with collections of radars not integrated into MSRSs. In many cases where stringent requirements are imposed on radar information, interference resistance, reliability and survivability, MSRSs can be a good, cost-effective solution;
- theoretical fundamentals of MSRSs have been developed on the whole [1]. Principal theoretical results are based on the general statistical detection theory, estimation theory and filtration theory. At the same time, there are many important problems which have not been fully solved. Among them are the problems of detection, parameter estimation and target tracking in multi-target environments, parameter estimation and tracking of extended targets, target

imaging, effective control and management of power and information resources of MSRSs, and some others. It is important to study the promising combinations of MSRSs with SAR, ISAR, and UWB radars;

- most MSRSs have been developed and are now under development for military applications. However, during the last years one can observe a trend toward a more intensive development of civil MSRSs and MSRSs for dual applications.
- recent practical applications of MSRSs show that in certain cases a MSRS, while superior in performance, turns out to be cheaper and simpler than a monostatic radar designed for the same purposes;
- multisite Radar Systems have good prospects for both military and civil applications. This is a technology for the XXI century.

## References

[1] V.S. Chernyak, Fundamentals of Multisite Radar Systems. Multistatic Radars and Multiradar Systems, Gordon and Breach Science Publishers, 1988.

This page intentionally left blank

# 6. APPLICATIONS AND OPPORTUNITIES FOR FUSION

This page intentionally left blank

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Monitoring Search and Rescue Operations in Large-Scale Disasters

## *A Multi-Agent Approach to Information Fusion*

Alessandro FARINELLI, Luca IOCCHI and Daniele NARDI
*Dipartimento di Informatica e Sistemisica, Università di Roma "La Sapienza,"*
*Via Salaria 113, 00198 Roma, Italy*

**Abstract.** The present contribution is concerned with designing tools to monitor a situation after a large-scale disaster, with a particular focus on the task of high-level Information Fusion within a multi-agent approach. The Multi-Agent System is based on the RoboCup-Rescue simulator: a simulation environment used for the RoboCup-Rescue competition, allowing for the design of both agents operating in the scenario and simulators for modeling various aspects of the situation, including a graphical interface to monitor the disaster site. The design of a Multi-Agent System with planning, information fusion, and coordination capabilities is described according to the agent model underlying the Cognitive Agent Development Toolkit, which is one of the outcomes of our recent research. Finally, we discuss the issues related to the evaluation of the performances achieved by Multi-Agent Systems in search and rescue operations according to the proposed approach and discuss some results obtained in a case study relative to the Umbria and Marche earthquake of 1997.

**Keywords.** Multi-agent systems, information fusion, search and rescue

### 1. Introduction

The analysis of the problems encountered in rescue operations concerning catastrophic events, specifically earthquakes, has driven the attention of the international scientific community towards the research on tools and techniques that can improve the effectiveness of rescue aids. Both in Japan and in the USA the problem has given rise to specific research programs (see for example [1,2]).

One major goal of these researches is the realization of tools supporting search and rescue operations, to be used in provisional analyses, training of personnel and field operations. The achievement of this research goal involves the methods and techniques for cooperation in a scenario with multiple heterogeneous autonomous agents. In this context, the RoboCup Rescue initiative [3] supported the realization of the RoboCup Rescue simulator [4], a basic framework for the simulation of the post-earthquake scenario, in which user designed agents interact with the simulated environment to search and rescue the victims of the disaster. The RoboCup Rescue simulator, which was initially designed for the Kobe earthquake scenario, integrates three major components: (1) modeling of the events, activated directly, or indirectly, by the main catastrophe, (2) acquisition and integration of data coming from heterogeneous sources, (3) operation resource monitoring/modeling/planning. The simulator thus provides a framework

based on a multi-agent approach (see for example [5,6]), where Information Fusion plays a significant role. In fact, according, for example, to the reference model for Information Fusion proposed in [7], the process of Information Fusion is concerned with different types of data processing that are required to obtain different kinds of information about the environment: data assessment, object assessment, situation assessment, impact assessment and process refinement.

The goal of the present paper is to provide an introduction to the multi-agent approach to Information Fusion, by adopting the RoboCup Rescue simulator as an experimental framework. The use of the RoboCup Rescue simulator requires the design of a Multi-Agent System based on an autonomous agent model (see for example [8]). Specifically, we propose an agent model which combines planning, information fusion and coordination capabilities. Such a model is the basis of a tool specifically developed at our University to design cognitive agents for many applications and, in particular, search and rescue operations.

A research issue relevant to the multi-agent approach to Information Fusion is the need for suitable frameworks and methodologies to evaluate the performance of the system. In fact, the inherent complexity of the tasks that a system is expected to perform, makes it very difficult to apply standard techniques for testing, validation and measure of performance. The RoboCup Rescue simulator provides a synthetic simulation scenario, which is suitable for the development of evaluation techniques of multi-agent approaches [9]. In the paper, we address the evaluation of performance of MAS in search and rescue [10].

Finally, the RoboCup Rescue simulator has been adopted in research projects carried out at our University [11,12] to address a specific case study relative to the Umbria and Marche earthquake (1997).

The paper is organized as follows. In the next section we provide some background on Multi-Agent Systems and Information Fusion; in Section 3 we sketch the main features of the RoboCup-Rescue simulator; in Section 4 we describe the design of the agents and their implementation; in the last section we address the experimentation carried out on the mentioned case study and discuss some outcomes of this experience and possible future developments.

## 2. Background and Approach

### *2.1. Multi-Agent Systems*

In the last ten years, agents have become a very important research topic. Many authors often consider agents, with respect to the software design process, as the descendants of objects: they both encapsulate a state and rely on the concept of message exchange. Developing software by making use of agent techniques is also referred to as Agent Oriented Programming [6]. While it is hard to provide a precise and generally accepted definition of Agent, it is generally agreed that there are some important features an agent must exhibit, such as autonomy in its actions and behaviors, the ability to cooperate with other agents and/or users, the capability to learn from other agents and/or users and the ability to "Understand" the environment, acquiring and maintaining knowledge about it (see for example [8,5,6]).

The main focus for Multi-Agent Systems (MAS) is the study of interactions that arise among active entities which are co-located in the same working environment. The

interest in MAS is due both to theoretical and practical issues: from a theoretical viewpoint many researchers argue that the same concept of intelligence as we normally intend is deeply related to interacting systems, and even more, that interaction is a necessary property for an intelligent system, mainly because intelligence is not an individual characteristic [5]. From a more practical perspective, the problems that we have to face in real world applications are often physically widely distributed (air traffic management, space applications); moreover, the complexity of such applications often requires combining local solutions to smaller sub-problems. Finally, from a software engineering perspective, designing solutions based on the interaction of specific autonomous modules entails several nice properties for the developed system, such as modularity, flexibility and re-usability.

MAS are widely applied in several areas addressing different practical problems. One significant application of MAS is the simulation of synthetic scenarios, which require the software agents to act in the simulated environment and interact with other entities in a way that resembles, as closely as possible, real life. Human support in hard real time, dangerous situations, such as disaster response or military applications, is becoming more and more interesting from the viewpoint of MAS applications. In such domains the main focus is to coordinate several agents involved in the operations facing the hard-time constraints imposed by the application.

In the following sections of this paper we will primarily focus on cooperative environments, where agents have the same goal and are not self interested. Moreover, we will consider dynamic and unpredictable environments, in which agents are embedded entities and therefore do not have a global vision of the environment. In such domains, Information Fusion becomes a fundamental issue in order to provide efficient planning/execution of activities and effective coordination among the cooperating agents.

### *2.2. Information Fusion*

Information Fusion is also a research area that has been growing significantly in the past years (see for example [7,13,14]). In particular, Information Fusion is broadening the scope of Data Fusion, by addressing complex dynamic scenarios, where the knowledge that enables decision making is gathered by a number and variety of different sources. Information Fusion is broadly used in various application fields such as defense, geosciences, robotics, health, industry and many techniques have been developed for the fusion process. In the following, we briefly address two aspects that are strictly related to MAS: the levels of the fusion process and the architecture of the fusion system.

#### *2.2.1. Fusion Levels*

The structure and the nature of the fusion process are thoroughly investigated by the literature concerning Information Fusion. The main notion in this respect is the *level* of fusion. Basically, data can be fused at three levels (see [14]): *signal*, *feature* and *symbol* level.

The choice of fusing data at a certain level is strictly related to the purpose and characteristics of data fusion and strongly influences the techniques adopted for the fusion. When the fusion is made at symbol level, the objects to be combined are a logical representation of data; therefore to fuse data at the symbol level in most cases logical operators are used. When the fusion is made at signal level, data are directly extracted from sensors and no intermediate representation is given; the techniques used to

fuse this kind of data are taken from signal processing. Finally, feature level fusion involves some kind of preprocessing of raw data; aggregating raw data in some kind of higher-level representation allows to avoid noisy data coming from the sensors measurement and to reduce the data communication payload. Several efforts have been made to approach the problem of Information Fusion focusing on the level of fusion, and to integrate different levels of Information Fusion in coherent architectures [15]. In a MAS approach, Information Fusion can be performed at different levels depending on the chosen representation. In the synthetic scenarios addressed in this paper, feature and symbolic levels are normally considered, while in other applications, such as sensor networks [16,17], the signal level may be relevant.

### *2.2.2. Fusion Architecture*

The design and development of system architectures is a central issue in the design of a fusion system. The architectures proposed in the literature may be grouped into three categories: Centralized, Hierarchical and Distributed.

The centralized approach to the development of fusion architectures is the most popular in the literature due to the fact that centralized fusion can be characterized as a well-defined problem. Data collected from all the sensors are processed in a single central unit that performs the fusion task. This approach is optimal when there are no communication problems (bandwidth, noise) and the central unit has enough computational resources to perform fusion of data. Most fusion algorithms have been developed for the centralized fusion architecture, and many applications have been realized using this approach.

However, in recent years, distributed and hierarchical approaches are becoming more popular, due to the wider availability of communication technology. Hierarchical fusion architectures are based on different layers of nodes: at the lowest layer, fusion nodes collect data from sensors to perform a first fusion process on these data, sending their results to a higher layer of fusion nodes. Each of the higher layer nodes collects the results of fusion from lower layers and performs a different fusion process among them. The overall architecture can be seen like a tree where each node is a fusion node and the leaves of the tree are the sensors (see for example [18]).

Finally, distributed architectures differ from hierarchical ones in the topology of the fusion nodes. Also in this case, each node performs locally a fusion process and sends the results to other fusion nodes. However, in distributed architectures there are no hierarchical layers, but each fusion node can communicate with every other fusion node. The connections are thus arbitrary and the overall architecture can be represented as a graph of fusion nodes. A distributed approach to Information Fusion is very frequent in agent-based systems, see for example [16,17]. As compared with the centralized ones, distributed and hierarchical architectures have the following advantages:

- lighter processing load at each fusion node, due to the distribution over multiple nodes;
- lower communication load, due to the reduced amount of data to be communicated;
- faster user access to fusion results, due to reduced communication delay.

On the other hand, distributed or hierarchical architectures require the development of fusion algorithms that are specialized for those architectures.

## 3. A Simulation Environment

A simulation environment for emergency scenarios is an extremely useful tool for evaluating emergency plans and for decision support. In a recent research project [12] we have analyzed the tools available to the Italian Fire Department: while they are up-to-date with the current technologies, they suffer from some limitations. First, the tools used are often realized as closed applications: they are not integrated with similar applications, or other kinds of systems, which bring relevant information to plan rescue actions. Moreover, the tools that the current technology offers to plan interventions are mainly static and they are not designed to continuously acquire information about changes in the situation and, thus, they are not able to supply the desired support where the dynamics of the events need a continuous update both of the strategy and of the intervention plan. Finally, another technological issue is the difficulty of integrating different provisional models connected to different phenomena, such as fire, house and building damages, disruption of roads, electricity, water supply, gas and other infrastructures, movement of refugees, status of victims, hospital operations, etc.

For some of the above-mentioned situations simulators have already been developed, but most of them are not dynamic and, most of all, they are not integrated. For example, existing fire simulators model this process as stochastic heat propagating and fire catching with threshold functions over static terrain, but damages of buildings and effort of fire-fighting are not considered; the few simulators which can predict road blockage are not coupled with other simulators; the simulators that, using Artificial Life approaches, model the behavior of refugees, use a static knowledge of the terrain, while a dynamic one should be considered (the disaster continuously changes the ground surface). For all these reasons, only a comprehensive simulator for large-scale disaster rescue makes it possible to compare numbers of different approaches, so that strategies and tactics, which best improve the quality of the intervention, can be chosen.

A simulator with the above-mentioned features is a fundamental tool to develop real-time systems, which allows for monitoring and planning rescue operation. The availability of the RoboCup Rescue simulator [19] represents a great chance to develop and apply the methods and techniques for action planning in a coordinated MAS, referring to a real context much larger in size and complexity with respect to those considered until now.

The RoboCup-Rescue simulator has a distributed architecture, formed by several modules, each of them being a separate process running in a workstation on a network. For a detailed description of the simulator the interested reader can refer to [4].

Below, we summarize the main components of the simulator:

- *Geographic Information System* – The GIS module holds the state of the simulated world. Before simulation begins, it is initialized by the user in order to reflect the state of the simulated area at a given time, then it is automatically updated at each simulation cycle by the kernel module;
- *kernel* – This module is connected to any other module. At each step it collects the action requests of the agents and the output of the simulators, merging them in a consistent way. Then the kernel updates the static objects in the GIS and sends the world update to all the connected modules;
- *simulators* – Fire-simulator, Collapse-simulator, Traffic-simulator, etc. are modules connected to the Kernel, each one simulating a particular disaster

feature (fire, collapses, traffic, etc.). At the beginning of every simulation cycle, they receive from the kernel the state of the world; then, they send back to the kernel the pool of GIS objects modified by the simulated feature (for example, a pool of burned or collapsed buildings, obstructed roads, etc.);
- *agents* – agent modules are connected to the kernel and represent "intelligent" entities in the real world, such as civilians, police agents, fire agents, etc. They can do some basic actions, such as extinguishing a fire, freeing obstructions from roads, talking with other agents, etc. Agents can also represent non-human entities: for example they can simulate a police station, a fire station, an ambulance-center, etc;
- *viewers* – their task is to get the state of the world, communicating with the Kernel module, and graphically displaying it, allowing the user to easily follow the simulation progress.

In order to use the RoboCup-Rescue simulator two aspects must be considered: 1) the model of the area on which we want to run disaster simulations; 2) the model of the agents involved in the intervention. These two issues are described in the next paragraph, while the next section describes in detail a framework for developing MAS in this environment.

### *3.1. World Model*

The world model adopted in the RoboCup-Rescue simulator is somewhat minimal, but it could be easily extended to fit real scenarios more closely. It deals with three main entities: *buildings*, *roads* and *nodes*. The road network is described by a graph having one or more edges for each road and one node for each crossroad and for each junction between adjacent edges constituting a road. Also, a node can represent a linkage point (*access-point*) between a building and a road. Each object class (*building*, *road*, *node*) is characterized by a number of attributes describing a specific instance of the class.

Building objects represent every kind of building on the map: houses, police offices, hospitals, fire stations, ambulance centers, refugees, etc. As a result of an earthquake shock, a building can collapse and obstruct a road; moreover, a building is more or less likely to catch fire according to its constituent material; for example, a concrete building is less flammable than a wooden one. Further, buildings can have one or more floors and one or more linkage points with the surrounding roads. The main attributes of buildings are: Plant, Kind, Material, Fieriness, Brokenness, Floors, and Entrances.

Road objects are the edges of the road network graph; they represent every street, lane, tunnel, bridge, etc. in the map. A road can be partially or totally obstructed by rubble, as a consequence of the collapsing of an adjacent building. Furthermore, a road has one or more traffic lanes on each side and can have a sidewalk or not. The most relevant road attributes: Kind, Length, Width, Block, Repair-Cost, Lines-to-head/ Lines-to-tail, Sidewalk-width.

Nodes represent crossroads or linkage points between buildings and roads. Moreover, a whole road can be split into two or more adjacent edges, connected to the others by nodes. The following are the most important attributes of this class: Roads, Signal, and Signal-timing.

In order to adapt the simulator to a generic intervention area a GIS Editor [12] allows the user to define an initial rescue scenario by specifying both the static structure of the environment (e.g. buildings, roads, etc.) and dynamic information about the ini-

tial status of the world (e.g. burning buildings, blocked roads, etc.). While editing a map it is possible to generate inconsistent situations, therefore the GIS editor can also perform checks in the edited map and warn the map designer about possible inconsistencies.

### *3.2. Agents*

Agents modeled within the RoboCup-Rescue simulator are of four kinds: *firefighting* agents in charge of extinguishing fires; *medical* agents, that bring injured civilians to hospitals; *police* agents that are in charge of removing road blockages or solving traffic jam; *civilians,* that can be injured by fire or collapsing building and may need assistance.

In addition, firefighting, police and medical agents can also be modeled with a central station that acquires information from the agents and distributes directives on how to perform in the environment.

The definition of a MAS, that represents the behavior of agents during the intervention, is an important process that the user has to accomplish in order to evaluate intervention plans in the modeled area. This process can be supported by a design and implementation framework, which is described in Section 4.

Moreover, in order to evaluate the behavior of the MAS during search and rescue operations, evaluation methodologies that measure the *goodness* of the agent's activities are used. These measures are important for evaluating performances of intervention plans as well as for comparison of different approaches to a specific problem, e.g. information fusion, planning strategies, etc.

For example, the evaluation of the general behavior of the agents is given by a formula that computes the number of civilians saved, the number of fires extinguished and the number of roads cleaned from debris. This formula allows for evaluating the effectiveness of a MAS operating in a scenario and is currently used during the RoboCup-Rescue competitions. A more detailed description of evaluation methods for such systems is given in Section 5.

Another important issue in a rescue scenario is the reconstruction of the status of the environment from the information acquired by the agents. In order to evaluate such a process, we make use of "Knowledge Viewers," graphical tools that show the knowledge that a set of agents has about the environment, which is usually different from the real status of the world. This visualization is extremely useful for monitoring the evolution of the knowledge acquisition process during the mission and for evaluating the techniques of Information Fusion that have been implemented.

## 4. Development of MAS for Search and Rescue

The development of MAS for search and rescue operations can be effectively performed by taking advantage of a design and implementation framework for MAS development. The Agent Development toolkit (ADK) [20] is a tool for implementing agents for the RoboCup-Rescue simulator. Such a tool hides the agent-simulator communication details from the user.

The framework presented in this section represents an extension of the ADK, by providing planning, cooperation and information fusion capabilities. Specifically, the Cognitive Agent Development Kit (CADK) [21], briefly described here, allows the

users to design and implement agents acting in a dynamic environment for accomplishing complex tasks. The agents developed with his tool have the following characteristics: (i) they can act autonomously in the environment by selecting the actions to be performed according to the information acquired in the environment; (ii) they can communicate with each other and cooperate to achieve a common goal; (iii) they can exchange information about the environment in order to reconstruct a global situation by using appropriate information fusion techniques.

All the agents are realized with three fundamental components:

1. *Plan Executor*, that is responsible for executing a plan (i.e. appropriately execute elementary actions that the agent can perform) for accomplishing a given task; plans are stored in a plan library and can be generated by an automatic planner, as well as by using a graphical tool; the actual plan to be executed depends on the information coming from the *Coordination Manager* which is described below;
2. *Information Integrator*, that is in charge of applying a technique for fusing the information about the world coming from its own sensors and from communication by other agents;
3. *Coordination Manager*, that is responsible for analyzing the current world state and the other agents' coordination information, and choosing the agent specific goal (and thus the corresponding plan) in order to achieve a global goal for the team; the coordination protocol is distributed and thus it is robust to network failures and allows the agents to remain autonomous.

In the following, we describe these components and their relations in more details.

The agent architecture uses a *world* structure that represents the whole agent knowledge about the world. In order to grant consistency it can only be modified by the *Information Integrator*, which updates the world in a sound way, with respect to the updating sources. For instance, in the rescue domain the world is composed by the rescue world objects such as roads, agents, buildings, etc., plus some information about the agent state. At each instant the world contains the result of the integration of the information received since the starting time, and, hence, it contains the knowledge for decision making.

In order to acquire information from the environment an agent can be equipped with a large variety of sensors and/or sensing capabilities, each one providing a different type of input. The task of the information integrator is to calculate the new state of the world starting from the previous states and the incoming information. As previously mentioned, there are different levels at which the integration can be performed, depending on the properties of the data being integrated. In our approach, Information Fusion is performed at symbol and feature level, in terms of properties of the world objects, since the RoboCup-Rescue domain, in which we tested the system, is well suited for this kind of data.

According to the situation at hand an agent must decide which actions must be performed. To this end we make use of the notion of *plan*. A plan may be seen as a graph that specifies the actions an agent has to perform in order to reach a goal. Each node is a state, while edges specify the state transitions caused by action execution. A state represents a particular world configuration in terms of properties that are true in the current situation. In order to handle both action failures and dynamic environments (where some world property can change in a way that does not depend on the agent), the states have to be considered as *epistemic*. An epistemic state represents the knowl-

edge of the agent, not the real world status. An agent can act properly, when its epistemic state matches the world configuration; this can be achieved by the introduction of particular kinds of actions that increase knowledge, and allow for conditional plans as described in [22]. The plan library is a collection of plans; each of those plans is indexed by its goal. These plans may be generated either by an automatic planner or directly specified by means of a graphical tool.

The *Plan Executor* performs two tasks:

1. it receives from the Coordination Manager the goal to be reached and peeks into the plan library the plan to achieve it;
2. it executes a plan by starting or stopping primitive actions at each state, finding which edge to follow in case of a conditional branch, according to the world status. A plan switch can occur either after the recognition of a plan failure or a when the goal is changed by the coordination module.

Finally, coordination among the agents is a fundamental issue and there are many schemas that can be adopted in coordinating a MAS. It is possible to perform a fully distributed approach as well as a centralized one, or to combine both leading on a hybrid coordination schema.

In our architecture we allow both a hybrid and distributed coordination approach. Hierarchical coordination is obtained by the use of messages (commands) sent to a low level agent from a higher-level one. The commands instruct an agent on its particular goal. The high level agent performs the task allocation on the basis of the owned information and the knowledge about the usage of resources. Distributed coordination is achieved by a distributed coordination protocol that allows for a dynamic assignment of roles to the members of the team.

## 5. Experiments and Evaluation

This section provides a description of a typical use of the RoboCup-Rescue environment for performing experiments in a rescue scenario. Specifically, we describe a set of experiments performed in the rescue scenario of the city of Foligno, that has been derived from actual data from an earthquake which occurred in that region in 1997 (see also [12]).

The initial situation contains information about the magnitude of collapses, burning buildings, blocked roads and the position of the agents (civilians and rescue agents). In Figure 1 we show a 2D visualization of one initial situation as provided by the standard viewer.

This initial situation is not completely known to the central stations, they only know the map of the city and the initial position of all its agents. Moreover, the central stations communicate with each other and perform information integration and cooperation in order to collect all available information from the environment, reconstruct a global view of the environment, remove debris from the streets, extinguish fires, and rescue the civilians.

In order to evaluate the performance of a rescue system, we have defined a methodology [10], which considers not only their efficiency under normal conditions, but also their robustness under nonstandard operative circumstances, which often occur in emergency situations.

To acquire a measure of the efficiency and the robustness of a MAS, a series of simulations have been executed by varying operative conditions. These tests give a measure of the system adaptability to unexpected situations. The parameters that we have considered for variation are: (i) perception radius; (ii) number of agents; (iii) errors in the communication system. Each parameter characterizes a particular series of simulations, referred to as the *visibility test*, the *disabled agents test* and the *noisy communications test*, respectively.


**Figure 1.** The map of Foligno city in the RoboCup Rescue simulator.

The performance of a rescue MAS is measured in terms of efficiency and reliability. The efficiency is directly evaluated by the formula used in RoboCup-Rescue tournaments (2003), that takes into account the number of living agents, the remaining hit points (health level) of all agents, and the area of houses that are not burnt. The reliability describes how much system efficiency is affected by the variation of operative conditions and is evaluated with a linear regression slope formula.

Measures of efficiency and reliability of a single MAS are of little significance if not compared with the results obtained from simulations of other rescue systems. Performance comparison allows one to establish the effectiveness of a new technique over the previous ones, or over the state-of-the-art. Rarely, in these tests the same rescue system is best for both measures, since usually, sophisticated techniques that improve efficiency turn out to be less robust to nonstandard operative conditions.

Therefore, the optimality of the system is hard to cast in absolute terms. Depending on the application, the system, which offers the best score with respect to efficiency, reliability, or to a (weighted) combination of the two, may be selected. Indeed, the choice of a measure to select the best approach is a non-trivial task.

We conclude the paper with a few comments on the development of tools to support search and rescue operations in large-scale disasters. The approach described in

this paper shows a significant use of multi-agent technology to support the acquisition of information as well as the planning of activities, when there is the need to act immediately with partial information about the situation, as in a typical emergency scenario. The RoboCup-Rescue simulator has been the basis for a prototype implementation, which is currently designed for demonstration and is not intended for actual operation. However, it has the merit to show the potential benefits of an integrated approach to the simulation and monitoring of a real search and rescue scenario. While it is premature to consider the effectiveness of the tool in the management of operation, both the analysis of past scenarios as well as the training of personnel seem to be already suitable for application.

## Acknowledgements

This research was supported under contract number F61775-01-WE-030 of the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, United States Air Force Research Laboratory and under the programme Agenzia 2000 of the Italian National Research Council.

We also gratefully acknowledge the technical support of the RoboCup Rescue Project.

## References

[1] J. Llinas. "Information fusion for natural and man-made disasters." *Proc. of IEEE Int. Conf. on Information Fusion*, AnnaPolis, July 2002.
[2] H. Kitano, et al. "RoboCup Rescue: Search and Rescue in Large Scale Disasters as a Domain for Autonomous Agents Research." *Proc. of IEEE Int. Conf. on System, Man, and cybernetics (SMC99)*, Tokyo, 1999.
[3] S. Tadokoro, et al. "The RoboCup Rescue Project: a multi-agent approach to the disaster mitigation problem." *Proc. of IEEE International Conference on Robotics and Automation (ICRA00)*, San Francisco, 2000.
[4] Robocup Rescue Web Site. http://robomec.cs.kobe-u.ac.jp/robocup-rescue.
[5] J. Ferber. "Multi-Agent Systems." *Addison-Wesley*, 1999.
[6] M. Wooldridge. "An Introduction to Multi-Agent Systems." *John Wiley and Sons,* 2002.
[7] J. Salerno. Information fusion: a high-level architecture overview. *Proc. of IEEE Int. Conf. on Information Fusion*, AnnaPolis, July 2002.
[8] S. Russell and P. Norvig. "Artificial Intelligence: a Modern Approach," *Prentice Hall*, 2003.
[9] G.A. Kaminka, L. Frank, K. Arai and K. Tanaka-Ishii. "Performance Competitions as Research Infrastructure: Large Scale Comparative Studies on Multi-Agent Systems." *Journal of Autonomous and Multi-Agent Systems*, 2002.
[10] A. Farinelli, G. Grisetti, L. Iocchi, S. Lo Cascio, D. Nardi. "Design and Evaluation of Multi-Agent Systems for Rescue Operations." *Proc. of Int. Conf. on Intelligent Robots and Systems (IROS'03)*, 2003.
[11] F. D'Agostino, A. Farinelli, G. Grisetti, L. Iocchi, D. Nardi. "Monitoring and Information Fusion for Search and Rescue Operations in Large-scale Disasters." *Proc. of IEEE Int. Conf. on Information Fusion*, AnnaPolis, USA, July 2002.
[12] D. Nardi, A. Biagetti, G. Colombo, L. Iocchi, and R. Zaccaria. "Real-time planning and monitoring for search and rescue operations in large-scale disasters." *Technical Report DIS-15-2002*, University "La Sapienza" Rome, 2002. Available at http://www.dis.uniroma1.it/~rescue/.
[13] D.L. Hall and J. Llinas (Eds.). "Handbook of Multisensor Data Fusion." *CRC Press*, 2001.
[14] L. Valet, G. Mauris and P. Bolon. "A statistical overview of recent literature in information fusion." *IEEE Aerospace and Electronics Systems Magazine*, 16(3): pages 7–14, 2001.
[15] B.L. Ferrell, J.L. Cruickshank, B.J. Gilmartin, C. Massam, S.J. Fisher and F.D. Gass. "Case study methodology for information fusion interface definition." *Aerospace Conference, 2001, IEEE Proceedings.* Volume: 6, pages 3003–3015, 2001.

[16] A. Knoll and J. Meinkoehn. "Data fusion using large multi-agent networks: an analysis of network structure and performance." *Proc. of Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI '94)*, pages 113–120, 1994.
[17] V. Regis and W. Thomas. "The soft real-time agent control architecture." *Technical Report 02-14*, University of Massachussets, 2002.
[18] Du Qingdong, Xu Lingyu and Zhao Hai. "D-s evidence theory applied to fault diagnosis generator based on embedded sensors." *Proc of the IEEE Int. Conf. on Information Fusion*, 2000.
[19] A. Farinelli, G. Grisetti, L. Iocchi, S. Lo Cascio, D. Nardi. Using the RoboCup-Rescue Simulator in an Italian Earthquake Scenario. *Proc. of the 1st Int. Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster*, Padova, Italy, July 2003.
[20] Micheal Bowling. "Robocup rescue: Agent Developement Kit." Available at RoboCup-Rescue Web Site, 2000.
[21] A. Farinelli, G. Grisetti, L. Iocchi, D. Nardi. Design and Implementation of a Rescue-Agent Development Tool. *Proc. of Int. Workshop on Planning and real-time monitoring of rescue operations*, Foligno, Italy, October 2002.
[22] Luca Iocchi, Daniele Nardi, and Riccardo Rosati. Planning with sensing, concurrency, and exogenous events: logical framework and implementation. In *Proc of the Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'2000)*, Boston, USA, 2000.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Fusion of Various Methods for Resolving the Shakespeare Controversy

Mikhail B. MALYUTOV
*Department of Mathematics, Northeastern University, Boston, MA 02115, USA*

**Abstract.** Documentary and literary evidence on the authorship attribution of works traditionally ascribed to Shakespeare is complemented by several micro-style, macro-style tests and a study of the validity of anagrams deciphered in the Shakespearean canon.

**Keywords.** Micro-style, macro-style analysis, anagrams presence testing

## 1. Introduction

Controversy concerning authorship of the works traditionally attributed to W. Shakespeare dates back several centuries. A **bibliography** of material relevant to the controversy that was compiled by J. Galland in 1947 is about **1500 pages** long [1]. A comparable work written today might well be at least four times as large. Resolving the controversy would certainly aid our understanding of what the author intended to convey in his works and thus would contribute to a better insight into the history of culture. Methodology developed during this investigation could also be useful in other applications including the attribution of newly discovered non-attributed texts. The goal of this overview is to stimulate further research by scholars with diverse areas of expertise in order to resolve the Shakespeare authorship mystery. My own minor contribution concerns the existence of certain steganography in the sonnets and plausibility of longer messages hidden there. If additional incentive to undertake this study is needed, note that the Calvin Hoffman prize, presently worth about one million British pounds will be awarded to the person who resolves this controversy.

The orthodox side, consisting of those who believe the traditional figure to be the true author of these works or simply of those who find it appropriate to maintain this version, mostly keeps silent about arguments put forth against the authorship of W. Shaxpere (W.S.) from Stratford on Avon (*this one of several spellings of the name is used to distinguish the traditional figure from the as yet undecided author of the Shakespeare canon*). When not silent, the orthodox accuse the heretics of being lunatics or snobbish. A collection of their arguments can be found in [2].

## 2. Documentary and Literary Arguments

Anti-Stratfordian snobbish lunatics (including to some extent M. Twain, S. Freud, Ch. Chaplin, Ch. Dickens, B. Disraeli, J. Galsworthy, V. Nabokov, W. Whitman, R. Emerson, J. Joyce, and H. James: “divine William is the biggest and most successful

fraud ever practiced"), point to numerous documentary and literary reasons for rejecting/doubting the W.S.' authorship.

One early survey of these grave doubts in *several hundred pages* was written by a US presidential hopeful *Donnelly* [3]. Similar doubts were expressed in many subsequent books. Among the recent books, [4,5] seem to be very convincing.

By a careful analysis of available documents and literary references from Elizabethan times, Mitchell and Price apparently decide that *W.S. was an arrogant administrator and producer, shareholder and financial manager of the Globe and some other theaters, at most of minimal literacy, occasionally performing secondary scenic roles, profiteering through all possible ways including criminal ones*. In particular, there is evidence that W.S. lent money to dramatists for writing plays performed and published under his name and ruthlessly prosecuted those failing to give the money back in time. This is revealed by Mitchell and Price in their discussions of *Groatsworth of Wit* published in 1592 after the death of well-known dramatist R. Green, where apparently W.S. is called *Terence and Batillus* with the obvious meaning of appropriating somebody else's plays.

In a manuscript (http://ist-socrates.berkeley.edu/ahnelson/Roscius.html) written during W.S.'s retirement in Stratford prior to 1623 (First Folio) W.S. was called *our humble Roscius* by a local educated Stratfordian author, meaning a famous Roman who profited from special laws allowing him to hawk or sell seats in the theater, and who was not known as an actor/playwright, merely as a businessman who profited on special favor.[1]

As a Russian scholar, I knew several Russian *Terences* in Math Sciences who used their Communist party privileges to produce remarkable lists of publications "borrowed" from others, say, persons **condemned as dissidents** or enemies of the State, who were meant to be forgotten in the Soviet Union, and for whom **any reference to their work was strictly forbidden**. It is not sufficiently remembered that Elizabethan England was an *equally closed society with its ruthless censorship and persecution.*

This concise overview cannot touch on the *hundreds of grave diverse questions* raised in the books mentioned above.[2] W.S.'s authorship is hardly compatible with any of them and is *extremely unlikely* to answer *all of them* (compare with *naive Bayes classifier* discussed further). In my experience as a statistical consultant in forensic cases (especially a disputed paternity) involving DNA profiling, a much milder mismatch would be sufficient for a court to reject paternity. *Forensic* (in addition to literary) experts must play a decisive role in resolving the controversy as shown further.

The major issues for anti-Stratfordians to resolve are: whose works were published under the Shakespeare name, and why this disguise of authorship happened, and then remained hidden for such a long time.

*Francis Bacon* became the first candidate for an alternate author probably because his knowledge of vast areas of culture matched well with that shown in the Shakespeare works.

*The pioneering stylometric study* [6] *of Shakespeare contemporaries using histograms of their word-length distribution demonstrated that the Shakespearean histogram was significantly different from those of his contemporaries (including Bacon) except for being practically identical to C. Marlowe's histogram.* However, *Williams,* [7], raised some doubts about the Shakespeare-Bacon divergence of styles, pointing to the lack of homogeneity of the texts that were analyzed. This objection deserves careful statistical analysis, its cost (*hours* vs. months in the privately funded study [6]) is now minor due to availability of software and texts in electronic form.

**Figure 1.** (a) Francis Bacon, (b) Ignatius Donnelly.

Century-long fruitless mining for *cryptography* in Shakespeare, allegedly installed there by F. Bacon, and multi-million expenditures for digging the ground in search of the documents proving that F. Bacon wrote Shakespearean works, are brilliantly analyzed in [1]. The *father of American military cryptography William Friedman* and his wife started their careers in cryptography assisting the deceptive (in their opinion) Bacon's cryptography discovery in Shakespeare by E. Gallup (which was officially endorsed by General Cartier, the head of the French military cryptography in those days!). This amusing book, full of historic examples, exercises and humor, should be read by everyone studying cryptography!

Up to now, one of the most attractive alternative candidates has been Edward de Vere, 17th earl of Oxford. De Vere's life seems by many to be reflected in the sonnets and Hamlet. Both de Vere and F. Bacon headed branches of the *English Secret Service* (ESS). De Vere was paid an enormous sum annually by Queen Elizabeth allegedly for heading the *Theater Wing of the ESS*, which was designed in order to *prepare plays and actors to serve the propaganda and intelligence collecting aims of the Queen's regime.*[3] De Vere's active public support of the corrupt establishment of the official Anglican Church in the dramatic *Marprelate* religious discussions confirms him as one of the principal Elizabethan propaganda chiefs.

Other major candidates for Shakespeare authorship include R. Manners, 5th earl of Rutland, W. Stanley, 6th earl of Derby and several other members of an aristocratic Inner Circle surrounding the Queen and including F. Bacon, Edward de Vere and *Mary Sidney Herbert (who ran a literary academy at her estate for the University wits)* together with her sons. *Judging by the works that can be firmly attributed* with reasonable certainty to each of them, none was a genius in poetry.

Some from this circle might have been able to produce plots and first versions of plays but these attempts would have required a master in order to be transformed into masterpieces. Some of these people may in fact have done the *editing work on some of the Shakespeare works* (*Mary Sidney Herbert* and her sons). One should also consider

that the voluntary hiding of authorship on any of their parts seems unlikely. Due to the wide extent of the Inner Circle, authorship information would inevitably have become known to everyone. And yet, there should have been *dramatic reasons* for the true author of the plays and poems not to have claimed the works universally recognized as "immortal." Note also that the author of the works mastered more than 30,000 English words (as estimated in [10]) compared to about 3,000 words used in average by a qualified poet. He had also mastered Greek, Latin and several contemporary European languages. In addition, he must have had a profound knowledge of classical literature, philosophy, mythology, geography, diplomacy, court life and legal systems, science, sport, marine terminology and so forth.

**Figure 2.** (a) Mary Sidney Herbert, countess of Pembroke, (b) William Friedman.

After the paper [6], a famous poet, translator and playwright Christopher Marlowe emerged as one of the main candidates. In an **unprecedented** petition by Elizabethan Privy Council, Marlowe's important service on behalf of the ESS was acknowledged, and granting him a Masters Degree by Cambridge University was requested in spite of his frequent long absences.[4] ***His*** *blank iambic pentameter*, developed further in Shakespearean works, remained the principal style of English verse for several centuries. In 29, Marlowe was among the most popular (if not the most popular) London dramatists during his allegedly last 5 years.

Arraigned into custody after T. Kyd's confessions under torture, and let out on bail by his ESS guarantors, he was allegedly killed by an ESS agent in presence of another one responsible for smuggling agents to the continent at their conventional departure house in Deptford a few days after crucial evidence of Marlowe's heresy was received by the court implying an imminent death sentence. There is evidence of Marlowe's involvement in the *Marprelate* affair which made him a personal enemy of powerful ruthless archbishop Whitgift of Canterbury, who did everything possible to expose him for ages as a heretic and eliminate him. Then, two weeks after Marlowe's supposed demise, the manuscript of the poem *Venus and Adonis*, which had been *anonymously*

submitted to a publisher some months before was *amended with a dedication to the earl of Southampton* that listed for the ***very first time the name of W. Shakespeare as author***.

There are numerous documentary and literary reasons to believe that Marlowe's death was faked by his ESS chiefs (expecting further outstanding service from him) in exchange for his obligation to live and work forever after under alternate names. These arguments are shown on the informative web-site http://www2.prestel.co.uk/rey/ of P. Farey. One of them is *obvious*: spending the entirety of his last day in Deptford, Marlowe defied a strict regulation of daily reporting to the court, hence he *knew beforehand that he would never come back under his name*. Farey also reviews extracts from the sonnets and other works of Shakespeare hinting at their authorship by Marlowe *after the Deptford affair*. He gives the results of *various stylometric tests, showing that the micro-styles of Marlowe and Shakespeare are either identical, or the latter's style is a natural development of the former*. This micro-style fingerprint close relationship between the two canons would give strong evidence for Marlowe's authorship of Shakespearean work if strengthened by further comprehensive study.[5]

Some scholars believe that the ingenious propaganda chiefs of the ESS partly inspired and paid for the production of C. Marlowe and, perhaps, of some other politically unreliable dramatists, and directed it using the Shakespeare pipeline to avoid problems with censorship proceedings.

During the O. Cromwell puritan revolt in the forties-fifties of the 17th century all theaters were closed, many intelligence documents were either lost or burnt, and the revival of interest to the Shakespearean creative work came only in 18th century making the authorship attribution problematic.

## 3. Micro-Style Analysis Review

The stylometric tables in the section Stylometrics and Parallelisms, Chapter Deception in Deptford found on Farey's site, include convincing tables of word-length usage frequencies including those made by T. Mendenhall, as well as of *function words, feminine endings, run-on lines,* etc. in both Marlowe and Shakespeare *as functions of presumable time* of writing corresponding texts showing a natural progressive style enrichment.

After T. Mendenhall's pioneering work, word-length histograms were routinely used to attribute authorship in various case studies including the successful rejection of Quintus Curtius Snodgrass articles' attribution to M. Twain, as described in *Brinegar*, [10].

An extension of Farey's study that would include alternative micro-stylometric tests described further is desirable. Moreover, the micro-style of some additional works attributed to C. Marlowe by some scholars (say, the first English translation of Don Quixote) would make for an interesting statistical analysis! Evolution plots of micro-style characters are of interest.

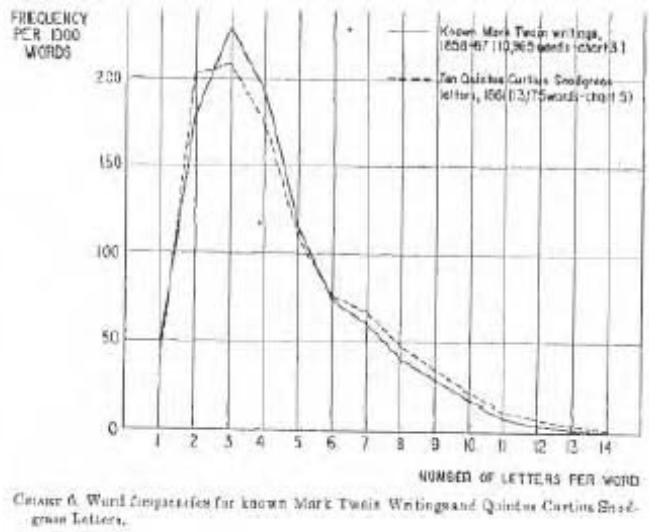



**Figure 3.** Histograms of word length in Mark Twain and Quintus Curtius.

The frequencies and histograms mentioned above characterize the stationary distribution of words or letters when an author has a large body (canon) of known work. Another popular attribution tool of this kind is a naive Bayes classifier (NBC) of *Mosteller and Wallace* [11] developed in their multi-year costly work over binary authorship attribution (Madison vs. Hamilton) of certain Federalist papers supported by Federal funding.

After fitting an appropriate parametric family of distributions (Poisson or negative binomial), they follow the Bayes rule for odds when multiplying the odds "Madison vs. Hamilton" by the sequence of likelihood ratios corresponding to the frequencies of a certain collection of relatively frequent function words, obtaining astronomical odds in favor of Madison.

This classifier presumes independence of function words usage, which is obviously **false**. This premise should be kept in mind when estimating the significance of similar studies ( for example, the NBC-attribution study of certain Shakespeare works as a byproduct of cardiac diagnosis software, well advertised by the Boston Globe on August 5, 2003, or certain Molière-Corneille controversy studies). NBC is routinely used for screening out junk mail (see [12] or Katirai, H. Filtering junk e-mail, 1999, on his web-site: http://members.rogers.com/hoomank/).

In contrast, *Thisted and Efron* [13] use the *new words usage* in a newly discovered non-attributed anapest poem "Shall I die, shall I fly?" discovered in the Yale University library, 1985. Applying this test, they neglect the obvious *enrichment of an author's language with time*. Thus the distribution of new words in a disputed work *preceding* the canon of an author and that for a text *following* the canon can be *significantly different* as can be the case for comparing Marlowe and Shakespeare works.

More promising tools (in my opinion) use the modeling of long canons as Markov chains of some order composed of English letters and auxiliary symbols. Given a non-attributed text T and a collection of firmly attributed (to author k) canons T(k) of approximately the same length for training k-th Markov model of, say, order 1, with tran-

sition probabilities $P(k,i,j)$ between symbols i and j, k=1,..., M, the log likelihood of T to be written by the k-th author is

$$\sum \log(p(k,i,j))N(i,j) + \log \pi_k(x(1)),$$

where the sum is over all i and j, $N(i,j)$ is the frequency of i followed by j, $\pi_k$ denotes the stationary probability of the k-th Markov chain, and $x(1)$ is the first symbol in T. Second order Markov chain modeling admits similar expressions for the likelihood. The author with maximal likelihood is chosen which is practically equivalent to minimizing the cross entropy of empirical and fitted Markov distributions and to minimizing the prediction error probability of a next symbol given the preceding text [14,15]. A pioneering work on Markov modeling of languages is [16]. The *power of this inference* can be approximated theoretically for large sizes of canons T(k) and T under rather natural conditions of asymptotic behavior of their sizes (Kharin and Kostevich, personal communication). A *regularization* of small transition frequencies is required. In a canon apparently written jointly by several authors (say, the King James English bible) a Hidden Markov modeling is more appropriate.

Even better attribution performance in certain tests is shown in [17], by the now very popular conditional complexity of compression minimizing classifiers discussed also in [18]. There, the idea (approximating a more abstract Kolmogorov conditional complexity concept which may appear theoretically the best authorship attributor based on micro-style) is the following: every good compressor adapts to the patterns in the text which it is compressing, reading the text from its beginning (some compressors use various extensions of the Markov modeling described above, including those based on the variants of the Lempel-Ziv algorithm). Let us define concatenated texts $C(k) = T(k)T$ as texts starting with $T(k)$ and proceeding to $T$ without stop, and corresponding compressed texts $T'(k)$ and $C'(k)$. Define the relative compressing complexity to be the difference between the lengths of compressed texts $|C'(k)| - |T'(k)|$ and choose the author minimizing it. Certainly, this definition depends on the compressor used. In the tests described in [17], the best attributing performance was shown to be that of the rarw, one of the publicly available albeit not the most popular compressor.

A comparable performance is shown by some ad hoc classification methods such as *Support Vector Machines* [19]. These methods are based on sets of characters chosen ad hoc and not unified between different applications, which does not permit a valid comparison.

I skip any discussion of methods based on grammar parsing since these methods are yet not fully automated. Also, their application for classifying archaic texts written by Shakespearean contemporaries seems doubtful.

## 4. Macro-Style Analysis

The stylometric approaches discussed up to now (their list might be easily enlarged) characterize the unconscious micro-styles of authors.

An interesting controversial comparative study of Shakespeare's and Marlowe's macro-styles[6] exists on the web-site of the late Alfred Barkov http://www.geocities.com/shakesp_marlowe/.

Barkov's analysis of the inner controversies in Marlowe's and Shakespeare works including Hamlet, well-known for a long time, enables him to claim that the texts were intentionally used to encode the story in such a way that the authors' actual messages remain misunderstood by laymen and well-understandable to advanced attentive readers. Barkov calls this style menippea, probably considering it similar to the satira menippea, a style found in many classical works and discussed by prominent Russian philosopher M. Bakhtin [20]. Menippeas often appear in closed societies, since authors tend to use Aesopian language to express their views. This language was very characteristic for Marlowe: he used his poetic genius for provoking the Queen's enemies to expose their views in response to his ambiguous statements for subsequent reporting to the ESS ([9]).

Barkov's analysis of the inner controversies in Hamlet is parallel to the independent analysis of other authors. For instance, the well-known contemporary novelist publishing under the nickname B. Akunin, presented recently his version of Hamlet in Russian (available in the Internet via the search inside the web-library www.lib.ru) with a point of view rather similar to that of Barkov, including the sinister decisive role played by *Horatio*.

## 5. Cryptography Mining

In November 2002, a Florida linguist, R. Ballantine, sent me her decipherment of Marlowe's anagrams in consecutive bi-lines (that is, pairs of lines) of most of Shakespeare and also of some other works, revealing the author's amazing life story as a master English spy both in Britain and overseas.

**Figure 4.** Roberta Ballantine.

Her unpublished novels of around 1000 pages about Marlowe's life, and overview with commentaries of around 200 pages are based also on her previous 20 years of

documentary studies. Irrespective of the authenticity of the historic information conveyed in this overview, the story is so compelling that it might become a hit of the century if supplied with dialogues and elaboration and published as fiction novels by a master story teller and screened for, say, TV serials (see several chapters of her unpublished novels and anagram examples on the web-site: http://www.geocities.com/chr_marlowe/.

Barkov claims that Ballantine's deciphered anagram texts follow the menippea macro-style of Marlowe's works. If established as true, this story will construct a bridge between golden periods of poetry in South-Western Europe and England because in it C. Marlowe is revealed as a close friend of such leading Renaissance figures as M. Cervantes and C. Monteverdi, as well as the main lucky rival in love and theater of Lope de Vega.

It is almost unbelievable that the author of Shakespearean works could pursue additional goals while writing such magnificent poetry. However, caution is warranted: *Thompson and Padover*, [21], p. 253, claim that Greek authors of tragedies used to anagram their names and time of writing in the first lines of their tragedies, which Marlowe could well learn from the best teachers in the King's school, Canterbury and University of Cambridge; a similar tradition was shared by Armenian ancient writers as *a protection against plagiarism of copyists* [22]. Also, announcing discoveries by anagrams was very popular those times (Galileo, Huygens, Kepler, and Newton among other prominent authors); anagrams were certainly used by professional spies.

It is extremely challenging to attempt a proof of cryptography content in Shakespeare after the discouraging book [1]. Moreover, serious doubts remain concerning the appropriateness of anagrams as a hidden communication (or steganography) tool, as will be discussed further on.

It is natural to consider two stages in the analysis of the validity of deciphered anagrams. The first question to address is the *existence of anagrams* in the texts. *This* we have attempted to test statistically starting from our observation that all the anagrams deciphered in Shakespeare contain various forms of Marlowe's signature at the beginning.

R. Ballantine has considered bi-lines (couplets) as suitable periods for anagramming. After deciphering an initial bi-line, she proceeds to the very next one, and so on, until the final signature. In a given play the first bi-line that begins an anagramming, is usually at the beginning of a dialogue, or after a special, but otherwise meaningless sign, a number of which appear in early editions of Shakespeare's works.

Following *Thompson and Padover*, [21], we mine for Marlowe's signature in the first bi-lines, which makes for an easier test since a disastrous multiplicity-of-decisions problem is avoided in this way. Besides, the 154 sonnets, with only a tiny part of them deciphered so far, constitute a homogeneous sample of 14 lines (7 bi-lines) each (with a single exception). Hence we chose to focus on the sonnets for statistical testing of the presence of anagrams leaving aside almost all other Shakespeare works, which allegedly also contain anagrams.

An important requirement is the careful choice of an accurate published version, which varied over time. I was fortunate to find help from an expert in the field, Dr. D. Khmelev, University of Toronto, who was previously involved in a joint Shakespeare-Marlowe stylometry study with certain British linguists.

Our first result based on the computations of Khmelev follows: the numbers of first, second, etc. bi-lines in the sonnets *containing the set of case-insensitive letters*

*M,A,R,L,O,W,E* (this "event *M*" is equivalent for this name to be a part of an anagram) are respectively 111, 112, 88, 98, 97, 101, 102 out of 154 sonnets.

Combining *two first bi-lines* in one population and the rest into the second population, the *standard test for equality of probabilities* of the event *M* occurrence in these two populations has the P-value of 0.2%. Similarly, combining the *first* bi-lines in one population, and the rest into the alternative, the P-value of homogeneity is around 3.75%.

Apparently, this anomaly in the homogeneity of bi-lines signals that the first bi-lines were specially designed to include this set of letters as part of an anagram signature (note that we mine for one signature out of many!). Another Marlowe's favorite signature, Kit M., occurs unusually more often inside the *last* two bi-lines than in the rest (homogeneity P-value is 5%).

Of course, other explanations of this statistical anomaly might also be possible. To deal with this possibility, I applied to a recognized expert in statistics on Shakespeare and on English verses in general who is with the University of Washington. Unfortunately, she turned out to be a Stratfordian, and so she chose not to reply at all.

Thus, the existence of anagrams hidden by Marlowe in Shakespeare looks rather likely.

A much more difficult task is to study the authenticity (or uniqueness) of the anagrams deciphered by R. Ballantine. This is due to a notorious ambiguity of anagrams seemingly overlooked by the great men who have used anagrams to claim priority, see above. An amazing example of this ambiguity is shown on pp. 110–111, [1], namely: **3100** different meaningful lines-anagrams in Latin have been constructed by 1711 for the salutation "Ave Maria, gratia plena, Dominus tecum."

A theory of anagram ambiguity can be developed along the lines of the famous approach to cryptography given in famous C. Shannon's *Communication Theory of Secrecy Systems* written in 1946 and declassified in 1949. An English text is modeled in it as a stationary ergodic sequence of letters with its entropy per letter characterizing the uncertainty of predicting the next letter given a long preceding text. The binary entropy of English turns out to be around 1.1 (depending on the author and style) estimated as a result of long experimentation.

Shannon showed that this value of the entropy implies the existence of around $2^{1.1N}$ meaningful English texts of large length $N$. Due to the ergodicity of long texts, the frequencies of all letters in all typical long messages are about the same, and so all typical texts could be viewed as almost anagrams of each other. Thus, the number of anagrams to a given text seems to grow with the same exponential rate as the number of English texts.

This is a discouraging result for considering anagrams as a communication tool beyond other disadvantages of anagrams, namely excessive complexity of encoding and decoding. Moreover, the aim of putative anagrams that would become known to an addressee only after the long process of publication is unclear, unless an ESS editor would pass it directly to an addressee. Again a parallelism: many M. Bulgakov's menippeas with hidden anti-Soviet content were prepared for publication by an active informer of Stalin's secret police!

There still remains hope that R. Ballantine's claim about the uniqueness of the anagrams she deciphered may prove correct due to the following reasons:

Every one of her deciphered anagrams starts with one of the variants of Marlowe signature, which restricts the remaining space on the first bi-line, and makes the com-

bination of remaining letters atypical, thereby narrowing the set of meaningful anagrams. Furthermore, the names and topics conveyed by Marlowe in the hidden text, might be familiar to his intended receiver (say, the earl of Southampton, whom he might have tutored in the University of Cambridge, or M. Sidney Herbert with her sons), who might decipher the anagrams using a type of a Bayesian inference, looking for familiar names (or using other keys unknown to us) and getting rid of possible anagrams that did not make sense for him/her.

It should also be noted that the hidden sentence on the first bi-line is usually continued on the next bi-line (run-on line) giving the decipherer additional information as to how to start deciphering the next bi-line, and so forth. Surely, these arguments are rather shaky. Only a **costly experimentation in deciphering anagrams by specially prepared experts** can lead to sound results about the authenticity of anagrams deciphered from these texts. Various types of software are available to ease the deciphering of anagrams, although it is questionable if any of them is suitable for these archaic texts.

In summary, the anagram problem in Shakespeare *remains unresolved*, although I regard it as worthy of further study.

C. Shannon himself developed an important theory for breaking codes. His *Unicity* theory specifies the minimal length of encoded messages that admit a unique decoding of a hidden message by a code breaker due to the redundancy of English. Unfortunately, his *main assumption of the key and message independence,* crucial for his results about unicity in cryptography, is *obviously not valid for anagrams*, which use special keys for each bi-line depending on the combination of letters in the bi-line.

Our statistical result on the special structure of the first bi-lines shows that the encoding (if it took place at all!) had to be iterative: if the poetic bi-line was not suitable for placing Marlowe's anagram-signature there, the line and hidden message were to be revised in order to make the enciphering possible. This is exactly a situation where *knowledge of an incredible number of English words* demonstrated by Shakespeare, could have been put to perfect use permitting flexibility in the choice of a relevant revised text!

## 6. Hopes for Genetic Evidence

It turns out that the **critical argument** against Marlowe's authorship of Shakespeare is the inquest by Queen Elizabeth's personal coroner (made in violation of several instructions) stating that Marlowe was killed on May 30, 1593. The question of the validity of this inquest is discussed by Farey and Nicholl [9] in great detail. If the inquest was faked and C. Marlowe's survival for several more years is proven, then his authorship of Shakespeare's works becomes very likely: Marlowe could have written these masterpieces with abundant features to be ascribed to him, and he had more than enough reasons to hide under a fictitious name.


**Figure 5.** (a) A fragment of the title page of the web-site www.muchadoaboutsomething.com, (b) The posthumous mask ascribed to Shakespeare.

One long-shot way to prove Marlowe's survival is as follows. A mysterious posthumous mask is kept in Darmstadt, Germany, ascribed to Shakespeare by two reasons: The Encyclopaedia Britannica states that it matches perfectly the known portraits of the bard (which are likely actually versions of Marlowe's portraits as shown brilliantly, say, on the title page of the web-site of a recent award-winning documentary film *Much ado about something*. A second reason is the following: this mask was sold to its penultimate owner-collectioner together with a posthumous portrait of apparently the same dead man in laurels lying in his bed.

The mask contains 16 hairs that presumably belonged to the portrayed person. A specialist from the University of Oxford has claimed in a personal letter to me his ability to extract mitochondrial DNA from these hairs and match it with that from the bones of W.S. (or W.S.'s mother Mary Arden) and Marlowe's mother Kate or any of Marlowe's or W.S.'s siblings. As is well-known, mtDNA is inherited strictly from the maternal side since sperm does not contain mitochondria. This study is in the planning stage, and serious legal, bureaucratic, financial and experimental obstacles must first be overcome before the study can proceed.

## 7. Conclusions

The problem of Shakespeare authorship is old, and the documents are scarce. Therefore, only some sort of statistical approach, say comparing the likelihoods of hypotheses based on the fusion of all kinds of evidence, seems feasible in trying to resolve it.

An explosion in computing power, and the emergence and development of new methods of investigation and their fusion have led me to believe that in this framework the Shakespeare controversy will eventually be resolved with sufficient conviction in spite of the four-century long history of puzzles and conspiracies.

The methods that are now being developed are promising and could also very well apply in other similar problems of authorship attribution, some of which might even have significant security applications.

## Acknowledgements

This study was proposed to the very obliged author by R. Ahlswede at the beginning of the author's two-month stay with the program Information transfer at ZIF, University of Bielefeld. The author thanks numerous colleagues (especially R. Ballantine!) for their helpful remarks and E. Haroutunian for citing the book by Abramyan. I am extremely grateful to E. Gover, D. Massey and I. Malioutov for considerable improvement of my English and to D. Malioutov for TEX-nical help.

## Notes

1. A possible visual pattern for W.S. is father Doolittle from "My fair lady," who also did little for creating the treasure of arts bearing his name.
2. See also a vast recent collection in http://www2.localaccess.com/marlowe/.
3. see www.shakespeareauthorship.org/collaboration.htm referring to Holinshed's chronicles commissioned by W. Cecil, head, Elizabethan Privy council.
4. This and many other documents about the tragic life of Marlowe are revealed in [9].
5. Imagine James Bond let out on bail and reported killed by his colleague soon after a DNA test proved his unauthorized crime. Will you believe in his death if his DNA was later repeatedly found on his victims?
6. Namely, a sophisticated architecture of their works and ambiguity of many statements.

## References

[1] Friedman, W. and Friedman, E. The Shakespearean Ciphers exposed, Cambridge University Press, 1957.
[2] Matus, I. Shakespeare, in fact, Continuum, N.Y., 1994.
[3] I. Donnelly, The great cryptogram, 1888, reprinted by Bell and Howell, Cleveland, 1969.
[4] Mitchell, J. Who wrote Shakespeare, Thames and Hudson Ltd., London, 1996.
[5] Price, D. Shakespeare's Unorthodox Biography, Greenwood Press, London, 2001.
[6] Mendenhall, T. A mechanical solution to a literary problem. Popular Science Monthly, 60, 97–105, 1901.
[7] Williams, C. Word-length distribution in the works of Shakespeare and Bacon, Biometrika, 62, 207–212, 1975.
[8] Efron, B. and Thisted, R. Estimating the number of unseen species; How many words did Shakespeare know? Biometrika, 63, 435–437, 1975.
[9] Nicholl, C. The reckoning. Chicago University Press, 1992.
[10] Brinegar, C. Mark Twain and the Quintus Curtis Snodgrass Letters: A statistical test of authorship, Journal of American Statistical Association, 58 (301), 85–96, 1963.
[11] Mosteller, F. and Wallace, D. Inference and Disputed Authorship, Addison-Wesley, Reading, 1964.
[12] De Vel, O., Anderson, A., Corney, M. and Mohay, G. Multi-Topic E-mail Authorship Attribution Forensics. Proceedings, Workshop on Data Mining for Security Applications, 8th ACM Conference on Computer Security (CCS'2001), 2001.
[13] Thisted, R. and Efron, B. Did Shakespeare write a newly discovered poem? Biometrika, **74**, 445–455, 1987.
[14] Rosenfeld, R. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. Computer, Speech and Language, **10**, 187–228, 1996, a shortened version of the author's Ph.D. thesis, Carnegie Mellon University, 1994.
[15] Khmelev, D. and Tweedy, F.J. Using Markov Chains for Identification of Writers, Literary and Linguistic Computing, **16**, No. 4, p. 299–307, 2001.

[16] Markov, A. On application of a statistical method, Izvestia (Comptes Rendus) of Imper. Academy of Sciences, Ser. VI, **X**, 1913, p. 153; 1916, p. 239 (In Russian).
[17] Kukushkina, O., Polikarpov, A. and Khmelev, D. Text Authorship attribution using letter and grammatical information, Problems of Information Transmission**, 37**, 172–184, 2001.
[18] Cilibasi, R. and Vitanyi, P., Clustering by compression, CWI manuscript, 2003.
[19] Bosch, R. and J. Smith, J. Separating hyperplanes and the authorship of the disputed Federalist Papers, American Mathematical Monthly, **105**(7), 601–608, 1998.
[20] Bakhtin, M. Problemy poetiki Dostoevskogo. English translation. University of Minnesota Press, 1984.
[21] Thompson, J.W. and Padover, S.K. Secret diplomacy; espionage and cryptography, 1500–1815, F. Ungar Pub. Co., N.Y, 1963.
[22] Abramyan, A. The Armenian Cryptography (in Armenian), Yerevan University Press, 1974.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# A Game Model for Effective Counteraction Against Computer Attacks in Intrusion Detection Systems

Edward POGOSSIAN[a,b] and Arsen JAVADYAN[b]
[a] *Academy of Science of Armenia, Institute for Informatics and Automation Problems, #24 Marshall Bagramyan Ave., Yerevan, Armenia, (epogossi@aua.am)*
[b] *State Engineering University of Armenia, Computer Software Division, #105 Teryan Str., Yerevan, Armenia, (arsen@arm.hpl.com)*

**Abstract.** A game model for dynamic protection against unauthorized access to data in network intrusion detection systems is presented. An agent based on the model stands against the intrusion by analyzing possible strategies throughout a game tree and searching for the best protection strategy. Acceptable search efficiency is achieved by a method based on the ideas of Botvinnik's "Intermediate Goals At First" and our "Common Planning and Dynamic Testing" methods. The adequacy of the model is demonstrated by its effective compatibility with system administrators and specialized programs for protection against 12 types of attacks chosen at random in a series of experiments. The model includes twelve single-level and multilevel identifier-classifiers for fusing data about the states of the base system constituents and generating a game tree relevant to those states.

**Keywords.** Intrusion detection systems, games, agents, strategy search, data fusion

## 1. Introduction

In competitive environments it is typical to be required to act according to optimal strategies. In corresponding models – *Optimal Strategy Provision (OSP)* problems, it is supposed that solutions have to deliver recommendations on how to interpret the real world and how to act in it.

We present a game OSP model for dynamic protection against unauthorized access to data in network intrusion detection systems, following the studies conducted in [1] where OSP models for chess and oligopoly market competitions were described.

Despite the fact that the protection of networks is becoming more effective, the detection of intrusions will remain an integral part of each serious secure system. There are two main categories of intrusion detection methods: the detection of anomalies and the detection of abuses.

The detection of anomalies is based on the interpretation of abnormalities as potential attacks. The advantage of this approach is in revealing abnormalities before unknown attacks occur. The disadvantages are in a large amount of false alarms, in decision making mechanisms against attacks and in its monolithic architecture, which hardly adapts to dynamic changes in the system configuration [2–4].

The abuse detection systems contains descriptions of attacks ("signatures") and searches for the correspondence to these descriptions in the tested data stream in order to discover the development of a known attack. The main advantage of such systems is in the focus on the analysis of tested data and in producing relatively few false alarms. The disadvantage is that it only detects (known) attacks which have a proper signature [5–9].

In the game OSP model an agent (decision making system) stands against an intrusion by analyzing possible strategies, and using a game tree to find the best protection strategy. In addition to known approaches of static identification, during the process of analyzing anomalies the dynamic protection model performs possible scenarios of intrusions and recommends the most effective method of protection.

Two fundamental issues on *correctness and attainability* of the OSP models arise, i.e. whether the model space of solutions includes the best solution to the problem and whether one can achieve that best solution in the model.

As it was discussed in [10], the computer chess OSP problem achieves both correctness and attainability if optimality is understood as the superiority of the program over the best human chess player. The issue of attainability for the management OSP problem coincides with the chess OSP problem and its correctness is derivable from an adequacy of models on the market and the spaces of alternative strategies for competitors.

We demonstrate the correctness and attainability of the OSP model for intrusion protection systems by its effectiveness in a series of experiments on protection against 12 types of attacks, such as Syn-Flood, Smurf, Fraggle, buffer overflow, IP-Hijacking, etc., where the model recommends decisions comparable to that of system administrators.

It was found that for making decisions comparable to experts it is sufficient to make an exhaustive, minimax search in a 3 ply game tree. The quality of decisions increases with the depth of the search, and at a depth of 5 plys the model avoided about 70% of experts' false alarms. Unfortunately, the search time increased exponentially and for depths of more than 5 it became unacceptable.

To manage the exponential time increase in the strategy search in order to achieve an acceptable level of performance we have developed a method using ideas from Botvinnik's Intermediate Goals At First (IGAF) [11] and our Common Planning and Dynamic Testing (CPDT) [10] algorithms.

The model includes over twelve single-level and multilevel identifier-classifiers that fuse data about the states of the constituents of the base system, thereafter generating a game tree relevant to those states.

Let us describe the counteraction game model and our version of the IGAF algorithm, followed by some empirical results.

## 2. Counteraction Game Model

The counteraction model represents a game between two agents with opposite interests, described by a set of states and a collection of conversion procedures from one state to another. It is assumed that the participants play in turns. Let us explain the following concepts for describing the model.

### 2.1. System Resources

System resources are, among others, CPU performance, size of the TCP buffer, and the number of incoming packages. Let $\mathbf{R = \{r\}}$ be a non-empty set of the system resources. Let **Q** be a set of *resource parameters*. Different measuring scales, such as seconds, bytes, among others, are used to measure different parameters. Let **W** be a set of possible scales. Each element $\mathbf{r \in R}$ is associated with a pair $\mathbf{<q; w>}$, where $\mathbf{q \in Q}$ & $\mathbf{q \neq \varnothing}$ & $\mathbf{w \in W}$ & $\mathbf{w \neq \varnothing}$. Such a pair, which has the corresponding system resource $\mathbf{r \in R}$, is called a *real system resource*.

### 2.2. System States

Let **Z** be a set of values taken from the interval **[**0**,** 1**]**. A *criterion function* is called an arbitrary function **f**, the range of values of which is **Z**, and **F** will denote the set of such functions **f**. A *local system resource* state on a non-empty set $\mathbf{R` \subseteq R}$ is called the value $\mathbf{l \in Z}$ of the criterion function $\mathbf{f \in F}$ on this set:

$$l = \mathbf{f(r_1, r_2, \ldots, r_k)},$$
$$\text{where } \mathbf{R` = (r_1, r_2, \ldots, r_k) \ \& \ \varnothing \neq R` \subseteq R}$$

The local state is called *normal* if $\mathbf{l = 0}$ and *critical* if $\mathbf{l = 1}$, and **L** will denote the set of local states **l**.

A *system state* on a non-empty set $\mathbf{L` \subseteq L}$ is called the value $\mathbf{s \in Z}$ of the criterion function $\mathbf{g \in F}$ on this set:

$$s = \mathbf{g(l_1, l_2, \ldots, l_n)},$$
$$\text{where } \mathbf{L` = (l_1, l_2, \ldots, l_n) \ \& \ \varnothing \neq L` \subseteq L}$$

The state is called *normal* if $\mathbf{s = 0}$ and *critical* if $\mathbf{s = 1}$, and **S** will denote the set of states **s**.

### 2.3. Conversion Procedure

Let us call an arbitrary function $\mathbf{p(s_i, s_j)}$, the ranges of definition and values of which are subsets of **R**, a *conversion procedure* of the system from the state $\mathbf{s_i}$ to $\mathbf{s_j}$:

$$\mathbf{p(s_i, s_j) : \{\{r_1, r_2, \ldots, r_k\}\} \rightarrow \{\{r`_1, r`_2, \ldots, r`_k\}\}},$$
$$\text{where } \mathbf{\{r_1, r_2, \ldots, r_k\} \subseteq R \ \& \ \{r`_1, r`_2, \ldots, r`_k\} \subseteq R}$$

**P** will denote the set of conversion procedures.

Let $\mathbf{P^a}$ and $\mathbf{P^d}$ be the sets of conversion procedures for the attacking and the counteracting sides correspondingly, and $\mathbf{P^a, P^b \subseteq P}$. Then $\mathbf{p^a(s_i, s_j) \in P^a}$ and $\mathbf{p^d(s_i, s_j) \in P^d}$ are the conversion procedures from the state $\mathbf{s_i \in S}$ to $\mathbf{s_j \in S}$ for the attacking and the counteracting sides accordingly.

The counteraction game model **M** is represented by an "AND/OR" oriented graph without cycles **G(S, P)**, where **S** and **P** are finite, non-empty sets of all states (vertices) and all possible conversion procedures (ribs) within **M** correspondingly (Fig. 1).

In order to generate a game tree relevant to the most suspicious of current intrusion resources of the base system, the model includes over twelve single-level and multi-level identifier-classifiers for the states of the base system's constituents.

The classifiers are distributed by the constituents. By fusing data from the constituents, they determine the list of resources which are suspected of being affected. This is followed by the listing of offensive and defensive actions, in the process generating the corresponding game tree, which is then used in the search for the best strategy. The final step is to apply the decision recommended by the chosen strategy.

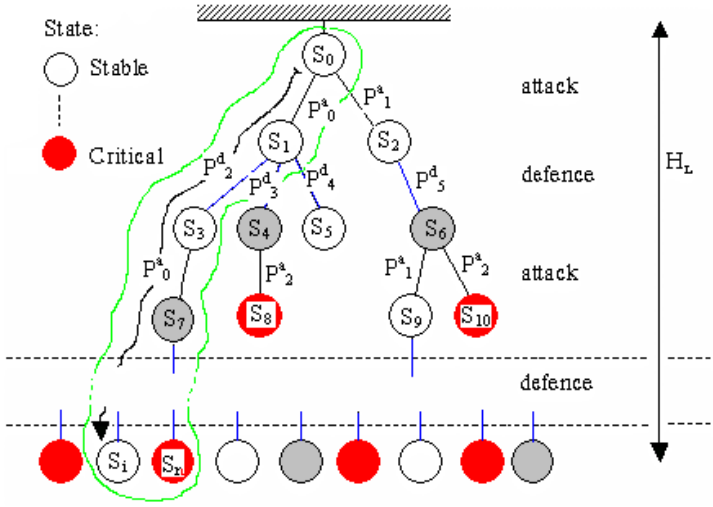



**Figure 1.** The counteraction game model.

## 3. The Minimax Algorithm for Searching Counteracting Strategy

Given a game model **M** and its related game tree, the minimax algorithm can be used to find the best counteracting strategy. The strategy found is optimal due to the exhaustive nature of the minimax algorithm.

Let **[A]** be the attacking agent and **[D]** be the defending agent.
Agent **[A]**'s goal is to bring the system into state $\mathbf{s_i} \in \mathbf{S}$, where $\mathbf{s_i} \to \mathbf{1}$.
Agent **[D]**'s goal is to bring the system into state $\mathbf{s_i} \in \mathbf{S}$, where $\mathbf{s_i} \to 0$.

Using the fact that the game model **M** is represented by an "AND/OR" tree, we will prove that player **[D]** can choose the best action strategy from the initial state $\mathbf{s_0} \in \mathbf{S}$. Then "OR" and "AND" vertices represent the states which appear after the moves of players **[D]** and **[A]** correspondingly. Let us consider that **[D]** moves first from the initial state $\mathbf{s_0} \in \mathbf{S}$ and the players move in turns. Consequently, "OR" vertices will be children vertices of "AND" vertices and vice versa. According to the suggestion made, we will consider that the initial vertex is an "AND" vertex. The terminal vertex corresponds to any state known to be the winning state for player **[D]** (Fig. 2).

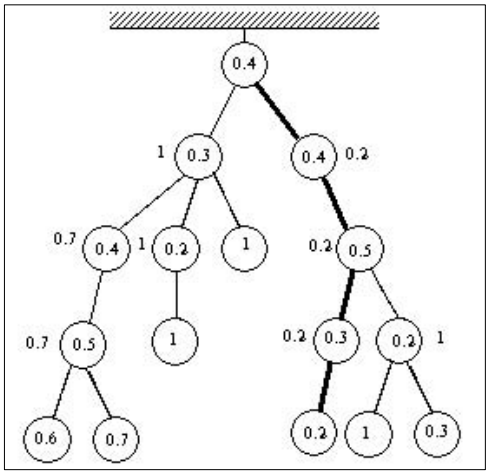



**Figure 2.** The algorithm of the best search strategy within the game tree.

Let us consider that at moment **t** a subset of **R**, i.e. the resources **{r1,…, rk}** $\subseteq$ **R**, is changing. Let **bADStep** be an element, the value of which is *true* if **[A]** moves, and *false* if **[D]** moves. $\mathbf{H_L}$ is the considered depth of a subtree during each finite step of a player from the starting vertex **i** (the root of the subtree) to the terminal vertices. $\mathbf{f_1}$ and $\mathbf{f_2}$ are criterion minimax functions for the players **[A]** and **[D]** correspondingly, where

$$\mathbf{f_1(s_1, \ldots, s_k) = max(s_1, \ldots, s_k)\ ,\ s_1, \ldots, s_k \in S}$$
$$\mathbf{f_2(s_1, \ldots, s_k) = min(s_1, \ldots, s_k)\ ,\ s_1, \ldots, s_k \in S}$$

Let the values of the criterion functions $\mathbf{f_1}$ and $\mathbf{f_2}$ denote $\mathbf{k^i_1}$ and $\mathbf{k^i_2}$ coefficients respectively, where $\mathbf{k^i_1, k^i_2 \in Z}$.

*The initial values:*
**bADStep :=** ***false;***
$\mathbf{H_L}$ **:= 1, 2, 3,….;**
**g = 0;**

*Step 1:* Determine the current state $\mathbf{s_i \in S}$. If $\mathbf{s_i = 0}$ or $\mathbf{s_i = 1}$, then it is necessary to stop looking down within the tree and to mark the vertex as closed. Otherwise, move to the next step.

*Step 2:* If the current depth $\mathbf{g = H_L}$, then move up to the closest vertex, i.e. **g := g – 1** and move to *Step 5*.

If **g = 0**, i.e. the first parent vertex has been reached, then move to *Step 6*. Otherwise, move to the next step.

*Step 3:* Determine all the resources $\mathbf{R_i = \{r_1^i, \ldots, r_k^i\} \subseteq R}$, the changing of which are the cause of switching from the previous state $\mathbf{s_j \in S}$ to the current state $\mathbf{s_i \in S}$. Move to the next step.

*Step 4:* Increase the parameter of the depth **g := g + 1**.

If **bADStep** = *false,* then pick out a subset of the defender's conversion procedures $\mathbf{P^d_i = \{ p^d_1, \ldots, p^d_k \} \subseteq P^d}$, which can change the subset $\mathbf{R_i \subseteq R}$. **bADStep :=** *true*.

If **bADStep** = *true,* then pick out a subset of the offensive's conversion procedures $\mathbf{P}^a_i = \{ \mathbf{p}^a_1, \ldots, \mathbf{p}^a_k \} \subseteq \mathbf{P}^a$, which can change the subset $\mathbf{R}_i \subseteq \mathbf{R}$. **bADStep :=** *false*.

Move to the next step.

*Step 5:* Determine the next (or the first) procedure $\mathbf{P}_i$ and run it. Return to *Step 1*.

If $\mathbf{g > 0}$ and $\mathbf{P}_i > \mathbf{P}_k$, i.e. all the conversions have been run, then move up to the closest vertex, i.e. $\mathbf{g := g - 1}$ and move to *Step 5*.

If $\mathbf{g = 0}$, i.e. the first parent vertex has been reached and moving up is impossible, then move to the next step.

*Step 6:* Coefficients assignment.

During recursive move from the low terminal vertices upward, i.e. from the children to the parents, $\mathbf{k}^i_1, \mathbf{k}^i_2 \in \mathbf{Z}$ coefficients are assigned to each parent vertex, where

$\mathbf{k}^i_1 = \mathbf{f_1}$, if **bADStep** = *true*,
$\mathbf{k}^i_2 = \mathbf{f_2}$, if **bADStep** = *false,*

This process continues to the starting vertex.

*Step 7:* Choosing the best move.

The best conversion procedure for player **[D]** is chosen by the action whose $\mathbf{k}^i_2$ coefficient has minimal minimax value. In other words, the move which brings the system to a more stable state is chosen.

*Step 8:* The end of the move. Wait for the opponent's actions.

## 4. The Scheme of Experiments

### *4.1. Gaming Based Dynamic Protection (GBDP)*

We have described a parametric class of game models inducing a corresponding class of protection systems. Are the GBDP systems adequate, or can the game model parameters be chosen in a way so that the GBDP systems perform equivalently to experts against arbitrary perturbations of the base system?

To answer the question we have tested a range of game models induced by different depths of search in the game tree while fighting against a variety of perturbations, and the results have been compared in equal environments with the performances of experts – system administrators and special protection programs.

The game models were ranged from depths 1 to 10. Six software developers having at least five years of system administrator experience and four protection programs – Real Secure (ISS), Snort, Linux and Windows standard protection software, were chosen to represent the variety of current means of protection.

The experimental perturbations have to be representative of the variety inherent in all possible perturbations. Because there is no proper specification of that variety we assume that perturbations are combinatorial, individual in nature and are unified in classes by similarity. Experimenting with a few representatives of the classes we hope to approximate the full coverage of the variety of all possible perturbations.

In the experiments we chose representatives from 12 classes of attacks. We are planning to study other types of perturbations in a subsequent period of time.

There is no unified classification for attacks. We found three intersecting groups of criteria to classify them (Table 1).

**Table 1.** Attack classification criteria

| attack classification criteria | | |
|---|---|---|
| group 1 | group 2 | group 3 |
| 1. internal | 1. information gathering | 1. remote penetration |
| 2. external | 2. unauthorized access attempts | 2. local penetration |
| | 3. denial of service | 3. remote denial of service |
| | 4. suspicious activity | 4. local denial of service |
| | 5. system attack | 5. network scanners |
| | | 6. vulnerability scanners |
| | | 7. password crackers |
| | | 8. sniffers |

The attacks chosen for the experiments [8,9] are grouped by the criteria they satisfy (Table 2).

The standard protection programs have been taken from [8,9] as well as from the sites of corporations which specialize in the area of information protection, such as SUN, Internet Security System, Snort, and others.

The criteria for comparing the GBDP systems with experts were the following.

1. *Distance to Safety* (DtS): estimate within the [0,1] interval of the distance from the current state of the system to a safe or normal state.
2. *Safety Fails* (SF): the percent of critical states the system enters during a particular experiment.
3. *Productivity* (P): the number of performance units preserved per unit of time for the system in average during an experiment. The performance units are defined in the context of the user's current focus of interest and may be measured, for example, by the number of packages or files the system processes, by the number of users served, etc.

**Table 2.** The chosen attacks chosen for experimentation grouped by criteria

| | attack name | by group1 | by group2 | by group3 |
|---|---|---|---|---|
| 1 | SYN-Flood | 1 | 3 | 3 |
| 2 | Smurf | 1 | 3 | 3 |
| 3 | Fraggle | 2 | 1(2,5) | 2(4) |
| 4 | Buffer overflow | 1(2) | 5(4) | 1(2) |
| 5 | IP-Hijacking | 1 | 1(2) | 1(8) |
| 6 | Boink | 1 | 3 | 3 |
| 7 | Ping o"Death | 1(2) | 3 | 3(2) |
| 8 | ICMP hack | 1 | 3(5,1) | 3(8) |
| 9 | Login-bomb | 1 | 2 | 7 |
| 10 | ARP-spoofing | 1 | 1(2,3) | 1(3,8) |
| 11 | HD-recycle | 2 | 3 | 4 |
| 12 | DNS-Flooding | 1 | 1(3) | 1(3,8) |

What is meant by measuring the productivity of a system? Is it trivial or is it defined in the context of particular experiments?

Each protection system was tested against all attacks and corresponding result vectors were compared and analyzed for each attack separately and by all attacks on average.

The performance of a protection system against any attack was estimated by the means of corresponding distributions of measurements by each of three criteria in the series of offense-defense pairs of actions.

All series included approximatively 180 pairs of such actions.

For each series the sampling mean of those 180 values measured by the criteria – Distance to Safety, Safety Fails and Productivity – were calculated and identified with the means of the distributions. That identification is consistent with the Central Limit Theorem applied to a large sample of measurements with unknown distribution.

### *4.2. Experiment Description*

A special tool to gather empirical data has been developed. The tool provides an ability to vary the elements of the experiments, such as estimated criteria, types of attacks, game model parameters (particularly the depth of search), etc. The investigated GBDP systems included the following components:

1. eight controlled system resource parameters: the new queue of incoming packages, the TCP connection queue, the number of processed packages, RAM, HD, working time, unauthorized access to files, system login;
2. Over twelve single-level and multilevel identifier-classifiers of the local system states;
3. 21 actions-procedures on the attacking side;
4. 25 “normal” actions/procedures of the system;
5. 82 known counteractions against actions/procedures of attacks (the database of counteractions).

Each experiment includes the emulation of the base system with/without suspicion of an attack (or any other perturbation in the system), during which the GBDP system (with the minimax or other algorithm) makes a decision about the best strategy and chooses the best action according to that strategy. The data gathered in experimenting with any attacks contain the safety estimate of the system state for each step, the actions taken by each side, and the performance of the system. All the data is saved for further use in the construction of relation charts.

To make decisions, the GBDP system, using identifiers-classifiers of the constituents of the base system, determines the following data:

1. the list of the base system’s resources that can potentially be affected by disturbances or suspicious attacks, causing critical states in the system;
2. the list of possible actions-procedures of the offensive player during the suspicious attack;
3. the list of possible “normal” actions-procedures to protect the base system;
4. the library of known procedures of possible counteractions.

Then the GBDP system functions as follows:

1. chooses at random offensive actions from the corresponding list;
2. chooses defensive actions from the corresponding list as follows:
   - by the GBDP system if it recognizes suspicious perturbations in the system,
   - randomly, if there are no perturbations;

3. processing the chosen actions in the game model to get new states for the base system;
4. returning to the first step if experiments are still in process or stop the system otherwise.

Let us illustrate the GBDP system for the SYN Flood attack.

## 5. GBDP System Against SYN-Flood Attack

### *5.1. The SYN-Flood Attack*

#### *5.1.1. TCP Connections*

How can two machines be "connected to each other over a global network?" Two machines, able to address and send packets of data to each other, negotiate a "connection agreement." The result of a successful negotiation is a "Virtual TCP Connection."

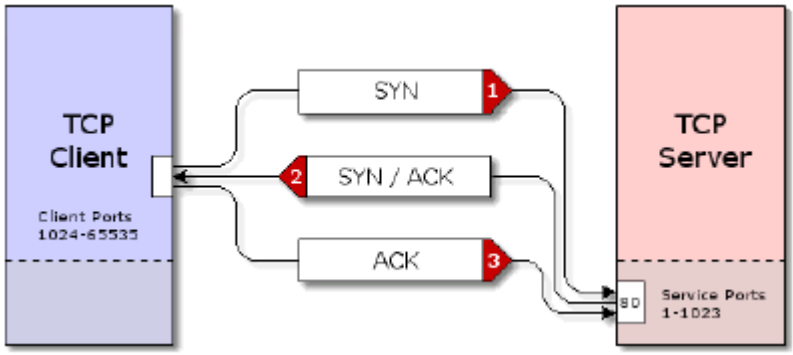



**Figure 3.** TCP Three-Way Handshake.

Individual TCP packets contain "flag bits" which specify the contents and purpose of each packet. For example, a packet with the "**SYN**" (synchronize) flag bit set initiates a connection from the sender to the recipient. A packet with the "**ACK**" (acknowledge) flag bit set acknowledges the receipt of information from the sender. A packet with the "**FIN**" (finish) bit set terminates the connection from the sender to the recipient. The establishment of a TCP connection typically requires the exchange of three Internet packets between two machines in an interchange known as the **TCP Three-Way Handshake** (Fig. 3). Here's how it works:

1) SYN: a TCP client (such as a web browser, ftp client, etc.) initiates a connection with a TCP server by sending a "SYN" packet to the server.

As shown in the diagram above, this **SYN** packet is usually sent from the client's port, numbered between 1024 and 65535, to the server's port, numbered between 1 and 1023. Client programs running on the client machine ask the operating system to "assign them a port" for use in connecting to a remote server. This upper range of ports is known as the "client" or "ephemeral" port range. Similarly, server programs running on the server machine ask the operating system for the privilege of "listening" for incoming traffic on specific port numbers. This lower port range is known as "service ports." For example, a web server program typically listens for incoming packets on

port 80 of its machine, and web-browsing clients generally send their web packets to port 80 of remote servers. Note that in addition to source and destination port numbers, each packet also contains the IP address of the machine, which originated the packet (the Source IP) and the address of the machine to which the Internet's routers will forward the packet (the Destination IP).

2) SYN/ACK: When a connection-requesting SYN packet is received at an "open" TCP service port, the server's operating system replies with a connection-accepting "SYN/ACK" packet.

Although TCP connections are bi-directional (full duplex), each direction of the connection is set up and managed independently. For this reason, a TCP server replies to the client's connection-requesting **SYN** packet by **ACK**nowledging the client's packet and sending its own **SYN** to initiate a connection in the returning direction. These two messages are combined into a single "**SYN/ACK**" response packet. The **SYN/ACK** packet is sent to the **SYN** sender by exchanging the source and destination IPs from the **SYN** packet and placing them into the answering **SYN/ACK** packet. This sets the **SYN/ACK** packet's destination to the source IP of the **SYN**, which is exactly what we want. Note that whereas the client's packet was sent to the server's service port — 80 in the example shown above — the server's replying packet is returned from the same service port. In other words, just as the source and destination IPs are exchanged in the returning packet, so are the source and destination ports. The client's reception of the server's **SYN/ACK** packet confirms the server's willingness to accept the client's connection. It also confirms, for the client, that a round-trip path exists between the client and server. If the server had been unable or unwilling to accept the client's TCP connection, it would have replied with a **RST/ACK** (Reset Acknowledgement) packet, or an ICMP Port Unreachable packet, to inform the client that its connection request had been denied.

3) ACK: When the client receives the server's acknowledging SYN/ACK packet for the pending connection, it replies with an ACK packet.

The client **ACK**nowledges the receipt of the **SYN** portion of the server's answering **SYN/ACK** by sending an **ACK** packet back to the server. At this point, from the client's perspective, a new two-way TCP connection has been established between the client and server, and data may now freely flow in either direction between the two TCP endpoints. The server's reception of the client's **ACK** packet confirms to the server that its **SYN/ACK** packet was able to return to the client across the Internet's packet routing system. At this point, the server considers that a new two-way TCP connection has been established between the client and server and data may now flow freely in either direction between the two TCP endpoints.

### *5.1.2. Abusing TCP: The Traditional SYN-Flood*

As shown in the TCP transaction diagram above, the server's receipt of a client's **SYN** packet causes the server to prepare for a connection. It typically allocates memory buffers for sending and receiving the connection's data, and it records the various details of the client's connection including the client's remote IP and connection port number. In this way, the server will be prepared to accept the client's final connection-opening **ACK** packet. Also, if the client's **ACK** packet should fail to arrive, the server will be able to resend its **SYN/ACK** packet, presuming that it might have been lost or dropped by an intermediate Internet router. But think about that for a minute. This means that memory and other significant server "connection resources" are allocated as

a consequence of the receipt of a single Internet "**SYN**" packet. Clever but malicious Internet hackers figured that there had to be a limit to the number of "half open" connections a TCP server could handle, and they came up with a simple means for exceeding those limits (Fig. 4):

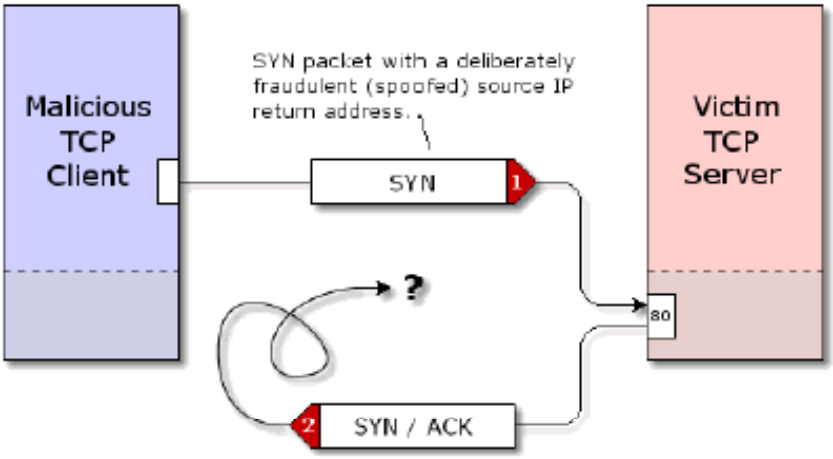



**Figure 4.** The traditional SYN-Flood.

Through the use of "Raw Sockets," the packet's "return address" (source IP) can be overridden and falsified. When a **SYN** packet with a spoofed source IP arrives at the server, it appears as any other valid connection request. The server will allocate the required memory buffers, record the information about the new connection, and send an answering **SYN/ACK** packet back to the client. But since the source IP contained in the **SYN** packet was deliberately falsified (it is often a random number), the **SYN/ACK** will be sent to a random IP address on the Internet. If the packet were addressed to a valid IP, the machine at that address might reply with a "**RST**" (reset) packet to let the server know that it did not request a connection. But with over 4 billion Internet addresses, the chances are that there will be no machine at the address and the packet will be discarded. The problem is that the server has no way of knowing that the malicious client's connection request was fraudulent, so it needs to treat it like any other valid pending connection. It needs to wait for some time for the client to complete the three-way handshake. If the **ACK** is not received, the server needs to resend the **SYN/ACK** in the belief that it might have been lost on its way back to the client. As you can imagine, all of this connection management consumes valuable and limited resources in the server. Meanwhile, the attacking TCP client continues firing additional fraudulent **SYN** packets at the server, forcing it to accumulate a continuously growing pool of incomplete connections. At some point, the server will be unable to accommodate any more "half-open" connections and even valid connections will fail, since the server's ability to accept any connections will have been completely consumed.

### *5.2. SYN-Flood, in Detail*

When SYN-Flood attack starts some changes take place in different constituents of the base system. The CBDP system using identifiers-classifiers of the constituents of the base system determines the lists of relevant resources, offensive and defensive actions followed by generation of a corresponding game tree, searching for the best strategy and applying the recommended decision.

Here are the lists of resources and actions identified by the GBDP system during the SYN-Flood attack. Integrated results of the protection are summarized in the next section.

The list of base system's resources identified by the SBDP system during the SYN-Flood attack

1. **TCPNewPacketBuffer** is the queue of new packets:
   where
   - **nTCPNewPacketBufferSize** is the maximal size of the buffer of new connections;
   - **nTCPNewPacketBufferLoadedSize** is the loaded size of the buffer of new connections;
2. **TCPBuffer** is the queue of the connections whose state is **SYN_RECV**:
   where
   - **nTCPNewPacketBufferSize** is the maximal size of the buffer of new connections;
   - **nTCPNewPacketBufferLoadedSize** is the loaded size of the buffer of new connections;
3. **TCPService** is the number of processed packets in a unit of time:
   where
   - **nTime** is the unit of time, during which the set connections number must be recalculated;
   - **nTCPPacketCount** is the number of set connections in **nTime** period;
   - **nMaxTCPPacketCount** is the maximum possible number of set connections in **nTime** period.

The list of local states classifiers activated in the SBDP system during the SYN-Flood attack

1. $\mathbf{l_1(r_1) \in L}$ – local states by parameter $\mathbf{r_1}$
   $\mathbf{f(r_1) = 1.0 - (nTCPNewPacketBufferSize - nTCPNewPacketBufferLoadedSize)/ nTCPNewPacketBufferSize;}$
2. $\mathbf{l_2(r_2) \in L}$ – local states by parameter $\mathbf{r_2}$
   $\mathbf{f(r_2) = 1.0 - (nTCPBufferSize - nTCPBufferLoadedSize)/nTCPBufferSize;}$
3. $\mathbf{l_3(r_3) \in L}$ – local states by parameter $\mathbf{r_3}$
   $\mathbf{f(r_3) = 1.0 - (nTCPPacketCount/nMaxTCPPacketCount)}$;

The list of offensive actions-procedures identified by the SBDP system during the SYN-Flood attack

1. $\mathbf{p^a_1(r_1) \in P^a}$ – sending of a packet for a new connection **SYN($r_1$)**;
2. $\mathbf{p^a_2(r_1) \in P^a}$ – sending of a packet for a new connection with double intensity **SYN2($r_1$)**;

The list of "normal" actions-procedures identified by the SBDP system during the SYN-Flood attack

1. $\mathbf{p^d_1(r_1, r_2, r_3) \in P^d}$ – sending of a synchronize/acknowledgment packet for a new connection **SYN/ACK($r_1$, $r_2$, $r_3$);**
2. $\mathbf{p^d_2(r_1) \in P^d}$ – sending of a reset packet for the requested connection **RST($r_1$);**

The list of defensive actions-procedures identified by the SBDP system during the SYN-Flood attack

1. **pd3(r1) ∈ Pd** – reallocating (grow) of a new packet buffer size **DynRealloc1(r1);**
2. **pd4(r2) ∈ Pd** – reallocating (grow) of a connection buffer size **DynRealloc2(r2);**
3. **pd5(r1, r3) ∈ Pd** – cutting down on packet life time for a new packet **CutTime1(r1, r3);**
4. **pd6(r2, r3) ∈ Pd** – cutting down on existing connection life time to **CutTime2(r2, r3);**
5. **pd7(r1) ∈ Pd** – dynamic random deleting of some packets from the new packet buffer **DynRandomFree1(r1);**
6. **pd8(r2) ∈ Pd** – dynamic random deleting of some packets from a connection buffer **DynRandomFree2(r2);**
7. **pd9(r1) ∈ Pd** – self sending of reset packet **RecursRST(r1);**
8. **pd10(r2) ∈ Pd** – using of SYN cookie **SYNCookie(r2);**
9. **pd11(r1) ∈ Pd** – decreasing of the new packet buffer size **DynRealloc3(r1);**
10. **pd12(r2) ∈ Pd** – decreasing of the connection buffer size **DynRealloc4(r2);**
11. **pd13(r1, r3) ∈ Pd** – decreasing of the waiting time of the new packet **CutTime3(r1, r3);**
12. **pd14(r2, r3) ∈ Pd** – decreasing of the waiting time of the connection record **CutTime3(r2, r3).**

## 6. Minimax vs. Experts: Results of the Experiments

In the Tables and Figures below the comparative results of the GBDP system with the minimax search algorithm vs. experts are presented. Each protection system – GBDP system for different depths of search (Mod 1–10), system administrators (Exp 1–6) and protection program systems (Aps 1–4), were tested against all 12 attacks and the corresponding means by three criteria – Distance to Safety (DtS), Safety Fails (SF) and Productivity (Prd), are allocated in corresponding tables.

Along with the means in the last column of the tables for each protection system, the sampling mean for the sample in 12 attacks (Sampling Mean) for the attacks distribution is presented.

### 6.1. GBDP System Performance

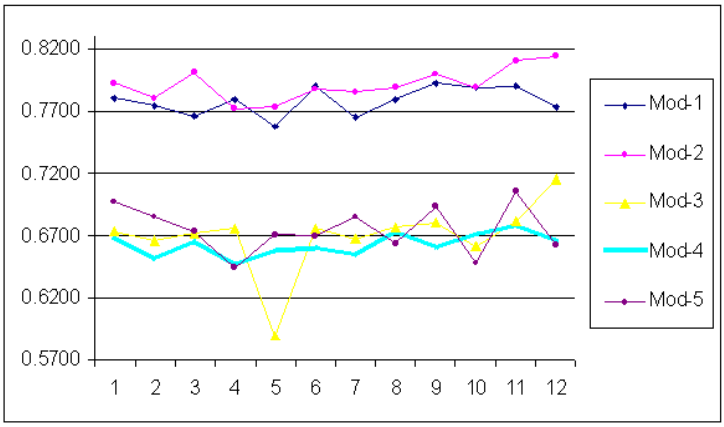



**Figure 5.** Distance to safety.

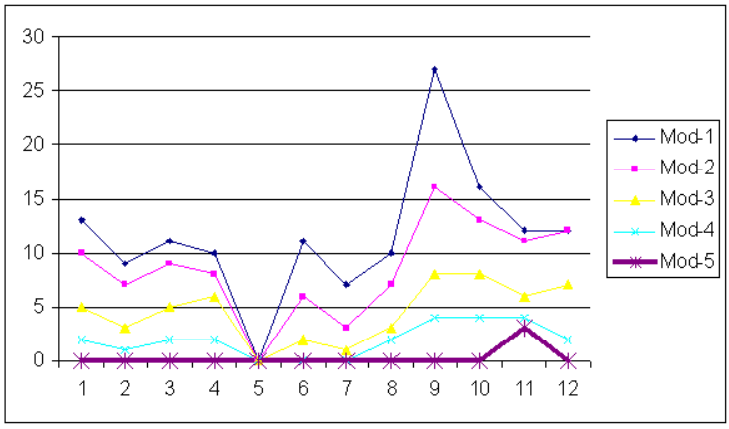



**Figure 6.** Safety fails.

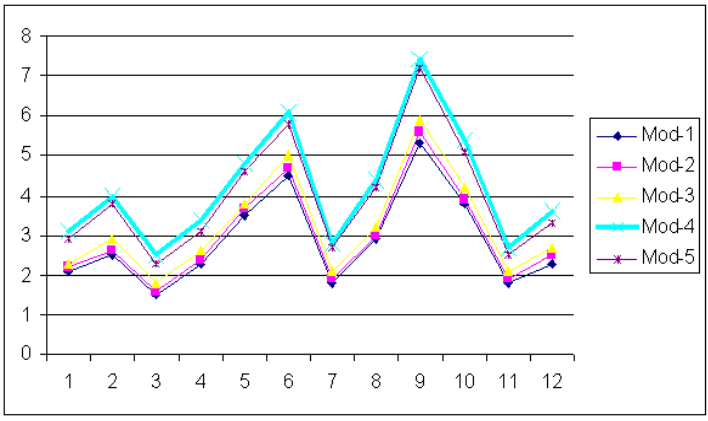



**Figure 7.** Productivity.

The number of nodes created by the minimax algorithm for a search depth of 1–10 and corresponding search time are shown in Table 3.

**Table 3.** Number of nodes created my the minimax algorithm

| depth of search/$H_L$ | search time (ms)/T | created search nodes, V |
|---|---|---|
| 1 | 120 | 1812 |
| 2 | 270 | 6531 |
| 3 | 345 | 18472 |
| 4 | 571 | 51872 |
| 5 | 1100 | 112534 |
| 6 | 1890 | 167550 |
| 7 | 3562 | 231124 |
| 8 | 6749 | 293453 |
| 9 | 10127 | 368980 |
| 10 | 15120 | 471652 |

### *6.2. System Administrators Performance Compared with the GBDP System for the Depth 4*

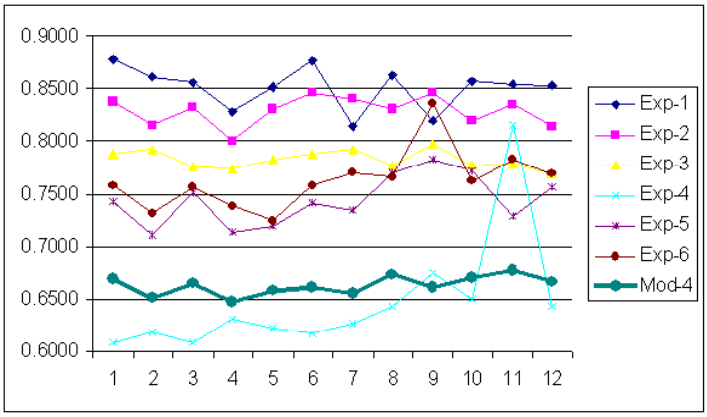



**Figure 8.** Distance to safety.

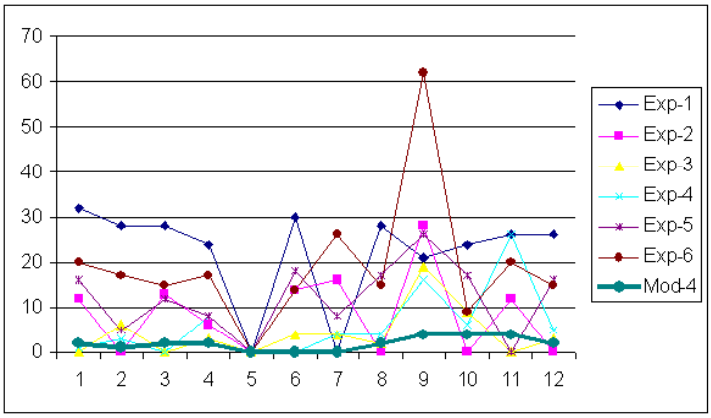



**Figure 9.** Safety fails.

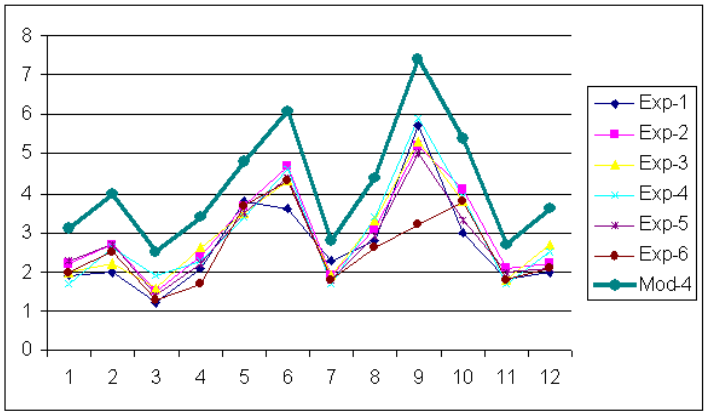



**Figure 10.** Productivity.

### *6.3. Protection Programs Performance Compared with the GBDP System for the Depth 4*

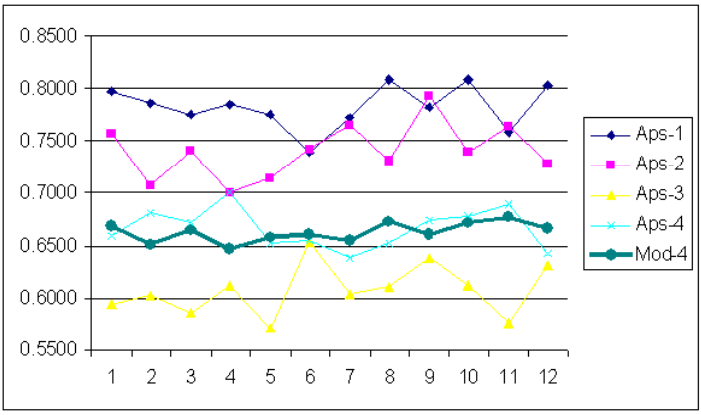



**Figure 11.** Distance to safety.

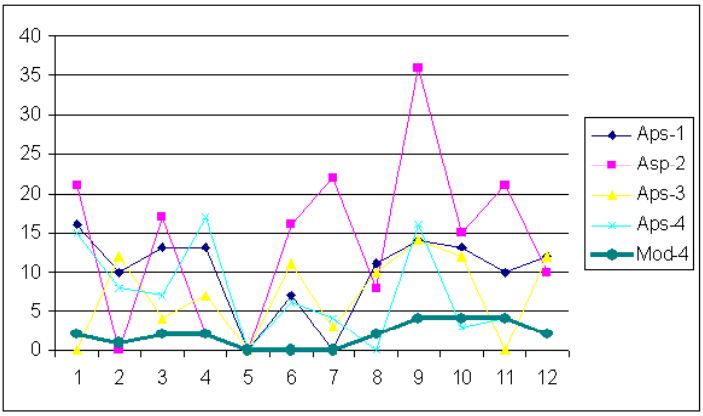



**Figure 12.** Safety fails.

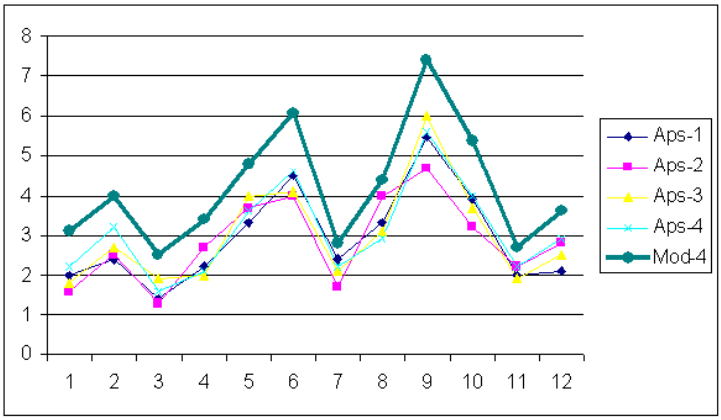



**Figure 13.** Productivity.

Sampling means for Distance to Safety, Safety Fails and Productivity for the sample in 12 attacks, variances and standard deviations for the Distance to Safety (Table 4).

**Table 4.** Sampling means for distance to safety, safety fails and productivity

| | DtS means | DtS variances | DtS stnd. dev. (σ) | safety fails | safety fails % | prd. |
|---|---|---|---|---|---|---|
| exp-1-6 | 0.7692 | 0.0152 | 0.1205 | 12.00 | 6.5% | 2.82 |
| aps-1-4 | 0.6993 | 0.0205 | 0.1409 | 9.46 | 5.1% | 2.96 |
| M1 | 0.7785 | 0.0170 | 0.1301 | 11.50 | 6.2% | 2.86 |
| M2 | 0.7915 | 0.0169 | 0.1295 | 8.50 | 4.5% | 3.00 |
| M3 | 0.6692 | 0.0206 | 0.1423 | 4.50 | 2.4% | 3.22 |
| M4 | 0.6629 | 0.0187 | 0.1357 | 1.92 | 1.0% | 4.18 |
| M5 | 0.6748 | 0.0177 | 0.1327 | 0.25 | 0.1% | 3.96 |

Improved sampling means for the GBDP systems compared with the experts (Table 5).

**Table 5.** Comparison of sampling means with experts

| decreasing the safety of fails in % | | | | | |
|---|---|---|---|---|---|
| | mod-1 | mod-2 | Mod-3 | mod-4 | mod-5 |
| exp-1-6 | 4.2% | 29.2% | 62.5% | 84.0% | 97.9% |
| aps-1-4 | –21.6% | 10.1% | 52.4% | 79.7% | 97.4% |
| decreasing the distance to safety in % | | | | | |
| | mod-1 | mod-2 | Mod-3 | mod-4 | mod-5 |
| exp-1-6 | –1.2% | –2.9% | 13.0% | 13.8% | 12.3% |
| aps-1-4 | –11.3% | –13.2% | 4.3% | 5.2% | 3.5% |
| increasing the productivity in % | | | | | |
| | mod-1 | mod-2 | Mod-3 | mod-4 | mod-5 |
| exp-1-6 | 1.4% | 6.0% | 12.3% | 32.6% | 28.8% |
| aps-1-4 | –3.7% | 1.2% | 7.8% | 29.1% | 25.1% |

Analyzing the experiments – minimax vs. experts, the following conclusions can be made:

1. the experts, on average, search at a depth of $\mathbf{H_L = 1–3}$. During the functioning of the GMDP under the same depth the system state and performance are the same as the experts;
2. known autonomous software systems search at a depth not over $\mathbf{H_L = 1,2}$. During the work of the GBDP under the same depth the system state and performance are either the same as the experts or greater by **0–15%**;
3. the GBDP under the minimal depth $\mathbf{H_L = 1,2}$ and under other similar conditions, chose the best protection strategy against intrusion better than the experts and autonomous software systems not less than in **10%** of the experiments;
4. by increasing the depth to $\mathbf{H_L = 3}$ and the number of false attacks, the model was able to avoid them in **62.5%** of the cases, and, as a result, prevented the system from going into critical states and maximally loaded it to execute "normal" actions;
5. depending on the controlled resources and the types of attacks, the system performance during the work of the GBDP was, on average, greater by **31.3%** than during the work of the experts and autonomous software systems;
6. under the depth $\mathbf{H_L = 5–10}$ the work of the system slows down because the time wasted on the construction of each subtree increases from **700 ms** to **15s**. This fact negatively affects the process of quickened decision making in real time systems.

Further increasing the efficiency of the system must evidently be realized by cutting-down (pruning) the search tree.

## 7. Intermediate Goals at First Algorithm for Searching Counteracting Strategy

We have successfully used and experimented with the minimax algorithm to counteract intrusions and have developed a new algorithm based on the ideas of Botvinnik's "Intermediate Goals At First"(IGAF) algorithm.

Suggested by Botvinnik, the cutting-down tree [11] algorithm for chess is based on the initial extraction of subgoals within a game tree, which allows to sharply reduce the searching tree as compared to the minimax method. The method allows the determination of the moving trajectories of confronting parties in order to construct a zone around the extracted subgoal trajectory.

Let us describe our implementation of the IGAF algorithm.

### *7.1. Trajectory of an Attack*

The trajectory of an attack is a subtree $\mathbf{G^a(S', P')}$, where **S'** is a subset of the system states $\mathbf{S' \subseteq S}$ and **P'** is a subset of the actions, consisting of an offensive's conversion procedures $\mathbf{P^a}$ and a defender's "normal" conversion procedures $\mathbf{P^{dH}}$, i.e. $\mathbf{P'=P^a \cup P^{d\acute{I}}}$, $\mathbf{P' \subseteq P}$, $\mathbf{P' \neq \varnothing}$.

### 7.2. Zone of Counteraction

The zone of counteraction is a subtree $\mathbf{G^z(S'', P'')}$ built around the graph of the trajectory of an attack $\mathbf{G^a(S', P')}$, i.e. $\mathbf{G^a \subseteq G^z}$, where **S"** is a subset of the system states $\mathbf{S'' \subseteq S}$, which belong to the trajectory of an attack, hence $\mathbf{S' \subseteq S''}$, and **P"** is a subset of the actions, which consist of the conversion procedures, defined on the trajectory of an attack, **P'** and the defender's conversion procedures $\mathbf{P^{dO}}$, i.e. $\mathbf{P''= P' \cup P^{dO}}$, $\mathbf{P'' \subseteq P}$, $\mathbf{P'' \neq \varnothing}$, hence $\mathbf{P' \subseteq P''}$.

In the given model **M,** the aim of the algorithm to find the best strategy search amounts to solving an approximate searching problem, which is solved by using a cutting-down tree procedure within a searching tree (Fig. 14).

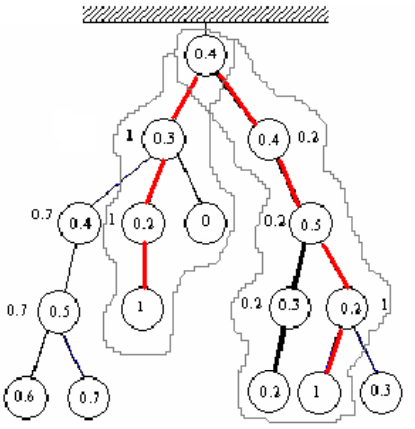



**Figure 14.** The cutting-down tree algorithm.

Using the fact that the game model **M** is represented by an "AND/OR" tree, we will prove that the player **[D]** can choose the best action strategy from the initial state $\mathbf{s_0 \in S}$. Then "OR" and "AND" vertices represent the states which appear after the moves of the players **[D]** and **[A]** correspondingly. Let us consider that **[D]** moves first from the initial state $\mathbf{s_0 \in S}$ and the players move in turns. Consequently, "OR" vertices will be children vertices of "AND" vertices and vice versa. According to the suggestion made, we will consider that the initial vertex is an "AND" vertex. The terminal vertex corresponds to any state, which is known to be the winning one for player **[D]**.

Let us consider that at the moment **t** a subset of **R**, i.e. the resources $\mathbf{\{r1,\ldots, rk\} \subseteq R}$, is changing. Let **bADStep** be an element, the value of which is *true* if **[A]** moves, and *false* if **[D]** moves. $\mathbf{H_L}$ is the considered depth of a subtree during each finite step of a player from the starting vertex **i** (the root of the subtree) to the terminal vertices. $\mathbf{f_1}$ and $\mathbf{f_2}$ are criterion minimax functions for the players **[A]** and **[D]** correspondingly, where

$$\mathbf{f_1(s_1, \ldots, s_k) = max(s_1, \ldots, s_k)\ ,\ s_1, \ldots, s_k \in S}$$
$$\mathbf{f_2(s_1, \ldots, s_k) = min(s_1, \ldots, s_k)\ ,\ s_1, \ldots, s_k \in S}$$

Let the values of the criterion functions $\mathbf{f_1}$ and $\mathbf{f_2}$ denote with $\mathbf{k^i_1}$ and $\mathbf{k^i_2}$ coefficients correspondingly, where $\mathbf{k^i_1, k^i_2 \in Z}$.

*The initial values:*

**bADStep :=** ***false;*** $\mathbf{H_L := 1, 2, 3,\ldots; g = 0;}$

*Step 1:* Determine the current state $\mathbf{s_i \in S}$. If $\mathbf{s_i = 0}$ then it is necessary to stop looking down within the tree and to mark the vertex as closed. If $\mathbf{s_i = 1}$, then it is necessary to extract the moving trajectory and to mark the vertex as closed. Otherwise, move to the next step.

*Step 2:* If the current depth $\mathbf{g = H_L}$, then move up to the closest vertex, i.e. $\mathbf{g := g - 1}$ and move to *Step 5*. If $\mathbf{g = 0}$, i.e. the first parent vertex has been reached, then move to *Step 6*. Otherwise, move to the next step.

*Step 3:* Determine all the resources $\mathbf{R_i = \{r_1^i, \ldots, r_k^i\} \subseteq R}$, changing of which are the cause of moving from the previous state $\mathbf{s_j \in S'}$ to the current state $\mathbf{s_i \in S'}$. Move to the next step.

*Step 4:* Increase the parameter of the depth $\mathbf{g := g + 1}$. If **bADStep** = *false,* then pick out a subset of the defender's conversion procedures $\mathbf{P^d_i = \{ p^d_1, \ldots, p^d_k \} \subseteq P'}$, which can change the subset $\mathbf{R_i \subseteq R}$. **bADStep :=** *true*. If **bADStep** = *true,* then pick out a subset of the offensive's conversion procedures $\mathbf{P^a_i = \{ p^a_1, \ldots, p^a_k \} \subseteq P'}$, which can change the subset $\mathbf{R_i \subseteq R}$. **bADStep :=** *false*. Move to the next step.

*Step 5:* Determine the next (or the first) procedure $\mathbf{P_i}$ where $\mathbf{P_i = P^{dH}_i}$ or $\mathbf{P_i = P^a_i}$ and run it. Return to *Step 1*. If $\mathbf{g > 0}$ and $\mathbf{P_i > P_k}$, i.e. all the conversions have been run, then move up to the closest vertex, i.e. $\mathbf{g := g - 1}$ and move to *Step 5*. If $\mathbf{g = 0}$, i.e. the first parent vertex has been reached and moving up is impossible, then cut off all the branches which are not in $\mathbf{G^a}$.

*Initialize:*

**bADStep :=** ***false;*** $\mathbf{H_L := 1, 2, 3,\ldots; g = 0; s_i := s_0;}$

*Step 6:* Determine the current state $\mathbf{s_i \in S''}$. If $\mathbf{s_i = 0}$ or $\mathbf{s_i = 1}$, then it is necessary to stop looking down within the tree and to mark the vertex as closed. Otherwise, move to the next step.

*Step 7:* If the current depth $\mathbf{g = H_L}$, then move up to the closest vertex, i.e. $\mathbf{g := g - 1}$ and move to *Step 10*. If $\mathbf{g = 0}$, i.e. the first parent vertex has been reached, then move to *Step 11*. Otherwise, move to the next step.

*Step 8:* Determine all the resources $\mathbf{R_i = \{r_1^i, \ldots, r_k^i\} \subseteq R}$, changing of which are the cause of moving from the previous state $\mathbf{s_j \in S''}$ to the current state $\mathbf{s_i \in S''}$. Move to the next step.

*Step 9:* Increase the parameter of the depth $\mathbf{g := g + 1}$. If **bADStep** = *false,* then pick out a subset of the defender's conversion procedures $\mathbf{P^d_i = \{ p^d_1, \ldots, p^d_k \} \subseteq P''}$, which can change the subset $\mathbf{R_i \subseteq R}$. **bADStep :=** *true*. If **bADStep** = *true,* then pick out a subset of the offensive's conversion procedures $\mathbf{P^a_i = \{ p^a_1, \ldots, p^a_k \} \subseteq P''}$, which can change the subset $\mathbf{R_i \subseteq R}$. **bADStep :=** *false*. Move to the next step.

*Step 10:* Determine the next (or the first) procedure $\mathbf{P_i}$ and run it. Return to *Step 6*. If $\mathbf{g > 0}$ and $\mathbf{P_i > P_k}$, i.e. all the conversions have been run, then move up to the closest vertex, i.e. $\mathbf{g := g - 1}$ and move to *Step 10*. If $\mathbf{g = 0}$, i.e. the first parent vertex has been reached and moving up is impossible, then move to the next step.

*Step 11:* Coefficients assignment.

During recursive move from the low terminal vertices upward, i.e. from the children's to the parents', $\mathbf{k^i_1, k^i_2 \in Z}$ coefficients are being assigned to each parent vertex, where $\mathbf{k^i_1 = f_1}$, if **bADStep** = *true*, $\mathbf{k^i_2 = f_2}$, if **bADStep** = *false,* This process continues to the starting vertex.

*Step 12:* Choosing the best move.

The best conversion procedure for player **[D]** is chosen by the action whose $\mathbf{k}^{i}_{2}$ coefficient has the minimal minimax value. In other words, the move which brings the system to the more stable state is chosen.

*Step 13:* The end of the move. Wait for the opponent's actions.

## 8. Intermediate Goals at First vs. Minimax: The Results of Experiments

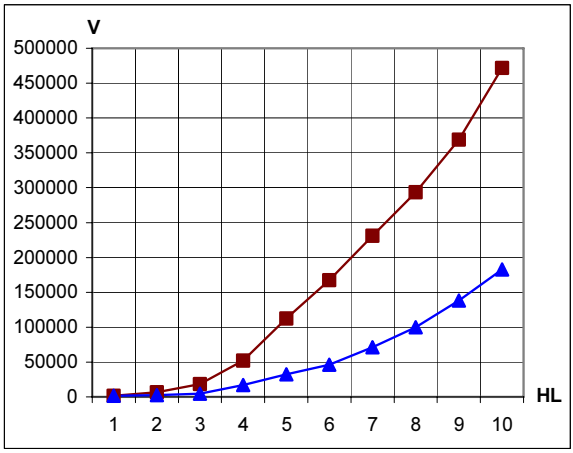



**Figure 15.** Reduced number of nodes searched by the IGAF algorithm compared with the minimax algorithm.

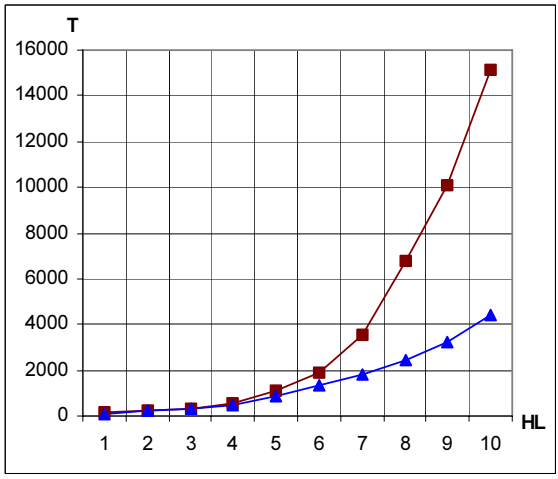



**Figure 16.** Reduced searching time by the IGAF algorithm compared with the minimax algorithm.

Upon analyzing the experiments the following conclusions can be drawn:

1. under the depth $\mathbf{H_L = 1–10}$ the system state and performance for the minimax and IGF algorithms were the same (0.1–1.2%). This means that the IGF is

able to preserve a quality of protection comparable with experts and exhaustive minimax search;
2. under the depth $\mathbf{H_L = 1–3}$ and other similar conditions, time wasted on the construction of the tree and the number of constructed vertices were the same during the computation of both algorithms;
3. by increasing the depth to $\mathbf{H_L = 4–10}$ the cutting-down tree algorithm compared to the minimax method, on average, has achieved the following results:
   a) The construction of the tree was faster by 1.2–4.7 times;
   b) The number of constructed vertices was reduced by 1.3–3.8 times.

## 9. Summary

The experiments were aimed at proving the following hypothesis:

- the model using the minimax algorithm is comparable to experts (system administrators or specialized programs) against intrusions or other forms of perturbations of the base system;
- the IGAF cutting-down tree algorithm, along with being comparable to the minimax algorithm, can work efficiently enough to be used in practice.

Thus, the empirical data has provided enough evidence of the fact that our game model, using the minimax or the IGAF algorithms, are comparable to the average expert in fighting against suspicious changes in the functionality of the base system, caused by some perturbations, particularly by attacks.

As the results of experiments the following statements can be formulated.

1. the game model at depth 4 prevails over each protection system against any of 12 attacks by all criteria. Only one expert by only one criterion – the DtS, was comparable with the model.
2. the game models improve their performances in all criteria by deepening the search up to depth 5.
3. at a depth higher than 4 the minimax based search exceeds the time limits allowed for making protection decisions in the base system. New approaches are required that preserve the advantages of increasing the depth of search but overcome the exponential increase in search time.
4. GMDP systems based on the Intermediate Goals At First search algorithms allow the consecutive increase of the quality of protection up to a depth of 9 with search times not exceeding the time limits of decision making in the base system.
5. preliminary experiments with the GMDP system based on the Intermediate Goals At First search algorithms demonstrate that up to a depth of 9 they maintain the quality of protection equal to that of minimax with search times not exceeding the time limits for decision making in the base system.

## References

[1] E. Pogossian. Adaptation of Combinatorial Algorithms.(in Russian), Yerevan., 1983, 293 pp.
[2] D.E. Denning, An Intrusion Detection Mode, *IEEE Trans. Software Eng., vol. 13, no. 2,* Feb. 1987.

[3] C. Ko, M. Ruschitzka, K. Levitt, Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-Based Approach, *Proc. 1997 IEEE Symp. Security and Privacy, IEEE CS Press,* Los Alamitos, Calif., 1997.
[4] A.K. Ghosh, J. Wanken, F. Charron, Detecting Anomalous and Unknown Intrusions Against Programs, *Proc. Annual Computer Security Application Conference (ACSAC'98), IEEE CS Press,* Los Alamitos, Calif., 1998.
[5] K. Ilgun, R.A. Kemmerer, P.A. Porras, State Transition Analysis: A Rule-Based Intrusion Detection System, *IEEE Trans. Software Eng. vol. 21, no. 3,* Mar. 1995.
[6] V. Paxson, Bro: A System for Detecting Network Intruders in Real-Time, *Proc. Seventh Usenix Security Symp., Usenix Assoc.*, Berkeley, Calif., 1998.
[7] U. Lindqvist, P.A. Porras, Detecting Computer and Network Misuse with the Production-Based Expert System Toolset, *IEEE Symp. Security and Privacy, IEEE CS Press,* Los Alamitos, Calif., 1999.
[8] N.G. Miloslavskaya, A.I. Tolstoj, *Intranetwork: Intrusion Detection,* Unity, Moscow, 2001 (in Russian).
[9] A. Lukacki, *Attack Detection,* BXB-Petersburg, St. Petersburg, 2001 (in Russian).
[10] E. Pogossian Focusing Management Strategy Provision Simulation. Proceedings of the CSIT2001, 3d International Conference in Computer Science and Information Technologies, Yerevan, 2001.
[11] M.M. Botvinnik, *About solving approximate problems*, Soviet Radio, Moscow, 1979 (in Russian).

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# The Use of G-Networks Models for the Assessment of Computer Networks Performance

Khanik KEROBYAN[a], Khatchik VARDANYAN[b] and N. KOSTANYAN[c]
[a] *Institute of Information and Automation Problems National Academy of Sciences, Republic of Armenia, Russian-Armenian (Slavon) State University, Armenian Association of Computer and Information Systems,* kerobyan@yandex.ru
[b] *Yerevan State University*
[c] *State Engineering University of Armenia*

**Abstract.** The development of IP technologies, as well as high-speed nets for data translation and the commercialization of the global net, has led in the last 10 years to the creation of many net services and systems geared toward improving the functionality of active services. A number of new services have appeared that in the past were accessible only to users of corporate local nets. Among such services based on OLAP technologies are systems of remote database access, data extraction and support of analytical decision making. OLAP makes it possible to aggregate and save information in large databases in order to search and utilize, with the help of standard requests, this same information for analytical decision making and creation of reports. Given the strong competition in the IT market, the advance (promotion) of new services and technologies depends on their functionality, quality of service (QoS) and effectivity of design process. QoS and effectivity of service design in turn depend on the specificity and parameters of the service environment. The following factors influence the effectivity of data fusion systems and decision making support systems: different types of attacks and viruses lead not only to the distortion and destruction of information saved in the system, but also to the overloading of service resources and to a disruption of normal functioning. In this article, the methods and models of the estimation of possible consequences following the influence of different types of attacks and viruses on the effectivity of the system and net have been considered. We propose to use queue models with the catastrophes and models of multipolar G-Networks with list oriented deletions (functioning in a random environment) in order to predict the impact of various attacks and intrusions on the productivity of computer networks. For queue model Mx/G/1 with message group arrival & catastrophes, the Laplace-Stieltjes transformation is determined for the distribution function of the system's busy period, queue length, queue capacity, expectation time, time before the first catastrophe & their moments. In the supposition that parameters of the model depend on the state vector of a network and that they can be represented in Henderson-Tailor form, the availability of the product form solution for stationary distribution of the network has been demonstrated, and conditions for network stationarity and stability have been formulated.

**Keywords.** Data fusion systems queuing model, disaster, G-networks, product form, network performance, attacks, intrusions, viruses

## 1. Introduction

Along with the expansion of fields utilizing contemporary information technologies (i.e. computer networks and the Internet), the protection of network resources – particularly, the protection of the information stored therein from unauthorized access by attacks and viruses – gains more urgency. Presently, rigorous studies involving the creation of systems, facilities and methods of exposure, localization, identification, counteraction and liquidation of intrusions, attacks and viruses are underway. A wide range of facilities is being developed – from data signature analysis programs to multi-agent systems based on artificial intelligence methods.

The difficulty in making a choice among protection systems in contemporary computer networks is conditioned by several factors as follows:

Firstly, the methods and forms of network intrusions and attacks are constantly improving and may pursue various aims, such as copying, moving, transforming or deleting the information stored in the network. It frequently happens that by capturing network resources, particular network resources and networks on the whole are partially or completely blocked; the network dataflow is thus changed, resulting in drastically reduced network productivity.

Secondly, the security facilities utilized in networks also consume a considerable amount of network resources, thus resulting in a decrease in network productivity on the whole.

In connection with the abovementioned, it is believed that the choice of a particular protection system must be based on both the assessment of potential damage caused by possible intrusions and attacks, as well as on the amount of money spent on a given protection system. It should be noted that the discrepancy between the cost of a protection system and the potential damage caused by the intrusion is quite often the determining criterion which compels many users to give up protection systems which although efficient, are costly enough and require enormous network resources.

While studying the consequences of various attacks, viruses and intrusions on network productivity the following important considerations must be taken into account.

First of all, it is necessary to imitate most precisely the uniqueness and peculiarities of different attacks, viruses and protection systems; for example, the peculiarities of viruses in reproduction and migration through a network's nodes, or the peculiarities of protection systems in the destruction of viruses and infections in the nodes.

Secondly, the research on network performance allows to expose the most dangerous types of attacks, viruses and intrusions and also to substantiate the choice of certain types of protection systems.

Thirdly, based on the anomalous modification of certain integral network specifications (for example, network productivity or the loading of its nodes) such research makes it possible to expose the existence of certain types of attacks, intrusions or viruses in the network.

As the results of a number of research efforts reveal, the variation of a network's productivity under the impact of certain types of viruses is accompanied by modifications in a number of important network specifications, the domain of stability thereof and certain network resource loads [1]. For example, viruses which delete the information stored in the network lead to a decrease of network productivity and at the same time, to the "expansion" of the network domain of stability [2]; as for viruses causing the information in the network to be moved, they lead to the change of certain network node loads [3]. Nevertheless, as mentioned in [4,5] the said "characteristics" of viruses

may be resourcefully used as effective facilities for the management of network resources and processes through the synthesis of their own network facilities. Such facilities may imitate certain functions of viruses to resolve many currently actual and complex problems of network theories; for example, for the dynamic balancing of different network node loads, correction of data stored in allocated databases, synthesis of new, more efficient data processing and transmitting methods, etc. In particular, the characteristics of certain viruses capable of selectively deleting information, may be used for the development of facilities (they may also be called agents) the functions of which include search, detection, correction and/or renewal of allocated databases. As for the characteristics of viruses that move data, they may be used for balancing the load of various network nodes and resources. The works of [6] must also be noted, where the properties of viruses capable of transforming data are uniquely used for the purpose of developing effective mechanisms for the arrangement of cashing and multi-address transmission of multi-media files.

The models of G-Networks recommended in [2] may be used to predict the consequences of intrusions and attacks on network productivity as well as to estimate the share of network resources consumed by the protection system. The specificity of the mentioned models is as follows: along with the calculation of regular messages, the calculation of the flow of "negative" message-signals through which various types of intrusions, attacks and viruses may be patterned, both the synchronism of a number of processes running in the network (for example, simultaneous entry, service or removal of messages in various network nodes) as well as the diversity of mechanisms for the service of messages in network nodes may be considered. Finally, an important peculiarity of these networks is the multiplicativity of network stationary distribution which makes it possible to use the whole range of methods, algorithms and programs, developed for the analysis and assessment of Markovian networks, in the research of G-Networks specifications.

Let us note that the works [7–16] are devoted to research problems on models of network nodes with consideration to a catastrophe's (disaster) impact on streams. In particular, in [11,12] with the help of matrix analytical methods, the models of type G/G/1, M/G/1, G/M/1 and M/M/1 with catastrophes have been investigated. The models of types G/G/1, M/G/1, G/M/1 and M/M/1 with different mechanisms of message deletion have been considered in [8,9,12–14].

The exhaustive overview of works devoted to G-Networks research as of the year of 2000 is presented in [1]. Within the context of the recommended work, [6] and [15] should also be noted, where nonuniform G-Networks with list oriented "deletions" are studied; these are currently the most general models of G-Networks which take into consideration practically the whole range of functions performed by signals. The issues of multiplicativity and stability of the mentioned models, together with the intensities, depending on the network state, of entering messages and signals, as well as the service parameters of messages, which allow the use of the formalism of Henderson-Tailor, are studied in [5].

The G-Networks studied in the present work summarize the results of [4,16–18] in the case when network parameters depend on the state of the "environment." Various time behavior and loadable characteristics of the network are determined. The theorem on the multiplicativity of network stationary distribution is formulated, and its stability conditions are defined.

It should be noted that network models functioning in a random (Markovian) environment most accurately describe the operation of a LAN, a WAN, and of networks

connected to the global network. As the numerous studies on measuring network traffic demonstrate, such as in [19,20], Internet and Ethernet traffic is characterized by its modulation and burstness. The models belonging to the category in question are also studied with various types of signals, in the works of [1–5,11]. Particularly, in [4] these models are used to explore ATM networks with the incoming inspection of the requests flow coming from outside; as for [6], these models are used to analyze the productivity and the choice of optimal parameters for the requests service by the deleted clients of the multimedia system. From among works that study systems functioning in a random environment, special attention should be given to the wonderful book [18] in which general stochastic construction, such as random evolutions, are studied on the basis of phase enlargement methods.

In the present work, the supposition is that signals circulating in the network may perform the following functions: to move regular messages from one network node to the other, as per a specific criterion to delete or remove regular messages or groups from the whole network or from a certain set of nodes thereof, to self-reproduce and expand throughout the network, as per a specific criterion to transform certain types of regular messages in the network.

## 2. Models of Network Nodes

In this section, the models of single-channel systems such as M/G/1 with an entering Poisson stream and disasters are considered. It has been considered that disasters do not form queues and have not been served. They destroy all regular messages taking place in the system at the time of their arrival. If at the time of arrival of a disaster the system is empty, they are ignored. Models of system $M^x/G/1$ with group (bulk) arrival have been investigated. For the specified models, the busy period of the system, the distribution of time the system is functioning before the first disaster, the distribution of queue length of regular messages in the system have been investigated.

We shall enter the following notations:

$\alpha$: exponentially distributed random variable (r.v.) with intensity $\lambda$ (an interval of time between two moments causing the arrival of message groups): $g$, $g_i$, $G(z)$ discrete r.v., its distribution ( $g_i$, $i=1,2,..$ ) and its probability generating function (PGF)

$$G(z)=\sum_{i\geq 1} z^i g_i,\quad G'(1)=\bar{g}=\sum_{i\geq 1} i g_i,\quad G(0)=\sum_{i\geq 1} g_i=1,$$

where $\bar{g}$ is the average value (size of group), $g_i$ is probability of that in message group $i, i=1,2,..$.

V: exponentially distributed r.v. with intensity $\nu$ (an interval of time between two moments of disaster arrivals):

$s, S(t), \tilde{S}(\theta)$ are r.v. of service time, its distribution function (SDF) and transformation of Laplace-Stieltjes (LST)

$$\tilde{S}(\theta)=\int_0^\infty e^{-\theta t}dS(t),\ \tilde{S}'(0)=\overline{s}=\int_0^\infty tdS(t)$$

where $\overline{s}$ is the average value of message service time.

$\pi,\overline{\pi},\pi(t),\tilde{\pi}(\theta)$ are the r.v., average value, SDF and LST of a standard model's busy period, and $\pi_v,\overline{\pi}_v,\pi_v(t),\tilde{\pi}_v(\theta)$ are the same notations for models with disasters:

$$\tilde{\pi}'(0)=\overline{\pi}=\int_0^\infty td\pi(t),\ \tilde{\pi}_v'(0)=\overline{\pi}_v=\int_0^\infty td\pi_v(t)$$

where $\overline{\pi},\overline{\pi}_v$ are the average values of the busy period for the standard model and for the model with disasters.

First, let us consider the model $M^x/G/1$ with disasters.

*2.1. Description of the System*

Let us consider a service system in which messages have arrived in groups. The group arrival times have formed a Poisson stream with parameter $\lambda$. With probability $g_i$ the quantity of messages in a group is equal to i. The message which arrives to an idle system is immediately served. If at the moment of message arrival the system is busy, the message is appended to the queue. The messages in the system are served according to the FIFO principle. The duration of each message service is r.v. with SDF $S(t)$. The duration of service of various messages are independent r.v.. It is supposed that at the initial moment $t=0$, the system is free from messages.

*2.2. Busy Period of a System*

The time interval called a busy system period, starts from the moment of the message arrival in the free system and ends at the time of the first release of the system (in consequence of an end of message service or a disaster arrival). Let us denote as $\pi_v(t)$ the SDF of a busy system period.

**The theorem.** LST and average value $\overline{\pi}_v$ of the busy period of the considered system $\tilde{\pi}_v(\theta)$ are set by the equations:

$$\tilde{\pi}_v(\theta)=\frac{v+\theta\tilde{\pi}(\theta)}{v+\theta},\quad \overline{\pi}_v=\frac{1-\tilde{\pi}(v)}{v}$$

where $\tilde{\pi}(\theta)$ is the LST of the busy period of standard $M^x/G/1$ systems without disasters, which is the unique solution of the functional equation

$$\tilde{\pi}(\theta) = \tilde{S}(\theta + \lambda - \lambda G(\tilde{\pi}(\theta))).$$

To prove this theorem, we can use a method of supplementary events [21,22]. Let event **A** arrive independently of the system's functioning. The stream of event **A** is exponential, with $\theta$ parameter. Then, $\tilde{\pi}_v(\theta)$ is equal to the probability of the absence of event **A** during the busy period of the system. For this purpose it is necessary and sufficient, that ***either*** the busy period has ended in consequence of an end of message service and that during the busy period neither disasters nor event **A** (probability $\tilde{\pi}(\theta+\nu)$) have arrived, ***or*** the busy period has ended in consequence of event **A**'s arrival

$$\text{(probability } \frac{\nu}{\nu+\theta}(1-\tilde{\pi}(\theta+\nu))$$

From here, for $\tilde{\pi}_v(\theta)$ it is obtained that

$$\tilde{\pi}_\nu(\theta) = \tilde{\pi}(\theta+\nu) + \frac{\nu}{\nu+\theta}(1-\tilde{\pi}(\theta+\nu)),$$

$$\bar{\pi}_\nu = -\tilde{\pi}'_\nu(\theta)\Big|_{\theta=0} = \frac{1-\tilde{\pi}(\nu)}{\nu}$$

*2.3. Period of Regeneration*

It is obvious that the moments of the system's transition into the idle condition are the moments of regeneration describing the functioning of a system of casual process. We understand, as regeneration period, a time interval between two consecutive transitions of a system in an idle condition. Let us note through the $\tau(t)$ SDF the period of regeneration of a system. Then, for the LST-duration of one period of regeneration $\tilde{\tau}(\theta)$ we shall obtain

$$\tilde{\tau}_1(\theta) = \frac{\lambda}{\lambda+\theta},\ \tilde{\tau}_i(\theta) = \tilde{\tau}(\theta) = \frac{\lambda}{\lambda+\theta}\tilde{\pi}_\nu(\theta),\ i \geq 2$$

From the abovementioned, the consecutive periods of regeneration form the renewal process. In accordance with [21,22], for the LST density of renewal of such a process we have

$$\tilde{h}(\theta) = \frac{\tilde{\tau}_1(\theta)}{1-\tilde{\tau}(\theta)} = \frac{\lambda}{\theta + \lambda(1-\tilde{\pi}_v(\theta))}$$

If, through $\tilde{P}_0(\theta)$, $\tilde{P}_b(\theta)$ we notice respectively the LST probabilities that in time $t$ the system is idle and the system is busy with message service, then for them we shall obtain

$$\tilde{P}_0(\theta) = \frac{1}{\lambda+\theta} + \frac{\tilde{\pi}_V(\theta)}{\lambda+\theta}\tilde{h}(\theta) = \frac{1}{\theta+\lambda-\lambda\tilde{\pi}_V(\theta)},$$
$$\tilde{P}_b(\theta) = 1-\tilde{P}_0(\theta),$$

whence, for stationary probabilities $P_0, P_b$ is obtained

$$P_0 = \lim_{t\to\infty} P_0(t) = \lim_{\theta\to 0}\theta\tilde{P}_0(\theta).$$
$$P_0 = \frac{1}{1+\lambda E[\min\{\pi,V\}]}, \quad P_b = \frac{\lambda E[\min\{\pi,V\}]}{1+\lambda E[\min\{\pi,V\}]}$$

Here $E[\min\{\pi,V\}]$ is the average of the minimum of two random variables $\pi$ and $V$.

If we are interested only in stationary probabilities of the presence of a system in idle or busy condition, in that case we should take into account that the average value of one regeneration period is equal to

$$\overline{\tau} = \frac{1}{\lambda} + \frac{1-\tilde{\pi}(v)}{v}$$

and the average sojourn time of a system in idle condition is equal to $\frac{1}{\lambda}$, for $P_0$ we shall obtain

$$P_0 = \frac{\frac{1}{\lambda}}{\frac{1}{\lambda}+\frac{1-\tilde{\pi}(v)}{v}} = \frac{v}{v+\lambda(1-\tilde{\pi}(v))} = \frac{1}{1+\rho(1-\tilde{\pi}(v))}.$$

where the coefficient $\rho$ is equal to $\rho = \lambda / v$.

*2.4. SDF of Time of System Functioning Before the First Disaster*

Let us consider a system functioning until the first moment of disaster arrival. For researching the DF of a random variable $\zeta$ we shall use the phase enlarging methods of

Semi-Markov Processes (SMP) [21,22]. Let $\xi(t)$ be the SMP describing the functioning of a system in one period of regeneration before a disaster. We shall enter the states SMP $\xi(t)$ E={$e_0$, $e_1$, $e_2$} where $e_0$ – the system is free from messages, $e_1$ – the system is busy with message service and $e_2$ – the disaster has arrived during message service time (absorbing state). Let us note by $\alpha_i$ the sojourn time of a SMP in $e_i$ state. Whereas the functioning of the system is considered until the moment it hits an absorbing state, the sojourn time in $e_2$ will not interest us.

Let, $P_i$(t), $Q_{ij}$(t), $p_{ij}$ be respectively the DF of the sojourn time of an SMP in an $e_i$ state, the transitive probabilities of an SMP and its embedded Markov chain, which are defined by the following equations:

$$Q_{01}(t)=1-e^{-\lambda t},\ Q_{10}(t)=\int_0^t e^{-vt}d\pi(t),\ Q_{12}(t)=\int_0^t (1-\pi(t))d(1-e^{-vt}),$$

$$p_{ij}=\lim_{t\to\infty}Q_{ij}(t)$$

$$p_{01}=1,\quad p_{10}=\tilde{\pi}(v),\quad p_{12}=1-\tilde{\pi}(v)$$

It is obvious that the DF and average values of the sojourn time of an SMP in states are equal to:

$$P_0(t)=1-e^{-\lambda t},\qquad P_1(t)=1-e^{-vt}(1-\pi(t))$$

$$\overline{\varphi}_0=\frac{1}{\lambda},\quad \overline{\varphi}_1=\frac{1}{v}(1-\tilde{\pi}(v))$$

where $\pi(t),\tilde{\pi}(\theta)$ are respectively the DF and LST of the busy period of a standard $M^x$/G/1 system. The time of functioning of a system before disaster is equivalent to sojourn time $\zeta$ of SMP in a subset of states $\{e_0, e_1\}$ before first hitting state $e_2$. Let us note by $\omega_i(t), \tilde{\omega}_i(\theta), \overline{\omega}_i$ respectively the DF, LST and average value $\zeta$ when the initial condition is $e_i$, $e_i \in E_0=\{e_0,e_1\}$.

According to [21], $\tilde{\omega}_i(\theta)$, $\overline{\omega}_i$ are defined by the following equations

$$\tilde{\omega}_0(\theta)=\tilde{Q}_{01}(\theta)\tilde{\omega}_1(\theta)$$

$$\tilde{\omega}_1(\theta)=\tilde{Q}_{12}(\theta)+\tilde{Q}_{10}(\theta)\tilde{\omega}_0(\theta)$$

$$\tilde{Q}_{01}(\theta)=\frac{\lambda}{\lambda+\theta},\ \tilde{Q}_{10}(\theta)=\tilde{\pi}(v+\theta),\ \tilde{Q}_{12}(\theta)=\frac{v}{v+\theta}(1-\tilde{\pi}(v+\theta))$$

$$\bar{\omega}_0 = \frac{1}{\lambda} + p_{01}\bar{\omega}_1$$

$$\bar{\omega}_1 = \frac{1}{v}(1-\tilde{\pi}(v)) + p_{10}\bar{\omega}_0.$$

Whence, taking into account that the initial condition is $e_0$, we obtain

$$\bar{\omega}_0 = \frac{\frac{1}{\lambda} + \frac{1}{v}(1-\tilde{\pi}(v))}{p_{12}}$$

$$\tilde{\omega}_0(\theta) = \frac{\tilde{Q}_{01}(\theta)\tilde{Q}_{12}(\theta)}{1-\tilde{Q}_{01}(\theta)\tilde{Q}_{10}(\theta)} = \frac{\lambda v(1-\tilde{\pi}(v+\theta))}{(v+\theta)(\lambda+\theta-\lambda\tilde{\pi}(v+\theta))}$$

It may be possible to investigate the asymptotic $\omega(t)$ in various conditions. In particular, $p_{12} \to 0$, when $v \to \infty$ at the fixed average value $\overline{s}_1$ of message service time in the system, when $\overline{s}_1 \to 0$, at the fixed value of intensity $v$ of disaster arrival. In work [22], as a research model of the asymptotic $\omega(t)$ it is possible to choose, Prob {the disaster will arrive during a busy period of the system} $= 1-\tilde{\pi}(v) \to 0$.

If as a small parameter we use $\varepsilon = 1-\tilde{\pi}(v)$, then as per theorem 6.1 [22], it is possible to formulate the following result for the asymptotic $\omega(t)$

$$\lim_{\varepsilon \to 0} P(\varepsilon\zeta \geq t) = e^{-\frac{1}{\bar{\omega}_0}t} = \exp\{-\frac{p_{12}}{\frac{1}{\lambda}+\frac{1}{v}(1-\tilde{\pi}(v))}t\} = \exp\{-\frac{v\lambda(1-\tilde{\pi}(v)}{v+\lambda(1-\tilde{\pi}(v))}\}t$$

*2.5. Queue Length*

The distribution of queue length in the system can be investigated with the method of additional variables and with the help of an embedded Markov chain. In the present work, both methods will be considered. Let $p_i(x)$

$$p_{i(x)} = \lim_{t \to \infty} P[m(t) = i, x < z < x + dx \,|\, m(0) = 0]$$

be the probability that in a system in stationary condition there are $i$ messages, and the message which is already taking place in the device is served at time

$z$, $x < z < x + dx$. For a definition of $p_i(x)$, we shall use the method of additional variables and in a standard way, for probabilities $p_i(x)$, we shall make the following system of integral-differential equations.

$$\frac{dp_1(x)}{dx} + (\lambda + v + s(x))p_1(x) = 0$$

$$\frac{dp_2(x)}{dx} + (\lambda + v + s(x))p_2(x) = \lambda g_1 p_1(x)$$

$$\frac{dp_i(x)}{\partial x} + (\lambda + v + s(x))p_i(x) = \sum_{n=1}^{i-1} \lambda g_{i-n} p_n(x)$$

with boundary conditions and normalizing condition of

$$\lambda p_0 = \int_0^\infty p_1(x)s(x)dx + v\sum_{i\geq 1} \int_0^\infty p_i(x)dx$$

$$p_1(0) = \int_0^\infty p_2(x)s(x)dx + \lambda g_1 p_0$$

$$p_i(0) = \int_0^\infty p_{i+1}(x)s(x)dx + \lambda g_i p_0$$

$$p_0 + \sum_{i\geq 1} \int_0^\infty p_i(x)dx = 1.$$

Here $s(x)$ is a conditional probability that the moment of an end of message service lies in an interval, provided that the message was not served at time $x$

$$s(x) = \frac{S(x)}{1 - \int_0^x S(x)dx} \quad \text{or} \quad S(x) = s(x)\exp[-\int_0^x s(x)dx].$$

Let us enter the probability generating function (PGF)

$$P(z,x) = \sum_{i=1}^{\infty} z^i p_i(x), \quad P(z) = \int_0^\infty P(z,x)dx, \quad P(1) + p_0 = 1$$

Then for the PGF distributions of queue length we shall obtain

$$\frac{\partial P(z,x)}{\partial x} + (\lambda + v + s(x) - \lambda G(z))P(z,x) = 0$$

$$zP(z,0) = \int_0^\infty P(z,x)s(x)dx - z\int_0^\infty p_1(x)s(x)dx + z\lambda p_0 G(z).$$

Using boundary conditions, we obtain

$$\begin{aligned} P(z,x) &= P(z,0)(1-S(x))\exp[-\{\lambda(1-G(z))+v\}]x \\ P(z,0) &= z\frac{p_0[\lambda(1-G(z))+v]-v}{S[\lambda(1-G(z))+v]-z} \\ P(z) &= P(z,0)\frac{1-S[\lambda(1-G(z))+v]}{\lambda(1-G(z))+v} \end{aligned}$$

Whence for the LST of the PGF we obtain

$$\begin{aligned} P(z,\theta) &= P(z,0)\frac{1-\tilde{S}(\theta+v+\lambda(1-G(z))}{\theta+v+\lambda(1-G(z)} \\ &= z\frac{p_0[\lambda(1-G(z))+v]-v}{S[\lambda(1-G(z))+v]-z}\cdot\frac{1-\tilde{S}(\theta+v+\lambda(1-G(z))}{\theta+v+\lambda(1-G(z)} \end{aligned}$$

$$P(z) = z\frac{p_0[\lambda(1-G(z))+v]-v}{S[\lambda(1-G(z))+v]-z}\cdot\frac{1-S[\lambda(1-G(z))+v]}{\lambda(1-G(z))+v}$$

The value of $p_0$ has been determined above.

The PGF of the distributions of the number of requests in system $Q(z)$ is defined by the formula

$$\begin{aligned} Q(z) &= p_0 + P(z) \\ &= p_0 + z\frac{p_0[\lambda(1-G(z))+v]-v}{\tilde{S}[\lambda(1-G(z))+v]-z}\cdot\frac{1-\tilde{S}[\lambda(1-G(z))+v]}{\lambda(1-G(z))+v} \end{aligned}$$

Through simple transformations of the formula, it is possible to present it in the standard form

$$zQ(z) = \{Q(z)-p_0+zp_0\}\tilde{S}[\lambda(1-G(z))+v] + z\frac{v\{1-\tilde{S}[\lambda(1-G(z))+v]\}}{\lambda(1-G(z))+v} \quad .$$

In the equation, the first component is the PGF of the demands of number distributions in the standard system $M^x/G/1$ on the assumption that no disasters took place, and the second component characterizes the probability of disaster approach in the holding time of a given message.

*2.6. Distribution of Expectation Time in the Queue and of Sojourn Time in the System*

Let us enter the notations:

$w, \overline{w}, \tilde{w}(\theta), W(t)$ are respectively the r.v., average value, LST and DF of a message's expectation time in the queue;

$u, \overline{u}, \tilde{u}(\theta), U(t)$ are respectively the r.v., average value, LST and DF of the sojourn time of messages in the system.

As it is known from [21], the LST of a message's expectation time in the queue can be defined with the help of the LST of the queue's length $\tilde{P}(z, \theta)$

$$\tilde{w}(\theta) = p_0 + \frac{\tilde{P}(\tilde{S}(\theta), \theta)}{\tilde{S}(\theta)}$$

On the other hand, the LST of the sojourn time of messages in the system is defined similarly to that of the busy period of a system

$$\begin{aligned}\tilde{u}(\theta) &= \tilde{w}(\theta+v)\tilde{S}(\theta+v) + \frac{v}{\theta+v}(1-\tilde{w}(\theta+v)\tilde{S}(\theta+v)) \\ &= \frac{v+\theta\tilde{w}(\theta+v)\tilde{S}(\theta+v)}{\theta+v}.\end{aligned}$$

Average values of message expectation times in the queue $w$ and sojourn time in system $\overline{u}$ are defined by equations

$$\overline{w} = -\tilde{w}'(\theta)|_{\theta=0}, \qquad \overline{u} = -\tilde{u}'(\theta)|_{\theta=0}.$$

$$\overline{u} = \frac{1-\tilde{w}(v)\tilde{S}(v)}{v}.$$

We shall note that it is also possible to define $\overline{w}$ with the help of Little's formula

$$\overline{w} = \overline{N} / \lambda \overline{g}.$$

$\overline{N}$ is an average quantity of messages in queue which is defined by

$$\overline{N} = -P'(z)\Big|_{z=1}$$

*2.7. Estimation of System Characteristics with Queue Length Consideration*

In the model research performed in the previous section, a standard assumption for the queue theory concerning message length was accepted. It was considered that all messages are identical and they have a normalized individual length. Such an assumption is justified in the case when the service time of a message does not depend on its length or when the indicated random variables are weakly correlated. However, in the case of strong correlation between the indicated random variables, for example in information transferring systems, in computer networks, etc., the application of the standard approach has a large margin of error [21,23,24].

In the present section, the characteristics of model $M^x/G/1$ will be investigated in view of the dependency between message service time and its length.

Let the messages entering a system be characterized (except at service time $\xi$) also by length $\varepsilon$. Let us accept that $\varepsilon$ are random variables that are entirely independent of the lengths of other messages. R.v. $\varepsilon$ are frequently interpreted as a memory capacity used by the message from the moment of its arrival in the system to the moment of its end of service. Further we shall consider, that r.v. $\xi$ are dependant only on the length of the message.

Dependence between r.v. $\varepsilon$ and $\xi$ in general is set with the help of joint DF

$$F(x,t) = P\{\varepsilon < x, \xi < t\}$$

Let $\Re(t)$ be the total memory capacity which has been used by all messages in the system at time $t$. Clearly, $\Re(t)$ is equal to the sum of lengths of expected $\Re_1(t)$ and served $\Re_2(t)$ messages at time $t$.

$$\Re(t) = \Re_1(t) + \Re_2(t)$$

Let us note through $D(x,t) = P\{\Re(t) < x\}$ the distribution function of r.v. $\Re(t)$, and through

$D_i(x,t) = P\{\Re_i(t) < x\}$ the DF of r.v. $\Re_i(t)$.

Let $\delta(s,\theta)$ and $\delta_i(s,\theta)$ be the transformations of the Laplace-Stieltjes DF, $D(x,t)$ , $D_i(x,t)$

$$\delta(s,\theta) = \int_0^\infty \int_0^\infty e^{-(sx+\theta t)} dD(x,t),\ \delta_i(s,\theta) = \int_0^\infty \int_0^\infty e^{-(sx+\theta t)} dD_i(x,t)$$

and $\alpha(s,\theta)$ be the transformations of the Laplace-Stieltjes DF $F(x,t)$

$$\alpha(s,\theta)=\int_0^\infty\int_0^\infty e^{-(sx+\theta t)}dF(x,t)$$

Let $\varphi(x)$ DF r.v. $\varepsilon$ be the lengths of messages arriving in the system, then $\varphi(x)=F(x,\infty)$, $S(t)=F(t,\infty)$.

From the properties of the LST it follows, that

$$\delta(s,\theta)=\delta_1(s,\theta)\delta_2(s,\theta)$$

Let $x(t)$ be the amount of time that a message service has been on a server before time $t$. Then for the suspended Distribution Function $E_y(x)=P\{\Re(t)<x/x(t)=y\}$ and for its LST we have

$$dE_y(x)=(1-S(y))^{-1}\int_y^\infty dF(x,u)$$

$$e_y(s)=(1-S(y))^{-1}\int_0^\infty e^{-sx}\int_y^\infty dF(x,u).$$

Let us enter the following notations

$$H_y(x)=(1-S(y))E_y(x),\ dH_y(x)=\int_{u=y}^\infty dF(x,u).$$

If at time $t$ the system is empty, then the total capacity is $\Re(t)=0$, and if it is busy, we have $\Re(t)=\Re_1(t)+\Re_2(t)$. Let us define the DF $\Re(t)$ provided that at time $t$ in the system there are $n$ messages and from the start of a message service that is on a server at time $t$, time passed on the server is $y$, $x(t)=y$. In view of the independence of random variables $\Re_1(t)$ and $\Re_2(t)$ we have:

$$P\{\Re(t)<x/n,y\}=P\{\Re_1(t)<x/n,y\}*P\{\Re_2(t)<x/n,y\} \quad (1)$$

where (*) is a sign of Laplace-Stieltjes convolution.

Let us define components (1). As the lengths of expected messages are independent, for the first component (1) we obtain

$$\begin{aligned} P\{\Re_1(t) < x/n, y\} &= P\{\Re_1(t) < x/n\} \\ &= (\varphi(x))_*^{n-1} \\ &= \underbrace{\varphi(x) * \varphi(x) * \ldots * \varphi(x)}_{n-1} \end{aligned}$$

Obviously, the second component is equal to

$$P\{\Re_2(t) < x/n, y\} = P\{\Re_2(t) < x/y\} = E_y(x)$$

Hence,

$$P\{\Re(t) < x/n, y\} = (\varphi(x))_*^{n-1} * E_y(x)$$

Whence for unconditional DF $D(x,t) = P\{\Re(t) < x\}$ we have

$$D(x,t) = P_0(t) + \sum_{n=1}^{\infty} \int_0^{\infty} p(n,y,t)[(\varphi(x))_*^{n-1} * E_y(x)]dy \tag{2}$$

Passing in (2) to a limit $t \to \infty$ and to a transformation of Laplace-Stieltjes for $\delta(s)$ we obtain:

$$\begin{aligned} \delta(s) &= P_0 + \sum_{n=1}^{\infty} (\varphi(s))^{n-1} \int_0^{\infty} p(n,y) e_y(s) dy \\ &= P_0 + \int_0^{\infty} P(\varphi(s), y) \frac{e_y(s)}{\varphi(s)} dy \end{aligned} \tag{3}$$

Substituting in (3) the value of probability generating function $P(z,x)$ we obtain

$$\delta(s) = P_0 \Big[1 + \int_0^{\infty} \frac{[\lambda(1 - G(\varphi(s))) + v] - v}{S[\lambda(1 - G(\varphi(s))) + v] - \varphi(s)} (1 - S(y)) \cdot$$
$$\exp[-\{\lambda(1 - G(\varphi(s))) + v\} y] e_y(s) dy \Big]$$

As $dF(x,u) = dS(u/\varepsilon = x) d\varphi(x)$, so

$$\int_0^\infty (1-S(y))\exp[-\{\lambda(1-G(\varphi(s)))+v\}y]e_y(s)dy =$$

$$= \int_0^\infty e^{-sx}d\varphi(x)\int_0^\infty dS(u/\varepsilon = x)\int_0^u \exp[-\{\lambda(1-G(\varphi(s)))+v\}y]dy =$$

$$= \frac{\varphi(s)-\alpha(s,\lambda(1-G(\varphi(s)))+v)}{\lambda(1-G(\varphi(s)))+v}.$$

Whence for the LST $\delta(s)$ of a random variable $\Re$ we obtain

$$\delta(s) = P_0\{1+\frac{[\lambda(1-G(\varphi(s)))+v]-v}{S[\lambda(1-G(\varphi(s)))+v]-\varphi(s)}\cdot\frac{\varphi(s)-\alpha(s,\lambda(1-G(\varphi(s)))+v)}{\lambda(1-G(\varphi(s)))+v}\}$$

The LST of length (capacity) for messages expected in queue $\delta_1(s)$ is defined by the equation:

$$\delta(s) = P_0\{1+\frac{[\lambda(1-G(\varphi(s)))+v]-v}{S[\lambda(1-G(\varphi(s)))+v]-\varphi(s)}\cdot\frac{1-S[\lambda(1-G(\varphi(s)))+v]}{\lambda(1-G(\varphi(s)))+v}\}$$

### 3. G-Networks Models with List Oriented Deletions

First of all, the model of a nonuniform G-Network with list oriented deletions which includes N servers, C type requests and H type signals will be considered. The supposition is that the incoming flow of the requests and signals is Poissonian, and the in-time service of requests in the network nodes is allocated exponentially. We denote as $\lambda_i^{(k)}(\bar{n})$, $\lambda_{i,k}^-(\vec{n})$ the intensity of the entrance of k type requests and h type signals in i node queue when the network is in $\vec{n}$ state; the intensity of k type requests service in i node as $\mu_{i,k}(\vec{n})$ when the network is in $\vec{n}$ state. After service completion in the i node the k type requests may either leave the network (with $d_i^{(k)}$ probability) or pass to j node.

In the second case they can pass to j node either as m type requests (with $P_{ij}^{+(k,m)}$ probability) or as h type signals (with $Q_{i,j}^{k,h}$ probability). The probabilities of passings meet the following requirement:

$$\sum_{j=1}^{N}\sum_{m=1}^{C}P_{ij}^{+(k,m)} + \sum_{j=1}^{N}\sum_{h=1}^{C}Q_{ij}^{k,h} + d_i^{(k)} = 1. \quad (4)$$

Each h type signal is compared to some list of $S_h$ nodes visited by it. The role of the signal lies in the deletion of one request from the queue of each node it visits. The length of $S_h$ list denoted as $L_h$ can be ending or infinite.

Entering in j node queue, an h type signal tries to delete one request from this queue. The probability of request deletion depends on the type of signal and the request deleted, on the network state, on the position of x node in $S_h$ list. This probability is denoted as $D_{j,\ m,\ h,\ x}$. Let $\eta(h, L_h, j, m)$ be a number of pair (j,m) occurence in $L_h$ length list connected with h type signal.

If in one of the nodes the request deletion attempt ends by "failure" then the signal disappears. Otherwise, the signal deletes requests in all nodes of the list and with $\alpha_{L_h+1,l}$ probability adds one new request in the definite queue. This queue in the list is denoted by $(L_h+1)$. The request added to this queue can be considered as the result of request deletion by $L_h$ signal from the queues included in the list. It is clear that the probability of "failure" deletion attempt will be equal to $1-\alpha_{L_h+1,l}$.

### *3.1. Definitions and Denotations*

Below, the network state is denoted as vector $\vec{n}=(\vec{n}_1,\vec{n}_2,\ldots,\vec{n}_N)$, $\vec{n}_i$ component of which describes the state of i node. The number of messages in i node is denoted as $|\vec{n}_i|$, and the state of network nodes is introduced by $\vec{n}_i=(r_{i,1},r_{i,2},\ldots,r_{i,\infty})$ vector, $r_{i,x}$ component of which shows x type request in the queue.

Let $(\vec{n}_i-e_{i,k})$ – be the state of i node when it includes $|\vec{n}_i|-1$ requests while k type requests are a unit less in the node; v(h,x) – is the node number which is on x position in $S_h$ list, and z(x) – is the request type deleted in this node upon the signal entrance; $\vec{y}$ – is the vector of deleted request types. Two following subsets are entered: $\Gamma(h)$ and $\Lambda(h)$ – sets of all possible lists of requests types of $L_h$ length and correspondingly smaller than $L_h$. If $\vec{y}$ components are the list elements from $\Gamma(h)$ set, then $|\vec{y}|=L_h$ and for $\Lambda(h)$ set – $|\vec{y}|<L_h$.

Let $\lambda_{\mathrm{i}}^{(\mathrm{k})}(\mathrm{n})$, $\lambda_{i,k}^{-}(\vec{n})$ and $\mu_{\mathrm{ik}}(\mathrm{n})$ meet the Henderson-Tailer requirements and have the form

$$\mu_{ik}(\overline{n})=\mu_{ik}\frac{\varphi(\overline{n}-\overline{e}_{ik})}{\psi(\overline{n})}\quad \lambda_{ik}^{-}(\vec{n})=\lambda_{ik}^{-}\frac{\varphi(\vec{n})}{\psi(\vec{n})}$$

$$\lambda_i^{(k)}(\overline{n})=\lambda_i^{(k)}\frac{\varphi(\overline{n})}{\psi(\overline{n})},\quad i=1..N,\ k=1..C, \tag{5}$$

where $\varphi(\bar{n})$ and $\psi(\bar{n})$ – are arbitrary non-negative functions [4].

$Z(h,\vec{n},\vec{u})$ is denoted as the probability of that after $|\vec{u}|$ request deletion by h type signal the network will appear in $\bar{n}$ state:

$$Z(h,\bar{n},\bar{e}_{ik}) = D_{ik,h1}\frac{\varphi(\bar{n}-\bar{e}_{ik})}{\varphi(\bar{n})}, Z(h,\bar{n},\bar{u}+\bar{e}_{ik}) = Z(h,\bar{n},\bar{u})D_{ik,h,|\bar{u}|+1}\frac{\varphi(\bar{n}+\bar{e}_{|\bar{u}|}-\bar{e}_{ik})}{\varphi(\bar{n}+\bar{e}_{|\bar{u}|})}$$

*3.2. Network Stationary Distribution*

The network stationary distribution is denoted as $\pi(\bar{n})$.

**Theorem 1.** For G-Network if a system of non-linear equations of the flow

$$\rho_{i,k} = \frac{\lambda_i^{(k)} + \sum_{j=1}^{N}\sum_{m=1}^{C}\mu_{j,m}\rho_{j,m}P_{ji}^{+(m,k)} + \Lambda_{i,k}^{+}}{\mu_{i,k} + \lambda_{i,k}^{-} + \Lambda_{i,k}^{-}} \tag{6}$$

where

$$\Lambda_{i,k}^{+} = \sum_{j=1}^{N}\sum_{h=1}^{H}\sum_{\vec{y}\in\Gamma(h)}\mu_{j,m}\rho_{j,m}Q_{j,i}^{m,h}\prod_{x=1}^{L_h}(\rho_{v(h,x),z(x)}D_{v(h,x),z(x),h,x})\alpha_{i,k}\mathbf{I}_{\{L_h<\infty\}}\mathbf{I}_{\{v(h,L_h+1)=i\}}$$

$$\Lambda_{i,k}^{-} = \sum_{j=1}^{N}\sum_{m=1}^{C}\sum_{\vec{y}\in\Lambda(h)}\mu_{j,m}\rho_{j,m}Q_{j,i}^{m,h}\prod_{x=1}^{|\vec{y}|}(\rho_{v(h,x),z(x)}D_{v(h,x),z(x),h,x})D_{i,k,h,|\vec{y}|+1}\mathbf{I}_{\{v(h,|\vec{y}|+1)=i\}}$$

has a positive solution $\rho_{i,k} > 0 \forall i,k$ , for each i node the condition

$\rho_i = \sum_{k=1}^{C}\rho_{ik} < 1$ is satisfied,

then there exists a network steady-state distribution, the conditions of partial balance are satisfied, and the stationary distribution $\pi(\bar{n})$ has the product form

$$\pi(\vec{n}) = G\psi(\vec{n})\prod_{i=1}^{N}\prod_{k=1}^{C}\rho_{ik}^{n_{ik}} \tag{7}$$

where G is the normalising constant.

To prove the theorem it is sufficient to substitute (6), (7) into the equation of network flows global balance (equations of Chapman-Kolmogorov).

Next is the formulation of conditions for the existence of a solution for the system of non-linear equations of the flow and stability of the G-Network concerned. For this purpose the methods of G-Networks stability research developed in [4,15] will be used.

The G-Network model with dynamic parameters discussed above is denoted as S(n) and the G-Network model as S, the parameters of which do not depend on the network state vector. Let $\pi(\vec{n})$ and $\hat{\pi}(\vec{n})$ – be the stationary distributions corresponding to S(n) and S networks. Thus, the conditions necessary for S(n) network multiplicativity and stability are formulated as follows [4].

**Theorem 2.** If the parameters of entrance and service of requests in S(n) network have the form (4), and the probability of deletion of one k type request from i node may be introduced in form

$$Z_{ik}(\vec{n}) = Z_{ik}\frac{\varphi(\vec{n}-e_{ik})}{\varphi(\vec{n})}$$

then for the stationary distributions of S(n) and S networks takes place

$$\pi(\vec{n}) =_{\tilde{N}} \psi(\vec{n})\ \hat{\pi}(\vec{n}),$$

the equations of both networks' flows coincide; the conditions and domain of stability of S(n) and S networks coincide.

It follows from theorem 2 that for S(n) network stability it is necessary and sufficient for $P^+$ routing matrixes of all types of requests to be transitive.

## 4. Description of the G-Network Model Functioning in a Random Environment

The supposition is that the environment dynamics can be described with the help of some Markovian chain with continuous time $\{\xi(t), t\geq 0\}$. For simplicity we admit that $\xi(t)$ is defined on the ending set of states Å. The infinitesimal chain matrix is denoted as Q, $q(u,v)$ components of which set the intensity of the chain transition from state u into the state v (u,v∈ E) and the stationary distribution of chain – as $\sigma=(\sigma_i, i\in Å)$.

$$\sigma_i \sum_{u\in E} q(i,u) = \sum_{j\in E} \sigma_j q(j,i), \quad i\in E.$$

$$\sum_{i\in E} \sigma_i = 1$$

The G-Network parameters (intensity of entrance and service of requests, routing probability of requests and signals, probability of requests deletion) depend on the state of Markovian chain $\xi(t)$, describing the environment dynamics. If the chain is in u state, then the network parameters are correspondingly equal to:

$$\lambda_i^{(k)}(\vec{n},u) = \lambda_i^{(k)}(u)\frac{\varphi(\vec{n})}{\psi(\vec{n})}, \quad i=1..N,\ k=1..C,$$

$$\mu_{ik}(\vec{n},u) = \mu_{ik}(u)\frac{\varphi(\vec{n}-\bar{e}_{ik})}{\psi(\vec{n})}, \quad Z_{ik}(\vec{n},u) = Z_{ik}(u)\frac{\varphi(\vec{n}-e_{ik})}{\varphi(\vec{n})} \quad (u\in Å)$$

For each u state of environment (u∈Å), probability of $P_{ij}^{+(k,m)}(u)$, $Q_{ij}^{+(k,h)}(u)$, $d_i^{(k)}(u)$ routing of requests and signals meets the requirement (4).

The functioning of the network in question can be described with the help of Markovian process $\beta(t)=(\vec{n}(t),\xi(t))$ . If $\beta(t)$ has a stationary distribution then the next theorem sets the conditions of the multiplicativity thereof.

**Theorem 3.** If in the discussed G-Network for each u (u∈Å) state the system of non-linear equations of the flow

$$\rho_{i,k}(u)=\frac{\lambda_i^{(k)}(u)+\sum_{j=1}^{N}\sum_{m=1}^{C}\mu_{j,k}(u)\rho_{j,m}(u)P_{ji}^{+(m,k)}+\Lambda_{i,k}^{+}(u)}{\mu_{i,k}(u)+\Lambda_{i,k}^{-}(u)} \tag{8}$$

has the positive solution $\rho_{i,k}(u)>\forall\ i,k$ , which satisfies the following conditions:

1. for all u (u∈Å) there is equality $\rho_{i,k}(u)=\rho_{i,k}$
2. for each i node there is inequality $\rho_i(u)=\sum_{k=1}^{C}\rho_{ik}(u)<1$

then there exist steady-state conditions of the network, partial balance conditions are satisfied, the stationary distribution of the network $\pi(\vec{n})$ has the product form

$$\pi(\vec{n},u)=G\sigma_u\psi(\vec{n})\prod_{i=1}^{N}\prod_{k=1}^{C}\rho_{ik}^{n_{ik}}(u) \tag{9}$$

where G is the normalising constant.

$\Lambda_{i,k}^{+}(u)$, $\Lambda_{i,k}^{-}(u)$ are defined analogously to the first model.

Similarly to theorem 2 for the stability of the discussed network it is necessary and sufficient for all u (u∈Å), $P^{+}(u)$ routing matrixes to be transitive.

In special cases, from the discussed model when $\psi(\vec{n})=1$, the results represented in [3,4,17] can be achieved.

## 5. Conclusions

The models and methods studied in this paper may be used to both predict the consequences of various types of attacks and intrusions as well as to assess the impact or effectiveness of protection systems on the productivity of a particular node, and of the network as a whole. The research of G-Networks stability has revealed that the existence of signals deleting regular messages in the network triggers the enlargement of the network domain of stability. Network administrators may utilize such outcomes for the purpose of regular “vaccination” of the network by generating in it the agents that

imitate functions of certain types of viruses which delete particular types of regular messages, for example, based on the information contained therein, the maximum permissible time limit for storing the information, etc. On the other hand, the developed models may also be used in the modeling of networks with information cashing and multiaddress transmission of message packets with the help of new mechanisms for message distribution control and management for the purpose of balancing the various network nodes load.

## References

[1] Artalejo J.R. G-networks: A versatile approach for work removal in queuing networks. // European Journal of Operational Research 126 (2000), 233±249.J.M.
[2] Gelenbe E. Random neural networks with negative and positive signals and product form solution. // Neural Computation 1 (1989), 502±510.
[3] Chao X., Miyazawa M., Pinedo M. Queuing Networks: customers, signals and product form solution. Pergamon Press, New York, (1999).
[4] Kerobyan Kh.V., Tovmasyan A.S. Multiplicativity and stability G-network. // ITM, 2, 2002.
[5] Henderson W., Taylor P.G. State-dependent signaling in queuing networks. // Adv. Appl. Probab. 26 (2) (1994) 436–455.
[6] Kerobyan Kh.V., Harutyunyan E.A., Tovmasyan A.S. Investigation and Optimization of Multimedia System Parameters by G-networks. IASTED 14-th Int. Conf. “Modelling and Simulation” (MS-2003), feb. 24–26, Palm Springs, USA.
[7] Jain G., Sigman K. A Pollaczek-Khintchine formula for M/G/1 queues with disasters. // J. Appl. Prob., 33, 1996, 1191–1200.
[8] Yang W.S., Chae K.C. A note on the GI/M/1 queue with Poisson negative arrivals // J. Appl. Prob., 38, 2001, 1081–1085.
[9] Chakka R., Harrison P.G. A Markov Modulated Multi-server Queue with Negative Customers-The MM Cpp/GE/c/L G-Queue. // Acta Informatica, 37, 2001, 881–919.
[10] Chao X. A queuing network model with catastrophes and product form solution. // Oper. Res. Letters, 18, 1995, 75–79.
[11] Boucherie R.J., Boxma O.J. The workload in the M/G/1 queue with work removal./Res. Rep., CWI, Amsterdam, 1995.
[12] Harrison P.G., Pitel E. The M/G/1 queue with negative customers. // Res. Rep., Imperial College, London, 1995.
[13] Dudin A.N., Karolik A.V. BMAP/SM/ queue with Markovian input of disaster and non-instantaneous recovery. Performance Evauluation, 895 (2000), pp. 1–13.
[14] Semenova O.V. An Optimal Threshold Control for a BMAP/SM/1 System with Map Disaster Flow, Automation and Remote Control 64 (9), September 2003, p. 1442–1454. Fourneau J.-M., Quessette F., Verchere D. Application of modulated G-networks to the control of ATM, 1996.
[15] Fourneau J.-M., Kloul L., Verchere D. Multiple class G-networks with list-oriented deletions. // European Journal of Operational Research 126 (2000) 250±272.
[16] Fourneau J.-M., Quessette F., Verchere D. Application of modulated G-networks to the control of ATM, 1996.
[17] Zhu Y. Markovian Queuing Networks in a Random Environment. // Oper. Research Letters, 1994.
[18] Koroljuk V.S., Swishchuk A.V. Random Evolutions. -Kluwer Academic, Dordrecht, 1994.
[19] Li, Q.L. and Zhao, Y. A MAP/G/1 queue with negative customers. Revised for Queueing Systems, 37, (2003), pp. 1–43.
[20] Kerobyan Kh.V., Internet. Basics & Applications. // Preface by Professor Leonard Kleinrock. Yerevan. Pabl. of Enc. Armenica, 2003. p. 256.
[21] Gnedenko B.V., Kovalenro I.N., Introduction to Queueing Theory. Moscow, Nauka, 1987, p. 336.
[22] Koroljuk V.S., Turbin A.F. Semi-Markov process and applications. Kiev, Naukova Dumka, 1976, p. 256.
[23] Tikhonenko O.M., Klimovitch K.G. Analysis of Random Length Queueing Systems with Restricted Summarized Size. Probl. Trans. Inform. 2001, v. 37, n. 1, pp. 78–88.
[24] Tikhonenko O.M., The problem of Determination of Summarized Messages volume in Queueing Systems and its Applications. J. Inf. Process. Cybern. EIK. 1987, v. 23, n. 7, pp. 339–352.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Resource Sharing Performance Models for Estimating Internet Service Characteristic in View of Restrictions on Buffer Capacity

Khanik KEROBYAN[a] and Khatchik VARDANYAN[b]
[a] *Institute of Information and Automation Problems National Academy of Sciences, Republic of Armenia, Russian-Armenian (Slavon) State University, Armenian Association of Computer and Information Systems, kerobyan@yandex.ru*
[b] *Yerevan State University*

**Abstract.** As a result of the development of net technologies and high-speed channels of data translation, new net services and systems have emerged; a new orientation in the field of net technologies & service creation has been established, namely, functional programming on the net service level. Functional programming technology is used to create an effective system for the rapid search and extraction of necessary data from vast net stores, to streamline the data and to develop decisions with the help of appropriate net service resources. Based on the above, a massive industry for data extraction has been developed. The success of these technologies in the net service market is derived from the effective organization of such systems. It should be noted that the effectivity of net services (systems for data fusion, support for decision making, etc), and the initial Quality of Service (QoS) depends on the parameters and properties of the net environment. The net environment has an essential influence on request receiving processes, i.e. on incoming traffic & on the serveice of replying files. Net environments define important characteristics of incoming traffic, such as its nature, interval distribution between incoming requests and their intercorrelation. And for replying processimportant characteristics are net environment influence on time for data translation by channels and channels loading. All of which are the subject of research on the influence of net traffic properties and parameters of net services characteristics, as well as the development of new models and methods for data translation and net channel resource division. This paper presents a decompositional approach to the modeling and analysis of characteristics of Internet service performance, with consideration to access line loading and different methods of their throughput distribution. The approach in based on the functional-time division of request processing in Internet services. As a mathematical model of request transfers on access lines, the combined queuing models Mn|Gn|1|N + MmGm|K|0 are considered, with registration of different restriction options on buffer capacity and on the number of requests in the system, request service time dependence on length, and different strategies for the division of system resources between request streams. In the first subsystem of the model the requests are processed by the Resource Sharing protocol, and in the second subsystem, by a K access line without waiting. The characteristics of Internet service performance with finite & infinite sources of requests, different strategies of resource access line division, and the distributed function of request length and service time, are investigated and defined.

**Keywords.** Data fusion systems, queuing model, internet service performance, quality of service, processor and resource sharing discipline, combined queuing models Mn|Gn|1|N + MmGm|K|0

## 1. Introduction

Over the past few years, the use of Internet services has undergone tremendous growth. For a short time, Internet applications evolved from standard document-retrieval functionality to advanced multimedia services and on-line systems of access to multidimensional databases. Following an exponential growth in the quantity of users connected to a global network, and the subsequent rapid increase in the number of Internet applications and advanced search, acquisition and communication information technologies, there has been a sharp explosion of traffic both in separate sites, and in all networks as a whole. Under the forecast of Internet experts, network traffic will increase substantially in a very short time, a large share of which will be the traffic that is caused by various agencies and service systems.

Therefore, as noted by L. Kleinrock [1], one of the main barriers to the continuing success of the Internet in the near future is network overload, which in many cases leads to unacceptably long response times, particularly for real-time applications like the WWW, telephony, E-business systems, multimedia systems and other on-line services. In a large amount of works, besides the research and measurement of traffic produced by various applications, the problems of both the optimal division of resources of data communication lines and the development of new principles and tools for the creation of on-line services are also considered. It should be noted that the ability to deliver a QoS guarantee to the end-user gives Internet service providers (ISPs) an enhanced competitive edge. In order to make decisions in respect to problems concerning the delivery of QoS, it is imperative to analyze the features of application functioning, to share out the main QoS factors, and to estimate the influence of a network environment on productivity and QoS applications at all stages of its life cycle, including design, use, development and modernization [2–5].

As research demonstrates, the services using IP technologies and working in a LAN, WAN or Internet environment have a number of common characteristics, the analysis of which permits the development of universal methods and tools to forecast and estimate productivity and, in the end, formulate general principles to maintain QoS at required levels [6,7]. Among such features let us note: the specific, modulated traffic characterized by long-time dependence and self-similarity with periodic bursts; streams of input and output information strongly distinguished from each other according to capacity; the complex, and in many systems, distributed structure of user services; the presence of restrictions on response time, channel throughput, buffer storage capacity, etc.; the strong correlation between the delivery-communication time on connection channels and transmitted information capacity, etc. [8–13].

QoS is an integral concept that includes various aspects of application organization, and depends both on specificity and requirements, and on features of its network environment. All of the abovementioned features of services complicate mathematical models used for the analysis and estimation of their productivity and QoS level. Currently, the models and methods of queue theory are widely used for the modeling of various services [14,15].

It can be noted, however, that the use of standard queue theory models, and in particular, of models using a Poisson approximation for the input stream, results in the large errors and excessive estimations encountered in the characteristics of researched services, owing to the specificity of traffic circulating in an IP-network [16–20].

On the other hand, the consideration of the following factors leads to complications in the service model: the time required to transfer a request through a channel is

dependent on its length; the restrictions placed on the capacity of storage of expected and served requests; the various ways that service resources are divided between separate types and categories of requests. The above features require both the development of new approaches to service modeling and the creation of their respective mathematical models [21–24].

In the present paper, an approach to service modeling, based on the functional decomposition of circulation in service streams, is presented. It is used to develop and investigate a series of non-standard models describing the functioning of services in different stages of request service as divided in time and independent from each other. The models take into account the specificity and restrictions inherent to the separate stages of request service and lead to the definition of general assumptions concerning the characteristics of service.

The principle of functional-time division of the stages of request service is based on the proposed decompositional approach.

We select three functionally different stages of user request processing in Internet services.

The first stage consists in the reception, registration and identification of arrived requests. The product of the given stage is either the deviation of a user request or the creation of a request for service. The rejection of requests can take place, for example, in the following cases: if the request was incorrectly formulated, if the user is, for whatever reason, blacklisted, etc.

The second stage consists in the processing of requests. At the time of processing, requests are passed on to various service subsystems, where they are processed and transformed. The characteristics of this stage depend both on the type and content of the request and specificity of service, composition and structure of its subsystems, and on principles of computing process organization. The product of this stage is the generation of an answer message and/or an outgoing file.

The third stage consists in the transfer/delivery of the answer message or file to the user through communication channels. The characteristics of a stage depend on its use of information transfer channels, its methods of resource division, and its principles of outgoing message buffering organization. It should be noted that the selected stages differ from each other not only functionally, but also on a number of important statistical characteristics, the structure of service subsystems, and on the requirements and restrictions imposed.

The characteristics of the different stages will now be considered.

If the informativeness of a request and the correctness of its representation are key for the first stage, and the composition and structure of the subsystems necessary for its processing is central to the second stage, then for the third stage, the capacity of an outgoing file, the capacity of the buffer selected for its storage, the required time of delivery, the throughput and loading of the information transfer channel are vital.

From a statistical point of view, the fractal, modulated, bursty character of entering traffic, the independence of requests capacity and time of its processing are typical of the first stage [11,14,27].

The second stage is characterized by the strong correlation between the type and service time of a request.

The third stage is distinguished by the strong correlation between delivery time, message length, channel loading, and principles of restricted channel resource division [22,28].

From the abovementioned, it should be noted that at the creation of an Internet service it is necessary to investigate and estimate the characteristics of both stages of user request reception and cutting (direct track) and the stage of transfer/delivery of an answer through communication channels (return track).

When analyzing the first track, it is important to choose the entering traffic model, the principles of queue forming, and the protocols and structure of the request service.

The most typical requirements placed on a traffic model, besides an adequate description of the real stream of entering messages, are that it possess an analytically closed presentation and that its respective mathematical model be simple upon investigation. For the analysis of the characteristics of the first track it is convenient to use one-channel service system models with a self-similar stream of entering streams. The following suppositions can be made in respect to the aforementioned models: all entering requests have a standardized length; requests differ from each other by type, category, service discipline, time and structure of service; the total quantity of requests in the model is restricted.

Numerous investigations and measurements of traffic in the different services and systems working in IP environments have shown that the most common models of traffic with self-similarity and long-time dependence properties [8] are the models of chaotic maps, fractional Brownian motion (FBM) [28], fractional autoregressive integrated moving average (FARIMA) [11], and also those using the Markovian Arrival Process (MAP) model formalism [30,35]. MAP models of traffic and their natural generalization, the Batch Markovian Arrival Process (BMAP) [31], were for the first time presented and studied by M. Neuts [29,30] and by his students [31–34]. The works [29–34] are devoted to the research of different models of service systems with MAP stream of entering requests. In particular, in [9,15,31–33] the MAP/G/1 models with restrictions, different types of vocations and with a finite number of entering MAP streams and their applications, have been considered in order to investigate the performance of different Internet applications.

The problems of researching the characteristics of the reception and processing of user requests in services working in IP environments have been considered in more detail in [19].

Let us now analyze the third stage – the track through which the answer messages are transferred to the service user. The basic resources required for the fulfillment of the given stage are a server, a buffer to store answer messages, and a channel for data translation.

Every answer message is characterized by its type, length and requested service level. Answer messages stored in the buffer are saved according the following principles:

- sharing out the total capacity (finite or infinite) for all types of messages in a sector of the buffer storage;
- sharing out different sectors (finite or infinite capacity) for the storage of different message types or their groups.

Of course, when a message is restricted from being saved in the buffer because it has reached its maximum capacity, the message is lost or blocked.

In order to provide the requested QoS level and a timely delivery of answer messages, the rented shared out channels and/or common usage channels are more frequently used, and their throughput is divided between the simultaneous translation of messages.

In addition to the two abovementioned options of channel resource division, the numerous methods of throughput division can also be used, e.g. the methods of main, proportional (correct), general and egotistical division, positional-balanced methods, et cetera [21,27,36].

A peculiarity of the communication track is also the necessity of registering the dependence between message transfer/delivery time and its capacity, and the registration of restrictions on channel throughput and the exit buffer of the system.

In publications on queue theory, insufficient attention is devoted to research on service systems that take into account the abovementioned peculiarities. The different queue models with random length request registration are considered in [23]; the models with registration of message service time dependency on length are considered in [22], and the models with one stream of entering requests, with registration of restrictions on buffer storage capacity and message service time dependency on length are considered in [24].

Unfortunately, works devoted to models with different request streams, combined restrictions, and with registration of service time dependency on request length are not known by the author.

In the present work, in order to investigate the characteristics of the communication track, the combined models $M_n|G_n|1|N$ (PS) + $M_mG_m|K|0$ with registration of different options of restrictions on buffer capacity and on the number of requests in the system, requests service time dependency on length, and different strategies of system resource division between request streams are considered.

## 2. Model Description

Let the total throughput of system C be divided between two groups of request streams in such a manner that for the stream from the first group there is a shared out part $C_1$ of system throughput and for the stream from the second group, a shared out part $C_2$. The part of throughput ($C_1$) is divided between requests of different streams with the help of one of the above considered methods, and the $C_2$ part is equally distributed between service servers K, i.e. for each server, 1/K part of the $C_2$ throughput is shared out.

Let us consider a service system with 2 groups of request streams and K+1 servers. The first group includes n Poisson streams of requests with parameters $\lambda_i^{(1)}$, $i=\overline{1,n}$, and the second group contains m streams with parameters $\lambda_i^{(2)}$, $i=\overline{1,m}$. The request service in the system is organized in the following manner.

The requests from the first group (with $\gamma_i^{(1)}$, $i=\overline{1,n}$ parameters) are served on the first server (number 1) in correspondence with the model $M_n|G_n|1|N$ (PS) [21], with the processor division (PS) protocol. Every request from the $i^{th}$ stream of the first group is characterized with random length $\gamma_i^{(1)}$, $i=\overline{1,n}$ that is independent from other request lengths and entering time in the system. The service time of a request from $i^{th}$ stream $\gamma_i^{(1)}$, $i=\overline{1,n}$ depends on its length and is proposed with the joint distribution function (DF) $F_i^{(1)}(x,t)=P\{\gamma_i^{(1)}<x,\xi_i^{(1)}<t\}$, $i=\overline{1,n}$. If we note the DF of random value

$\gamma_i^{(1)}, \xi_i^{(1)}$, $i = \overline{1,n}$ as $L_i^{(1)}(x) = P\{\gamma_i^{(1)} < x\}$ and $B_i^{(1)}(t) = P\{\xi_i^{(1)} < t\}$ then we have $L_i^{(1)}(x) = F_i^{(1)}(x,\infty)$, $B_i^{(1)}(t) = F_i^{(1)}(\infty,t)$.

If there are R requests from the first group in the system, then the requests from the i[th] stream $i = \overline{1,n}$ are served with $f_i^{(1)}(r_i,R)$, $R = \sum_{i=1}^{n} r_i$ rate.

Requests from the second group (with $\lambda_i^{(2)}$, $i = \overline{1,m}$ parameters) are served on servers number $\overline{2,k+1}$ in correspondence with model $M_mG_m|K|0$ [20], i.e. every request is served on one of $k$ servers.

The request from the i[th] stream of the second group is characterized with random length $\gamma_i^{(2)}$, $i = \overline{1,m}$ and service time $\xi_i^{(2)}$, $i = \overline{1,m}$, that is dependent on its length and joint distribution function $F_i^{(2)}(x,t) = P\{\gamma_i^{(2)} < x, \xi_i^{(2)} < t\}$, $i = \overline{1,m}$, after which, the distribution functions of random value $\gamma_i^{(2)}, \xi_i^{(2)}$, $i = \overline{1,m}$, $L_i^{(2)}(x) = P\{\gamma_i^{(2)} < x\}$ and $B_i^{(2)}(t) = P\{\xi_i^{(2)} < t\}$ are determined from $L_i^{(2)}(x) = F_i^{(2)}(x,\infty)$, $B_i^{(2)}(t) = F_i^{(2)}(\infty,t)$.

Let $\eta(t)$, $\eta_1(t)$, $\eta_2(t)$, $\eta(t) = \eta_1(t) + \eta_2(t)$, be respectively the total number of all requests, requests from the first and second groups that are in the system at time *t,* and $\sigma(t)$, $\sigma_1(t)$, $\sigma_2(t)$ be respectively the overall length of all requests, requests from the first and second groups that are in the system at time *t* $\sigma(t) = \sigma_1(t) + \sigma_2(t)$.

In the present paper, the characteristics of the system are investigated with the following options of restrictions on its parameters:

1. the restrictions on parameters are absent;
2. the total quantity of requests in the system is restricted;
3. the total length of requests in the system is restricted;
4. the quantity of requests from the first, second, or both groups in the system is restricted;
5. the total length of requests from the first, second, or both groups in the system is restricted;
6. various combinations of the previous 5 points.

The following options of system resource division are considered:

1. the proportional distribution of system throughput between all serving requests [21]

$$f_i^{(1)}(r_i,R) = \frac{1}{R},\ i = \overline{1,n},\ R = \sum_{i=1}^{n} r_i$$

2. the throughput of the system is distributed between streams in proportion to requests being served by a given stream, inside of which it is distributed equally between all serviced requests [19]

$$f_i^{(1)}(r_i,R)=\frac{r_i}{R}\cdot\frac{1}{r_i}=\frac{1}{R},\ i=\overline{1,n},\ R=\sum_{i=1}^{n}r_i$$

3. the throughput of the system is distributed between streams in accordance with option 2, and inside the i[th] stream the requests are served with $f_i^{(1)}(r_i),\ i=\overline{1,n},$ rate [27,36],

$$f_i^{(1)}(r_i,R)=\frac{r_i}{R}f_i^{(1)}(r_i),\ i=\overline{1,n},\ R=\sum_{i=1}^{n}r_i$$

The following options of request sources are considered:

- infinite source of requests from the i[th] stream with $\lambda_i^{(1)},\ i=\overline{1,n}$ parameter;
- finite source of requests from the i[th] stream with $\lambda_i^{(1)}(r_i),\ i=\overline{1,n}$ parameter;
- finite source of requests from the i[th] stream the parameter of which is dependent on the system condition vector $\lambda_i^{(1)}(r_1,r_2,..,r_n,),\ i=\overline{1,n}$ .

Cases of random and ordered numbering of requests in the system are considered.

Let us examine in further detail the case where both the total length of requests from both streams (V) and the quantity of requests from each group N and K are restricted.

If at time *t* the request from the i[th] stream of the first (second) group enters the system and there are N (K) requests in given group or $\sigma(t+0)+\gamma_i^{(1)}>V$ $(\sigma(t+0)+\gamma_i^{(2)}>V)$, then the request is rejected – and is lost.

The model will be investigated supposing that the numbering of both the requests being served in the first group and the servers serving the requests in the second group of streams is random; i.e. if in the system there are m requests in the first group of streams and a new (m+1)[th] request enters on condition that it has not been lost, then it can take with equal probability each of the possible (m+1) positions. In the case of requests in the second group of streams, if *r* servers are busy and a new (r+1)[th] request enters, then it can take with equal probability each of the free servers.

In the present paper the following characteristics of the system are investigated:

- the probability that the system will be busy;
- the probability of loss of requests from the first, second, or both groups of streams;
- the stationary probability of existence, in the system, of requests from the first, second or both groups of streams;
- the stationary density of probabilities that the system will exist in different conditions.

For the purpose of investigating the model's characteristics the method of supplementary variable input is used in this paper. The piece-linear Markovian process is considered.

$Z(t) = (\overline{\eta}_1(t), \overline{\eta}_2(t), \sigma(t), \overline{x}(t), \overline{y}(t))$, where

$\overline{x}(t) = (x_1(t), x_2(t), ...., x_{\eta_1(t)}(t))$, $\overline{y}(t) = (y_1(t), y_2(t), ...., y_{\eta_2(t)}(t))$

are vectors of the remaining time of service for the first and second groups of streams, and $\overline{\eta}_1(t) = (\eta_{11}(t), \eta_{12}(t), ...., \eta_{1n}(t))$, $\overline{\eta}_2(t) = (\eta_{21}(t), \eta_{22}(t), ...., \eta_{2m}(t))$ are vectors (ij), the components of which are equal to the number of requests of the j[th] stream from the i[th] group (i=1,2) that are in the system at time *t*. If at time *t* the system is empty, then $\sigma(t) = 0$ and supplementary variables are absent. Let such a condition of the $Z(t)$ process be noted by (0).

Let us make the following notations:

$$P(\overline{r}_1, \overline{r}_2, z, \overline{x}, \overline{y}) =$$

$$\lim_{t\to\infty} \frac{\partial^{r_1}\partial^{r_2}}{\partial x_{11}..\partial x_{1r_1}\partial y_{21}..\partial y_{2r_2}} P(\overline{\eta}_1(t) = \overline{r}_1, \overline{\eta}_2(t) = \overline{r}_2, \sigma(t) \le z, \overline{x}(t) \le \overline{x}, \overline{y}(t) \le \overline{y}),$$

$$r_1 = \sum_{i=1}^{n} r_{1i}, \; r_2 = \sum_{i=1}^{m} r_{2i},$$

are the stationary density of probabilities of process conditions $Z(t)$;

$$P(\overline{r}_1, \overline{r}_2) = \int_0^V \int_0^\infty .. \int_0^\infty P(\overline{r}_1, \overline{r}_2, z, \overline{x}, \overline{y})\, dz d\overline{x} d\overline{y} \tag{1}$$

are the stationary probabilities of process conditions $Z(t)$.

Considering the evolution of the process trajectory $Z(t)$ in infinitesimal time, for its stationary density of probabilities we can write the following differential equations of Kalmogorov, the solution of which is the following:

$$P(\overline{r}_1, \overline{r}_2, \overline{x}, \overline{y}) = P(0,0) \cdot \frac{\prod_{i=1}^{n} \lambda_{1i}^{r_{1i}} \prod_{j=1}^{m} \lambda_{2j}^{r_{2j}}}{r_{21}! r_{22}! ... r_{2m}!} \left[ \prod_{i=1}^{n} \prod_{e=1}^{r_{1i}} f_i(e) \right]^{-1} \times \tag{2}$$

$$\times H_{x_{11}}^{(1)} \times H_{x_{12}}^{(1)} \times .... \times H_{x_{1r}}^{(1)} \times H_{y_{21}}^{(2)} \times ..(v)$$

where

$$dH_{jy}^{(i)}(x) = \int_{u=y}^{\infty} dF_j^{(i)}(x,u)$$

and

$$H_{x_{11}}^{(1)} \times H_{x_{12}}^{(1)} \times .... \times H_{x_{1n}}^{(1)} \times H_{y_{21}}^{(2)} \times .... \times H_{y_{2m}}^{(2)}(x)$$

is the n+m-multiply Laplace-Stieltjes transform of

$$H_{x_{11}}^{(1)}(x), H_{x_{12}}^{(1)}(x), ..., H_{x_{1n}}^{(1)}(x),\ H_{y_{21}}^{(2)}(x), .., H_{y_{2m}}^{(2)}(x)\ \text{functions.}$$

Using (1,2) we find the stationary probabilities of system conditions

$$P(\overline{r}_1, \overline{r}_2) = P(0,0) \cdot \frac{\prod_{i=1}^{n} \lambda_{1i}^{r_{1i}} \prod_{j=1}^{m} \lambda_{2j}^{r_{2j}}}{r_{21}! r_{22}! ... r_{2m}!} \left[ \prod_{i=1}^{n} \prod_{e=1}^{r_{1i}} f_i(e) \right]^{-1} \times$$
$$R_1^{(1)}(V) \times ... \times R_n^{(1)}(V) \times R_1^{(2)}(V) \times .. R_m^{(2)}(V) \tag{3}$$

where

$$R_j^{(i)}(x) = \int_{u=0}^{x} \int_{y=0}^{\infty} y dF_j^{(i)}(u, y),\ i = 1, 2,$$

and $P(0,0)$ is defined from normalizing terms

$$\sum_{\overline{r}_1, \overline{r}_2} P(\overline{r}_1, \overline{r}_2) = 1$$

In case, when saving requests from streams of the first group, the sector of memory with $V_1$ capacity is shared out, and for requests from streams of the second group the sector of memory with $V_2$ capacity is shared out, then for stationary probabilities of system conditions we obtain

$$P(\overline{r}_1, \overline{r}_2) = P(0,0) \cdot \frac{\prod_{i=1}^{n} \lambda_{1i}^{r_{1i}} \prod_{j=1}^{m} \lambda_{2j}^{r_{2j}}}{r_{21}! r_{22}! ... r_{2m}!} \left[ \prod_{i=1}^{n} \prod_{e=1}^{r_{1i}} f_i(e) \right]^{-1} \times$$
$$R_1^{(1)}(V_1) \times .. \times R_n^{(1)}(V_1) \times R_1^{(2)}(V_2) \times .. R_m^{(2)}(V_2)$$

Let $P_e$ be the stationary probability of the loss of requests in the system. $P_e$ can be defined from the equilibrium equation [20,27]

$$(\lambda_1+\lambda_2)(1-P_e)=\sum_{r_2=1}^{k}\sum_{r_1=1}^{n}\frac{1}{r_1}\sum_{i=1}^{r_1}\int_0^\infty\!\!\ldots\int_0^\infty\mu_{1i}(x_{1i})P(\overline{r}_1,\overline{r}_2,\overline{x},\overline{y})d\,\overline{x}d\,\overline{y}+$$

$$+\sum_{r_1=1}^{n}\sum_{r_2=1}^{k}\sum_{i=1}^{r_2}\int_0^\infty\!\!\ldots\int_0^\infty\mu_{2i}(y_{2i})P(\overline{r}_1,\overline{r}_2,\overline{x},\overline{y})d\,\overline{x}d\,\overline{y}$$

where

$$M_i(x)=\int_0^\infty\frac{H_y^{(i)}(x)}{1-B_i(y)}dB_i(y).$$

## 3. Conclusions

In the present work, the system characteristics are investigated with all abovementioned options of parameters and restrictions, and also in the following particular cases:

- the request service time is linearly dependent on its length;
- the request service time is independent of its length;
- the request length has a geometrical or exponential distribution.

The model where requests from the second (first) group of streams, in some cases (if the general number of requests from a given group of streams reaches a critical level), can be served together with requests from the first (second) group of streams, is also considered.

## References

[1] Kh.V. Kerobyan, Internet. Basics & Applications. /Preface by Professor Leonard Kleinrock. Yerevan. Pabl. of Enc. Armenica, 2003. p. 256.
[2] T.F. Abdelzaher, Kang G. Shin, and N. Bhatti. Performance Guarantees for Web Server End-Systems: A Control-Theoretical Approach. IEEE Transactions on Parallel and Distributed Systems, v. 13, n. 1, pp. 80–96, January 2002.
[3] V. Almeida, M. Arlitt, and J. Rolia. Analyzing a Web based systems performance at multiple time scales. In Proceedings of the Practical Aspects of Performance Analysis Workshop (in conjunction with SIGMETRICS'02), Marina Del Ray, CA, June 2002.
[4] T. Ikenaga, K. Kawahara, T. Takine and Y. Oie, Analysis of Delayed Reservation Scheme in Server-based QoS Management Network, Proceedings of the 2003 International Conference on Advanced Information Networking and Applications (AINA 2003), pp. 140–145, Xi'an, China March 27–29, 2003.
[5] J.M. de Almeida, Cristina D. Murta, Adriana A. Oliveira, M.A. de Souza Mendes, and V.A.F. Almeida. Performance Analysis and Modeling of a WWW Internet Server. In Proceedings of the Fourth Int. Conf. on Telecom. Systems, pp. 297–300, Nashville, Tennessee USA, March 1996.
[6] J.E. Pitkow. Summary of WWW characterizations. World Wide Web, 2(1):3–13, January 1999.
[7] M. Arlitt and T. Jin. A Workload Characterization Study of the 1998 World Cup Web Site. IEEE Network, May/June 2000.
[8] V. Paxson, S. Floyd, Wide Area Traffic: The Failure of Poisson Modeling, IEEE/ACM Trans. Networking, 1995, v. 3, n. 3, pp. 226–244.
[9] Sh. Kasahara, Internet Traffic Modeling: Markovian Approach to Self-Similar Traffic and Prediction of Loss Probability for Finite Queues, IEICE Trans. Commun. V. E84–B, n. 8, 2001. 2134–2141.
[10] M. Abrams, S. Williams, G. Abdulla, S. Patel, R. Ribler, and E. Fox. Multimedia Traffice Analysis Using Chitra95. In ACM Multimedia '95 Conference, November 1995.

[11] A. Balamash and M. Krunz. A Parsimonious Multifractal Model for WWW Traffic. In Proceedings of the Web Engineering Workshop at the Networking 2002 Conference, Pisa, Italy, May 2002.
[12] W. Willinger, V. Paxson and M.S. Taqqu, Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic, in: A Practical Guide to Heavy Tails: Stat. Tech. for Analyzing Heavy Tailed Distributions, R. Adler, R. Feldman and M.S. Taqqu (Eds.), Birkhauser Verlag, Boston. 1998.
[13] W.-C. Lau, A. Erramilli, J.L. Wang and W. Willinger, Self-Similar Traffic Parameter Estimation: A Semi-Parametric Periodogram-Based Algorithm. Proc. of IEEE Globecom'95, Singapore, 1995, pp. 2225–2231.
[14] A. Erramilli, O. Narayan and W. Willinger. Experimental Queueing Analysis with Long-Range Dependent Packet Traffic, IEEE/ACM Transactions on Networking, 1996, v. 4, n. 2, pp. 209–223.
[15] A.T. Andersen, M. Neuts and B.F. Nielsen. A Markovian Approach for Modeling Packet Traffic with Long Range Dependence. IEEE JSAC, 1998, v. 16, n. 5, pp. 719–732.
[16] T. Ikenaga, K. Kawahara, T. Takine and Y. Oie, Analysis of Delayed Reservation Scheme in Server-based QoS Management Network, Proceedings of the 2003 International Conference on Advanced Information Networking and Applications (AINA 2003), pp. 140–145, Xi'an, China March 27–29, 2003.
[17] L. Kleinrock. Queueing Systems. Volume I: Theory, Wiley, 1975.
[18] X. Chao, M. Miyazawa and M. Pinedo, Queueing Networks: Customers, Signals and Product Form Solutions. Wiley. Chichester, 1999.
[19] Kh.V. Kerobyan, Computer Networks Random Models. Yerevan. Pabl. of EA RA, 1999, p. 210.
[20] B.V. Gnedenko, I.N. Kovalenro, Introduction to Queueing Theory. Moscow, Nauka, 1987, p. 336.
[21] C.F. Yashkov, Queueing Analysis for Compurers. Moscow, Radio&Svyaz, 1989, p. 216.
[22] O.M. Tikhonenko. Determination of Characteristics of Queueing Systems with Restricted Memory. J. Aut. And Telemekh., 1997, n. 6, pp. 105–110.
[23] B.A. Kac. On Service Random Length Messages and Queueing Theory. Proc. III Int. Conf. Of Queueing Theory. Moscow, MU Publ., 1976, v. 2, pp. 157–168.
[24] O.M. Tikhonenko, K.G. Klimovitch. Analysis of Random Length Queueing Systems with Restricted Summarized Size. Probl. Trans. Inform. 2001, v. 37, n. 1, pp. 78–88.
[25] B. Ryu and S. Lowen. Fractal Traffic Models for Internet Simulation, IEEE International Symposium on Computers Communications (ISCC), Juan-Les-Pins, France, July 2000.
[26] D.A. Menasce, V.A.F. Almeida, L.W. Dowdy. Capacity Planning and Performance Modeling: From Mainframes to Client-Server Systems. Prentice Hall, Inc. NJ, 1994.
[27] J.V.L. Beckers, I. Hendrawan and all. Generalized Processor Sharing Performance Models for Internet Access Lines. 9-th IFIP Int. Conf. ATM-2001, Budapest, 2001, pp. 1–12.
[28] B. Ryu, Fractal Network Traffic Modeling: Past, Present, and Future, Proc. 35th Allerton Conf. on Communication, Control, and Computing, Univ. Illinois at Urbaba-Champaign, Sep. 1997.
[29] M.F. Neuts. Matrix-geometric Solutions in Stochastic Models. Johns Hopkins University Press, Baltimore, MD, 1981.
[30] M.F. Neuts. Matrix-analytic Methods in Queueing Theory. In J.H. Dshalalow, editor, Advances in Queueing. Theory, Methods, and Open Problems, pp. 265–293. CRC Press, 1995.
[31] D.L. Lucantoni, The BMAP/G/1 Queue: A Tutorial, Models and Techniques for Perf. Eval. Of Comp. And Commun. Systems, Springer Verlag, 1993, pp. 330–358.
[32] S. Chakravarthy. Analysis of a finite M AP/G/1 queue with group services. Queueing Systems, 1993, v. 13, pp. 385–407.
[33] B. Sengupta and T. Takine, Distribution of Spatial Requirements for a MAP/G/1 Queue when Space and Service Times are Dependent, Queueing Systems, v. 22, n. 1, pp. 121–127, May 1996.
[34] D.M. Lucantoni, G.L. Further Transient Analysis of the BMAP/G/1 Queue. Stochastic Models, v. 11, n. 2, 1995, pp. 214–280.
[35] J.E. Diamond, A.S. Alfa. On approximating higher order MAPs with MAPs of order two. Queueing Systems, 2000, v. 34, n. 4, pp. 269–288.
[36] J.W. Cohen. The Multiple Phase Service Network with Generalized Processor Sharing. Acta Informatica, 1979, v. 12, n. 3, pp. 245–284.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Considerations of Possible Security-Related Nuclear Power Plants Incidents in Romania

Bogdan CONSTANTINESCU
*Institute of Atomic Physics, PO BOX MG-6, Bucharest, Romania*

**Abstract.** Two relevant cases are presented: the post-Chernobyl experience on Romanian territory and the situation of the Bulgarian Kozloduy Nuclear power plant (NPP), with a focus on aspects of situation monitoring, incident detection and response management. Four possible incident management strategies for NPP incidents (accidents) are discussed: prevention and limitation of dangerous effects, avoidance of impacts, defense against impacts, and adaptation to impacts.

**Keywords.** Nuclear power plants, incident, environment, response management

## 1. Chernobyl 1986 – an Overview of Situation Monitoring

Nuclear power plant (NPP) security has been the cause of much discussion in the last two years, especially in relation to potential terrorist actions such as various possible suicide attacks or illicit trafficking of radioactive materials. If such aspects are to be dealt with predominantly by the police, security services, and military, the "classical" problem of malfunctioning incidents (accidents) requires a more complex technical, political and social approach. Two such relevant cases will be discussed below: the post-Chernobyl experience on Romanian territory and the potential conflict between Romania and Bulgaria related to the old Bulgarian Kozloduy NPP.

On April 26, 1986, the Chernobyl nuclear power station in Ukraine suffered a major accident that was followed by a prolonged release into the atmosphere of large quantities of radioactive substances. The specific features of the release were a widespread distribution of radioactivity throughout the northern hemisphere, mainly across Europe. A contributing factor was the variation of meteorological conditions and wind regimes during the period of release. Activity transported by the multiple plumes from Chernobyl was measured not only in Northern and Southern Europe, but also in Canada, Japan and the United States. Only the Southern hemisphere remained free of contamination. This caused acute radiation injuries and deaths among plant workers and firemen. It also led to radiation exposure for thousands of persons involved in rescue and clean-up operations. There was severe radioactive contamination in the area ($10^{19}$ Bq total released radioactivity – approximately 300 times as much as for Hiroshima, from which $10^{18}$ Bq $^{131}I$ and $10^{17}$ Bq $^{134}Cs + ^{137}Cs$), resulting in the evacuation of people from a 30 km zone around the power plant. It became clear over the months following the accident that radioactive contamination of varying severity had also occurred in extensive areas of Eastern Europe, including Romania.

From a biological point of view, the most significant radioactive substances in the emissions from the accident were iodine, strontium, and plutonium. Different problems arise with different radioactive substances. Radioactive iodine is short-lived and practically disappeared some weeks after the accident. Its danger is due to the fact that, if inhaled or ingested, it accumulates in the thyroid gland, where it may deliver large radiation doses as it decays. The doses may result in impaired thyroid function and in possible thyroid cancer many years after the exposure Cesium is the element that clearly dominates the long-term radiological situation. Due to its penetrating radiation, cesium deposited on the ground may give an external dose. It may also enter the food chain and give an internal dose. It is eliminated metabolically in a matter of months. Cesium is relatively easy to measure. Plutonium and strontium, on the other hand, present difficulties in measurement, but there was relatively small amount of strontium in the fallout, and it was not dangerous unless ingested or inhaled. Very small amount of plutonium traveled far from the reactor site, but because of its chemical stability, it does not find its way easily into food chains.

In the first weeks following the accident, lethal doses were attained in local biota, notably in coniferous trees and voles (small mice) in the area within a few kilometers of the reactor. By autumn 1986, dose rates had fallen by a factor of 100, and by 1989, these local ecosystems had begun to recover. No sustained severe impacts on animal populations or ecosystems have been observed. Possible long-term genetic impacts and their significance remain to be studied [1].

A by-product of the environmental contamination was the contamination of foodstuffs produced in the affected areas. Although for some time after the accident key foodstuffs showed radioactivity levels exceeding the maximum levels permitted by the Codex Alimentarius (established by FAO and WHO, setting the maximum permitted level of radioactivity for foodstuffs moving in international trading, e.g. 370 Bq/kg of radiocaesium for milk products and 600 Bq/kg for any other food)., Wild food products – such as mushrooms, berries and game – from forests in the more affected areas, as well as fish from some European lakes, also exceeded Codex levels.

## 2. Chernobyl Accident Detection – Test Results

This paper summarizes the results obtained by measuring the radioactivity related to the nuclear accident from Chernobyl in a simple nuclear physics laboratory of Bucharest and discusses a potential role of small spectroscopy groups in possible future similar cases. So, since early May 1986, our Nuclear Spectroscopy Laboratory started an intense radiometric activity, by using common dosimetric and spectrometric devices available for population protection purposes. The results of the analysis of the measurements show that all the areas in Romania were affected by the radioactive cloud. The lowest concentrations (0.3–0.8 kBq/$m^2$) were measured in the Western Plain (the area Timisoara – Oradea – Satu Mare). High values (15–25 kBq/$m^2$, as compared to 555 kBq/$m^2$ for near the Chernobyl area) were reported at mountain stations (Parang, Fundata, Babele, Ceahlau – Toaca), in the Transylvania Plateau (Targu-Mures, Cluj), in the Eastern part of the country (Iasi – Tulcea – Buzau – Sf. Gheorghe Delta-Constanta), and in the Southern area (Bucharest – Pitesti – Tg. Jiu – Drobeta Tr. Severin). For the southeastern region of Romania, the radioactive particle deposition after the Chernobyl nuclear accident occurred mainly on 1–2 May, with rainfalls favoring the process. Its principal effect on population was radioactive exposure via

food consumption [2]. The contamination mechanisms were direct deposition of particles on existing plants and fruits during May 1–2, and consumption of contaminated milk products, eggs, meat in May-June 1986. Our activity was concentrated in three directions: environmental global gamma measurements, foodstuffs' contamination evaluation and measurement of $^{131}I$ in the thyroid. Using Our data obtained with Romanian type GAMMARAD portable dosimeters (based on Geiger-Mueller counters) show on May 3 a total gamma radioactivity level of 8.50–11μSv/h for Bucharest macadam (asphalt), but 20–25 μSv/h for water sewers (after raining), and 15–18 μSv for vegetation. For Campina (hill region-100 km north from Bucharest) on May 4, the gamma radioactivity level were 3.50-4 μSv/h (washed by rain macadam), 11μSv/h (vegetation) and for Sinaia (150 km north from Bucharest, in Carpati mountains), on May 5, 3 μSv/h (macadam) and 3.50–4 μSv/h (vegetation). Probably, the main radioactivity contribution in all cases was made by $^{131}I$. These data confirm the strong contamination of the Bucharest area and the importance of raining and washing processes, especially for macadam [3]. For foodstuffs (greens, milk products, meat, eggs) contamination evaluation, the extremely diverse samples were weighed and homogeneously encapsulated (except eggs) in three types of cylindrical transparent plastic containers (ϕ=5 cm and 1.5 cm height, ϕ=8 cm and 2.5 cm height and, especially for greens, ϕ=12 cm and 8 cm height). The measurements were performed using a lead-shielded CANBERRA Ge(Li) detector (70 cc volume, 10.5% efficiency, 2.8 keV resolution at 1332 keV $^{60}Co$). The spectra (1000s and 3000s) were recorded on a 1024 – channel CANBERRA-30 pulse-height analyzer. The calibration was made with $^{131}I$ and $^{152}Eu$ radioactive standard solutions. The global uncertainty was from 10% for cheese and meat to 20% for greens of very irregular appearance. The results are summarized in Tables 1, 2 and 3 [4].

The importance of washing greens in minimizing contamination is evident, especially in the first half of May, as well as the necessity to avoid milk product consumption during the same period (one month after the accident). Values in soils (e.g. $^{131}I$: 30–240 Bq/kg in the middle of May 1986, $^{137}Cs$: 45–200 Bq/kg for soil and 12–54 Bq/kg for sea sand) and in various grains (e.g. $^{131}I$: 180–360 Bq/kg for barley in the second part of May 1986, $^{137}Cs$: 135–630 Bq/kg for fodder and 15–3300 Bq/kg for granulated lucerne, depending on geographic zone) are reported. The scatter in radioactivity concentration values, especially for greens and cheese, can be explained by an significant dispersion of radioactive particles deposition values in the south-eastern region of Romania. For example, in the areas without rainfalls between May 1 and 10, 1986, the measured mean deposition values were 7500 Bq/m$^2$ for $^{131}I$ and 1100 Bq/m$^2$ for $^{137}Cs$. In the areas with intense rainfalls the values were up to $7.5\times10^4$ Bq/m$^2$ for $^{131}I$ and 18500 Bq/m$^2$ for $^{137}Cs$.

**Table 1.** Radioactive values for greens (Bq/kg "wet weight")

| period | 5–15 May 1986 | 15–31 May 1986 | October 1986 |
|---|---|---|---|
| $^{131}I$ unwashed | 5000–6200 | 37–185 | – |
| $^{131}I$ washed | 2300–1300 | 33–148 | – |
| $^{134}Cs$ unwashed | 185–370 | 10–75 | 7.5–26 |
| $^{134}Cs$ washed | 75–185 | 7.5–63 | 7.5–24 |
| $^{137}Cs$ unwashed | 333–777 | 19–166 | 12.5–44.5 |
| $^{137}Cs$ washed | 148–370 | 14.5–140 | 11.5–44.5 |

**Table 2.** Radioactive values for sheep cottage cheese (Bq/kg "wet weight")

| period | 7–15 May 1986 | 15 May–15 June 1986 | November 1986 |
|---|---|---|---|
| $^{131}I$ | 9250–25900 | 370–740 | – |
| $^{134}Cs$ | 1100–2200 | 55–95 | 75–110 |
| $^{137}Cs$ | 1850–3940 | 100–135 | 130–185 |

**Table 3.** $^{131}I$, $^{134}Cs$, $^{137}Cs$ concentration values for various foodstuffs (Bq/kg "wet weight")

| | time period | $^{131}I$ | $^{134}Cs$ | $^{137}Cs$ |
|---|---|---|---|---|
| eggs (values per egg) | 6–7 May 1986 | 110–270 | 10–15 | 15–33 |
| | 20–31 May 1986 | 12–37 | 6–10 | 10–15 |
| lamb meat (mean values) | 18–25 May 1986 | 18500 | 450 | 820 |
| honey (mean values) | 25–31 May 1986 | 110 | 55 | 98 |
| cherries and strawberries (unwashed) | 20 May–10 June 1986 | 45–110 | 20–65 | 65–165 |
| autumn fruit (unwashed) | August–Sept. 1986 | – | 20–65 | 40–150 |
| medicinal herbs | 18 May–6 June 1986 | 185–750 | 150–470 | 230–1000 |

Water bodies such as rivers, lakes and reservoirs can be, if contaminated, an important source of human radiation exposure because of their use for recreation, drinking and fishing. In the case of the Chernobyl accident this segment of the environment did not contribute significantly to the total radiation exposure of the population. It was estimated that the component of individual and collective doses that can be attributed to the water bodies and their products did not exceed 1 or 2 percent of the total exposure resulting from the accident. The contamination of the water system has not posed a public health problem during the last decade.

In Romania, the aquatic ecosystems were contaminated due to the specific transport processes created, the maximum concentration in the water only in May 1986, 2–3 days later than in the atmosphere. Concerning, An average concentrations of $^{137}Cs$ in surface water attained values of 50 Bq/m$^3$ were founding 1886, but only 7 Bq/m$^3$ in 1988. In Black Sea water, the same values were 230 and 60 Bq/m$^3$.

0.7 Bq/l, the maximum permissible concentration (MPC) for drinking water measured on May 1$^{st}$ after 6:30 am was exceeded (1.7 Bq/l). Since then the contamination level continuously increased to 20 Bq/l on May 2nd, 30.5 Bq/l on May 3rd, the maximum value being registered on May 4, 9.39 a.m. (49.3 Bq/l). After that, a period of decreasing contamination lasted until May 7, followed by a stationary state between May 7 and 13 with the contamination level in the range 3.7–5.6 Bq/l. The second period of decreasing contamination with fluctuating values was registered until 28 June. After this date, the tap water radioactivity was constantly below MPC. Measurements of the radioactivity of the underground waters in Bucharest showed the lack of any contamination during 1986. Unfortunately, underground waters were used only locally in Bucharest, because of some technical impediments. Under these circumstances, taking into account the major risk of epidemics as compared to radiological risks due to radioactive water consumption, the sanitation authority decided to continue the delivery of tap water from the river water supply [5].

## 3. Response Management in Romania After Chernobyl Accident

Some comments on response management aspects based on the 1986 Romanian experience are presented below.

A simple model to minimize contamination via foodstuffs can be deduced: careful washing of greens in the first two weeks after the accident, no milk and cheese in the first six weeks, limited eggs and meat consumption in the first three months and destruction of fodder in the first two weeks. The use of underground water is recommended. Fortunately, the consumption of contaminated products in May–June 1986 was strongly limited by monitoring and warnings, so their contribution to the internal dose was quite low. It could be prized that the population's additional Chernobyl internal irradiation in 1986 was compatible with usually applied medical irradiation, e.g. less than 2 mSv.

In the case of Chernobyl, as in many other radiological incidents, psychological effects were predominant. Information about the severity and significance of this contamination was often sparse and uneven; public opinion was uncertain and even many doctors were not sure how to interpret information that had become available. As a result, there was a loss of confidence in the information and in the countermeasures recommended.

In general, the most widespread countermeasures were those, which were not expected to impose, in the short time for which they were in effect, a significant burden on lifestyles or the economy. These included advice to wash fresh vegetables and fruit before consumption, advice not to use rainwater for drinking or cooking, and monitoring programs for citizens returning from potentially contaminated areas. In reality, experience has shown that even these types of measures had, in some cases, a negative impact that was not insignificant [6]. Protective actions having a more significant impact on dietary habits and imposing a relatively important economic and regulatory burden included restrictions or prohibitions on the marketing and consumption of milk, dairy products, fresh leafy vegetables and some types of meat, as well as the control of the outdoor grazing of dairy cattle. There was a minor disruption of normal life and economic activity in the affected areas. In particular, agricultural and forestry production was partially disturbed and some production losses were incurred.

After the Chernobyl accident, scientists who were not well versed in radiation effects attributed various biological and health effects to radiation exposure. These changes cannot be attributed to radiation exposure, especially when the normal incidence is unknown, and are much more likely to be due to psychological factors and stress. Attributing these effects to radiation not only increases psychological pressure on the population and provokes additional stress-related health problems, it also undermines confidence in the competence of radiation specialists. These observations are similar not only for Former Soviet Union (FSU) regions, but also for Romania.

The nature of these effects is complicated and it is wrong to dismiss them as irrational or to label them as "radiophobia." Many factors contribute to the development of this widespread association with nuclear bombs, either a lack of openness in the past on the part of governments, or the absence of intelligible explanations by scientists. Such effects are real and understandable, particularly in a mainly rural population whose work and recreation are closely interwoven with the land, where restrictions may have had to be imposed by authorities. Even physicians and others who might be looked to for guidance have often been confused. The result is that rumors multiply, fears increase, and any health problem is quickly attributed to a nuclear cause. Uncorroborated

narratives may become commonly held wisdom and unverifiable statistical data may be accepted with insufficient scrutiny.

For our country, as for psychological effects, we can mention a small rise (10–15%) in spontaneous abortions in some regions and a slight decline in pregnancy rates following the disaster. Similar effects were reported for Sweden, Norway and some regions of Russia. There are no data about induced abortions, strictly prohibited in 1986 in Romania. In our opinion, the 1986 activity of our Laboratory (foodstuffs measurements and $^{131}I$, $^{134}Cs + ^{137}Cs$ whole body control for a lot of people from Bucharest) contributed not only to a better understanding of the phenomena and to a better protection of individuals, but also to avoid possible psychological "radiophobia" effects in many people.

## 4. Kozloduy NPP Case – Possible Response Management Strategies

As concerning the case of the Bulgarian Kozloduy Nuclear power plant, it is widely recognized that environmental stress, especially environmental degradation could contribute, under certain political, economical and social conditions, to the appearance of serious conflicts mainly in the developing countries, e.g. in Central and Eastern Europe. Romania and Bulgaria are a potential example in this regard, in relation to their existing Nuclear power plants: four non-enveloped WWER – 440/230, 400 MW (a very old model developed during 1960s) and two enveloped WWER – 1000, 1000 MW (a model developed in the early 1980s) PWR (Pressurized Water Reactor) Soviet Units in Kozloduy near Danube (100 km from Bucharest) in Bulgaria and one 660 MW enveloped PHWR (Pressurized Heavy Water Reactor) Canadian CANDU Unit in Romania [7]. If the existing Romanian plant is very new (1996) and its CANDU type is unanimously recognized as having a high level of security, the four non-enveloped Soviet-Bulgarian reactors are old (one is from 1971), and small (up to now) incidents are often reported about them. These facts cause anxiety in both countries strongly affected by the Chernobyl accident [1]. Living in the shadow of an unsafe nuclear reactor is not much fun. But living in the dark and cold may be worse. All international experts reports conclude that it would be "technically and economically feasible" to meet electricity demand and still shut down the most dangerous reactors by the mid 2000s. Bulgaria's dilemma is that roughly a quarter of its electricity comes from the 230 – type reactors. The oldest pair (units 1 and 2) was shut down in 1991, due to fears about safety: as a consequence, in the past two winters the country was frequently reduced to eight hours of electricity a day. In 1993, one of the pair was patched up and reopened, and the other was started up in 1994, as well. The newest pair (5 and 6) at Kozloduy, are safer – both reactors are of the VVER 1000 MW design. But they often have to be taken out of use for running repairs, so tend to operate at only 50% of capacity. At present, Bulgaria must make a choice between nuclear pride and European Union goal, because EU has insisted that Sofia must close the four ageing reactors (400 MW) before 2006, as agreed in an accord allowing the country to begin EU membership talks. As concerns the "new" 1000 MW reactors, EU asks that they be closed in 2009 and 2011, respectively.

Response management strategies to this potentially threatening environmental change must be discussed. Four basic approaches that can be taken to enhance human security in both countries are considered [8]. The first is fundamentally a preventive strategy oriented toward minimizing, if not entirely averting the potential nuclear envi-

ronmental changes that threaten human security. The other three approaches presume that a (serious) incident will materialize, and thus are designed to reduce the vulnerability of human communities to it by avoiding the impacts of the changes by creating defenses against the impacts or by simply adapting to the changes. All of these options must be considered in relation to the 1986 Chernobyl accident experience (large area regional radiological impact, with serious consequences for health, agriculture and environment).

The first strategy – Preventing or Limiting Environmental Changes – involves special high cost technical and economical efforts (improving security systems, even replacing older reactors with modern nuclear ones or with classical power plants, or importing the necessary energy in the frame of a regional -international- arrangement). This strategy could be efficient if applied with priority on the "source" – the four old reactors WWER – 400 – 200 from Kozloduy: improving nuclear safety, fire protection, plant management and organization, quality assurance, operator training and qualification, conduct of operations maintenance, technical support and emergency planning. Such programs, which included international cooperation, are effectively applied (partially, because they are very expensive) in Bulgaria. Bulgaria could develop alternative energy sources: hydro, gas, coal, oil-based power plants. Unfortunately, the costs (a few billions dollars) are prohibitive at the moment for the countries, which are in a very difficult economical situation. Bulgaria could also replace the electricity obtained from Kozloduy by participating in the new Romanian CANDU nuclear power plants construction in Cernavoda (very safe facilities) or by importing electricity from other European Countries. All of these alternatives involve delicate financial aspects, and adequate international aid is essential. This is the standard strategy of aid agencies. The European Union used it in relation to Bulgaria through PHARE and TACIS programs and European Bank for Reconstruction and Development (BERD).

The second strategy – Avoidance of Impacts – seeks to avoid being in a position to be impacted by environmental threats should they materialize, for example by not establishing a home in an exclusion zone around the reactors – or by not using foodstuffs produced in such a zone. Impact avoidance presumes that the environmental threats can be anticipated and opportunities exist for avoiding exposure to them should they materialize. Environmental threats can be readily anticipated where there is a history of such events (in this respect, the post-Chernobyl experience is extremely useful). Even when a threat is foreseen, avoidance may not be an option for those who would be most directly impacted. Many people live in places where they are susceptible to human-induced catastrophes (e.g. chemical and nuclear) because they lack the means or the opportunity to locate elsewhere. This situation is worse in our countries due to the economical crisis and to the strong links between peasants and their land. It is oversimplistic to assume that all measures which would reduce future radioactive dose are beneficial and therefore should always be fully implemented. For example, one of the measures to be reconsidered is that of resettling people elsewhere. Moving people to an area of lower radioactive contamination will probably reduce dosage. Since further intake is considered to have a proportional effect on the future level of risk, relocation should reduce the risk of long-term radiation effects. However, it is known that the stress of extensive changes in lifestyle can have very serious psycho-social and even physical effects on people. A balance must be struck between potential reduction in dose and possible harm that might be avoided on the one hand and the possible detrimental and disruptive effects of resettlement.

The third strategy – Defense against Impacts – seeks to reduce vulnerability to environmental threats, not by avoiding them, but by taking measures that protect populations against adverse impacts. One example is the preventive distribution of potassium – iodine (KI) tablets for thyroid protection against I-131 radioisotope in the most potentially exposed area around the reactors. We could also consider the organization of a continuous surveillance network for radioactive emanations which could also be used for food, water and air control during an eventual accident, realization of a "healthy" (uncontaminated) food reserve.

The final strategy – Adaptation to Impacts – is adaptation or reaction to environmental changes once they take place. The experience achieved during the Chernobyl accident [2–4] is very useful in this case. So, a simple method of minimizing contamination via foodstuffs (the most prominent effect) was deduced: the strong washing of greens in the first two weeks after the accident, no milk and cheese in the first six weeks, limited eggs and meat consumption in the first three months and destruction of fodder in the first two weeks. The use of underground water is recommended. In the case of Chernobyl, as in many other radiological incidents, psychological effects have predominated. Information about the severity and significance of this contamination was often sparse and uneven; public opinion was uncertain and even many doctors were not sure how to interpret the information that did become available. We must underline the necessity of prompt, correct and sincere information by governmental authorities, essential in establishing a solid, confidential relation with the population and for minimizing psychological effects.

## 5. Conclusions

In Europe, the Nuclear Field situation (including NPPs, but also Uranium mining and processing, radioactive waste treatment, Plutonium reprocessing etc) will soon essentially depends on new rules giving the European Commission powers to supervise the safety of all nuclear installations. And until now, the problem of Soviet type NPPs (generally considered less safe than the enveloped western-type NPPs) was a very difficult test for the EU candidate countries during the process of their EU membership negotiation. For example, the Czech Republic closed the energy chapter in its EU accession talks in December 2001 only after one year of very tough discussions with Austria on the Temelin NPP safety improvements, as Austria asks for this as a condition of the accession to the EU. The Czech Republic spent a lot of money and efforts to fulfill the EU (mainly Austrian) requests. Lithuania has the same problem with its Ignalina NPP and, after receiving substantial EU financial help to improve nuclear security, will decide this year when the NPP will finally be shut-down (no later than 2008). Even Armenia announced last year that the unique Armenian NPP will be close in 2008. So, NPP security will continue to be a very important problem in the international public opinion many years from now.

In the case of new potential nuclear incidents, from our post-Chernobyl experience, as conclusions on how to organize a radioactive contamination measurements activity in non-dedicated small nuclear laboratories, we can mention:

- free access to many portable and robust radiometers;
- standardized measurement procedures for environmental and foodstuffs samples (specimens preparation, geometry);

- available standard radioactive solutions, point and volume sources;
- portable big volume NaI(Tl) counter, coupled to Single Channel Analyzers for $^{131}I$ - determination in thyroid.

We must add the necessity of prompt, correct and sincere information by governmental authorities, essential in establishing a solid, confidential relation with the population and for minimizing psychological effects. As concerns long-term effects (cancer and genetic anomalies), a serious international scientific effort to study, to cure and to avoid these phenomena is strongly recommended.

As concerns the response management strategies for nuclear incidents (accidents), for their efficient application we must find the answer to the following questions:

- what are the relative economic costs of these strategies – for Eastern European countries especially in connection to their integration in the EU;
- which proportion of society resources should be invested, as opposed to other priorities (food security, relative high rate of morbidity and mortality), etc.);
- to which extent the response strategies [9] adapted by one state add to the environmental or economical securities of another state;
- under what circumstances are states likely to opt for international co-operation as opposed to self-reliance in the pursuit of environmental security (see the potential role of the International Atomic Energy Agency – Vienna, European Union, or, why not, NATO);
- what roles can non-state actors, such as non-governmental organizations and corporations are expected to play in advancing environmental security.

All these aspects must be related to promoting bilateral (Bulgaria – Romania in the case of Kozloduy NPP) and international collaboration among scientists, politicians and academics, contributing to the integration of both our countries into the International and Development Community.

## References

[1] Gonzales, A.J. (1996) Chernobyl-Ten Years After, IAEA Bulletin 3.
[2] Constantinescu, B., Galeriu, D., Ivanov, E., Pascovici, G., Plostinaru, D. (1988) Determination of $^{131}I$, $^{134}Cs$, $^{137}Cs$ in Plants and Cheese after Chernobyl Accident in Romania, J. Radioanal. Nucl. Chem., Letters 128 (1), 15–21.
[3] Constantinescu, B., Dumitru, C., Puscalau, Mirela (1993) Some Aspects Concerning the Relationship Environmental Radioactivity Population in Romania after 1986, Romanian Reports in Physics 45 (7–8), 487–495.
[4] Constantinescu, B., Galeriu, D., Ivanov, E., Pascovici, G., Plostinaru, D. (1990) $^{131}I$, $^{134}Cs$, $^{137}Cs$ Concentrations in 1986 for Some Romanian Foodstuffs, J. Radioanal. Nucl. Chem., Letters 144 (6), 429–437.
[5] Oncescu M. (ed.) (1990) Artificial Radioactivity in Romania, Romanian Society for Radioprotectio.
[6] The International Chernobyl Project Assessment of Radiological Consequences and Evaluation of Protective Measures (1991) – Technical Reports, IAEA, Vienna.
[7] Soroos, M. (1997) Strategies for Enhancing Human Security in the Face of Global Change, NATO ARW "Environmental change, Adaptation and Security," Budapest, Hungary, October 9–12.
[8] Electricity Supply in Central and Eastern European Countries: the Role of Nuclear Energy (1993) IAEA Bulletin, vol. 35, no 1, 2–6.
[9] Le Prestre, P. (1997) Adapting to Environmental Insecurities: a Conceptual Model" – Philipps Le Prestre, NATO ARW "Environmental Change, Adaptation and Security," Budapest, Hungary, October 9–12.

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Blast Impact Assessment in the Iron Ore Mining Region of Ukraine

Mykola KHARYTONOV[a], Alexandr ZBEROVSKY[b] and Andriy BABIY[c]
[a]*State Agrarian University, Voroshilov st.25, Dnepropetrovsk, 49027, Ukraine, mykola_kh@yahoo.com*
[b]*National Mining Academy, K.Marx Av.19, Dnepropetrovsk, 49027, Ukraine, gis@creator.dp.ua*
[c]*Institute for Nature Management Problems and Ecology, National Academy of Sciences of Ukraine 6, Moskovskaya Street, Dnepropetrovsk, 49000, Ukraine, dea@email.dp.ua*

**Abstract.** This paper discusses environmental problems of blast impact assessment in the iron-ore mining region of Ukraine. There are multilateral and nonlinear connections between the air circulation parameters of the quarry's atmosphere and environment as a whole, between microclimate characteristics, geothermal and radiating parameters, and between the ecological and social characteristics of the investigated process. As such, the system "quarry–environment–human" is considered as a distributed system. Physical, heuristic and combined approaches utilizing different mathematical methods (statistics, numeric integration, etc) were considered. Problems concerning blast impacts around quarries on air, soil, water, humans, etc. are also examined.

**Keywords.** Iron-ore mining, quarry, blast impact assessment, aerology, pollution

## 1. Aerology of the Quarries

The aerology of quarries [1], as a component of the environmental protection of the area of mining, should promote the production process of required quantities of useful minerals without environmental degradation and natural ecosystems infringement, create comfortable working conditions for miners, and provide safe living conditions for populations in residential areas. Let us consider aspects of the aerology of quarries in the framework of the following system: "quarry – environment – human," henceforth referred to as the QEH system. The study of the aerology of quarries, revealing the essence and variety of environmental problems and the relationship between separate parts of the system, is a complicated matter. Consider a brief example of it in figure 1.

There are multilateral, complex, and nonlinear connections between the air circulation parameters of the quarry's atmosphere and environment as a whole, between microclimate characteristics, geothermal and radiating parameters, and between the ecological and social characteristics of the investigated process. In this sense, the QEH system can be considered a distributed system. All the relations between separate parts of this system are complex, changeable in time, and have a set of various constants in the equations describing quantitative characteristics of exchangeable processes. For this reason, when studying such a distributed system, it is usually inconvenient to separate

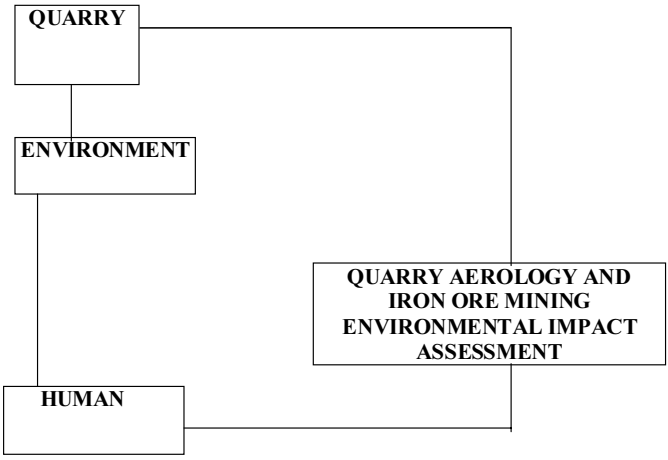



**Figure 1.** Scheme of relationships of different aspects of quarry aerology in the framework of the "quarry – environment – human" system.

the numerous phenomena of radiating, thermal, aerodynamic, geothermal, technical, ecological and social character, as they are closely bound among themselves and are diverse in nature, physical essence, and scale. It is possible to consider three approaches to overcome these difficulties.

### *1.1. Physical Approach*

The physical approach assumes the description of a complete picture of the aerology of quarries by the analysis of the physical laws controlling the separate processes and their set in the system. Such an approach allows for an improved understanding and description of the processes involved; it offers a mathematical interpretation and establishes some quantitative laws. Unfortunately, this approach is often impractical because of the complexity of the QEH system, and in practice can only be used to study the simplest processes of the aerology of quarries separately.

### *1.2. Heuristic Approach*

The heuristic approach establishes empirical connections by logical reasoning, typically utilizing the results of statistical analysis of the large quantity of numerical information characterizing various processes of the QEH system under various factors. Due to the distributed character of this system, the heuristic approach facilitates the use of various methods of mathematical statistics to establish regression equations connecting the characteristics of the investigated processes with the parameters that form them. Regression equations can be established in many cases, such as: "quarry radiating balance – humidity," "wind speed and environment temperature," "humidity, wind speed and environment temperature," "humidity, wind speed and environment temperature – atmosphere dust pollution – frequency of occupational diseases," or the more complex variant: "blasting mining works – geological conditions – atmosphere microclimate parameters – ecological damage." The regression equations contain a large number of parameters. A large number of samples are required to produce these equations, thus rendering the heuristic approach impractical.

### *1.3. Combined Approach*

The combined approach is a combination of the physical and heuristic approaches. From our point of view it is the most acceptable approach for studying a QEH system. The combined approach necessitates some research in establishing differential equations; each differential equation describes a separate part of the investigated phenomenon or process. The definition of constants in the equations is carried out by statistical analysis of information characterizing relationships between some problem aspects of quarry aerology and the primary complex of parameters.

## 2. Calculation of Atmospheric Parameters

The theory and methods for the calculation of atmospheric parameters in the quarries and the environment as a whole consist of a modeling of natural quarry ventilation dynamic schemes with the method of conform reflections. Several methods are applied:

- the straight flowing scheme for air circulation estimation in the quarry with simple configuration contour;
- the straight flowing scheme for air circulation estimation in the quarry with arbitrary configuration contour;
- the recirculation scheme for quarry ventilation estimation.

Thus, quarry aerology for open-pit mines exploits applications of the theory of functions of complex variables for computing concentrations of harmful substances and convective airflows in the quarries.

## 3. Harmful Substance Concentrations and Convective Air Flows

The dust-gas situation forecast in the quarry is based on the concentrations of components of harmful substances at an arbitrary point of the quarry space. From all point and linear sources, torches pass through that arbitrary point (Figure 2a).

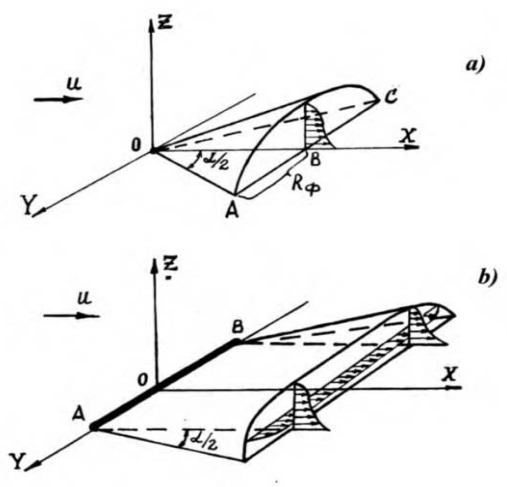



**Figure 2.** Scheme of the pollution spread by torches on the plain.

a) from a point source;
b) from a linear source.

Relations are obtained for the conditions of distribution of harmful substances by torches in the plain along a straight-line axis (Figure 2a and 2b). Gas or dust concentration in the torch point attribute with coordinates X, Y, Z can be determined by the following equations:

For the point source:

$$C = \frac{kG}{x^2 \phi^2 U} \exp \frac{-6.3\sqrt{\left(Y^2 + X^2\right)^{1.3}}}{\phi X^{1.3}} + C', mg/m^3 \quad (1)$$

For the linear source:

$$C = \frac{kG}{X \phi^2 Lu} \exp \frac{-3.9Z^2}{\phi X^2} + \text{C', mg/m}^3 \quad (2)$$

where:

$k$ – coefficient of proportionality obtained as the result of experiments

$G$ – pollution source intensity, *mg/s*

$U$ – airflow speed of a point source or a middle point on the linear source line, *m/s*

$L$ – linear source length, *m*

$C'$ – harmful substance concentration in the airflow to the source of pollution (background or control), *mg/m*$^3$

$\phi$ – angle tangents of pollution torch development

$U_1$ – minimal speed of airflow; $U_1 = 1m/s$

It is necessary to emphasize that equations (1) and (2) do not take into account the crooked-curving of the torch line axis of harmful substances spreading within the quarry space.

Since the stationary flow lines coincide with particle trajectories, it is suggested that the direction in which harmful substances are spread is such that the torch axis coincides with the airflow line passing through the source of harmful substances. A quarry plan showing the sources of harmful substances is located in the first quadrant of the XOY coordinate plane so that the OX direction coincides with the direction of the airflow (Figure 3).

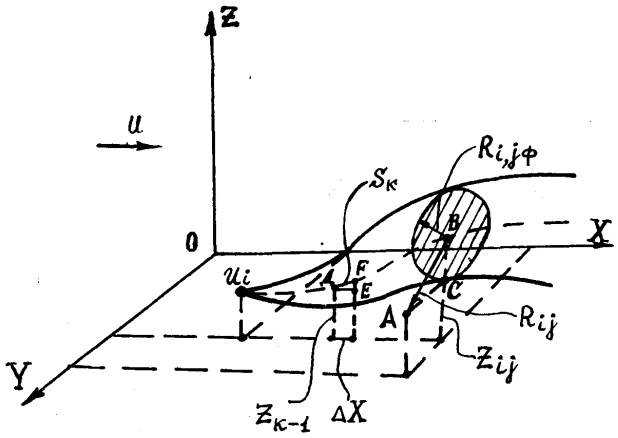



**Figure 3.** Scheme of the pollution spread by torches in the quarry.

A new algorithm based on the straight flowing scheme application for air circulation in the quarry was developed to calculate the airflow velocity and harmful substance content for each point of the quarry space [1].

## 4. The Nature of Convective Flow

The nature of convective flow in the quarry can be explained by two phenomena. First, above the heated horizontal sites of the terrestrial surface at a continuous receipt of solar radiation the vertical poles of rising air are shaped as free jets. Second, it is necessary to account for the "slope effect" attributed to the Archimedes force. The "slope effect" takes place if the heated surface is inclined. Both these effects are considered in the proposed model. A 2D flow mathematical model of air circulation in the quarry was developed, taking in account thermal convection from natural and artificial heat sources. Bi-dimensional non-stationary mixed convection was considered for the turbulent liquid in a "trench" with a curvilinear section. The model includes the famous Navier & Stokes heat conduction and diffusion equations, and an additional equation of heat transfer has been incorporated. The buoyancy force in the hydrodynamic equation is also taken into account. The solution of the system of differential equations for solving the problem of free thermal convection in the quarry was numerically computed [1]. The practical application of the proposed model is an attempt to establish a convection pattern arising in calm conditions from the heated part of the right quarry board at the center of the Krivbass region in the summer time (Figure 4).

The aerological method for computing the convective air flow in the quarries makes it possible to operatively and precisely predict speeds of convective flows at any point of the quarry space as a function of the size and direction of its structure, both for calm and windy conditions. This is important for the prediction of dust-gas phase situations both in the quarry and around it.

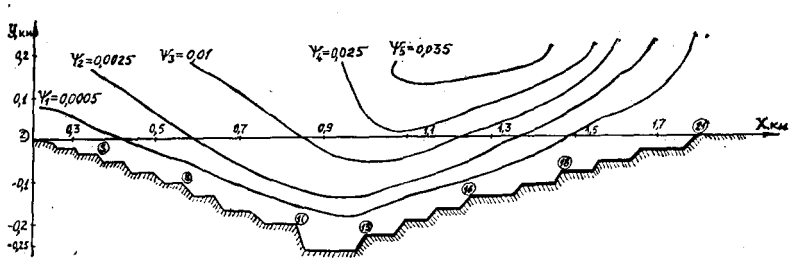



**Figure 4.** Canonic lines account for the disposition of whirlwind flow in the quarry in the case where the right part of the quarry board was heated up.

Several algorithms were developed and implemented for the practical implementation of the proposed methods for any given quarry:

1. "air" program for calculating atmospheric parameters in the case of the straight stream system of quarry ventilation;
2. "whirlwind" program for calculating atmospheric parameters in the case of the recirculation scheme of quarry ventilation;
3. "dust" program for calculating harmful substances in the quarry atmosphere;
4. "aeroterm" program for calculating atmospheric parameters of the quarry, taking into account wind and convective airflows;
5. "solver" program for calculating atmospheric parameters in the case of open iron ore pit-mines.
6. "cascade" program for calculating atmospheric parameters in the case of intensification by air screens.

The proposed algorithms and software were used for the estimation of circuits of quarry ventilation to define the levels of pollution in the Krivbass region so as to be able to propose recommendations. An example is presented in figure 5.

Thus, it was possible to determine the contours and volumes of zones with concentrations above 1MPC (maximum permissible concentration) based on calculation data on the distribution of harmful substances in the quarry atmosphere.

### *4.1. Microclimate Control, Atmospheric Parameters and Dust-Gas Cloud Formation Parameters*

Means for controlling microclimate in the quarry and methods of measuring atmospheric parameters and parameters of the process of birth and formation of dust-gas clouds include telemechanical control of meteorological parameters in an atmospheric framework with application lidars for the laser sounding of the quarry atmosphere, a laboratory and station for remote sensing, etc.

More than 40% of iron ore extracted in NIS countries by incurring explosions during open mining projects is taken from the quarries of the Krivbass. Annually, at Krivbass mining enterprises, up to 250 mass explosions with a charge of 600–800 (sometimes up to 1200) tons are set off simultaneously on 15 ledges. After a mass explosion, nitrogen oxide remains in the atmosphere for up to one hour and up to 6 hours within the mass of detonated rock. The significant volume of dust and poisonous gases

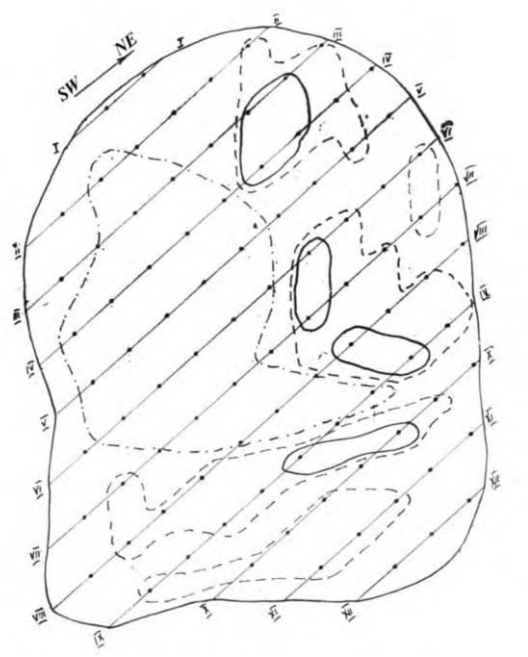



**Figure 5.** Contours of the recirculation zones and the location of dangerous zones of pollution under North-Eastern – South-Western wind direction.

emitted in the atmosphere is conducive to environmental degradation, the deterioration of the working conditions of miners, and is a serious health hazard to the population in nearby areas. Thus, modern computers must be used for the atmospheric monitoring and forecast of industrial pollution in the zones of activity of mining enterprises.

Air pollution is the most urgent ecological danger given the close relationship between human health and the environment.

Estimated levels of ecological danger for different types of safety norm violations in open-pit mines are shown in table 1.

**Table 1.** The levels of ecological danger for different types of violations of safety norms in open-pit mines

| | | |
|---|---|---|
| environment | the level of ecological danger | |
| | high | low |
| | types of ecological violations | |
| | activity | |
| | active | passive |
| air | dusting | dusting |
| | high level of toxic gases | |
| | sound and air waves | |
| bowels | infringement of geostratification | landscape changes |
| | loss | ground water level decreasing |
| | infringement of a fertile layer and biosystems | |
| | seismic waves | |

## 5. Process of Birth and Formation of Dust-Gas Clouds (DGC)

Observations on the process of the settling of iron-ore dust clouds in the conditions of local quarries confirmed that 100 tons of ES blasting produces 51 tons of dust and small particles (about 1000 mkm). The active time of gas emission is 2–24 hours. It was established that a cloud begins to spread just after blasting, and continues to do so for a period of 5–30 min. The harmful admixtures are spread in accordance to concentration levels, initial parameters of the dust-gas cloud, and meteorological atmospheric conditions. The maximum volume of simultaneously blasted explosive substances (ES) within one quarry of the South Mining Factory in the Krivbass iron-ore region is approximately 1000 tons. Typically, 63–80% of the dust-gas cloud particles settling around the iron ore quarry had a diameter size smaller than 1.4 mkm. Two types of dust-gas clouds were determined: primary and secondary [2,3]. A primary dust-gas cloud originates following dust uptake of the chink mouth. Secondary dust-gas clouds are born due to an additional rock mass crush along the cloud movement after the blast. The cloud shaping process is in effect for 30–45 s after the blast. This process is followed by an intensive period of 60–120 s during which a larger fraction of the dust cloud settles. Gases produced by the blasting process, as well as a large quantity of small particles spread over sizeable distances (up to 10–12000 m) according to wind velocity.

Kharytonov, Gritsan and Anisimova [3] studied the impact of mining dust on the concentration of metals as well as biological activity in the soil (invertase, phosphotase and urease enzymes) at a distance of 0.5–1.5, 3–5, 5–7 and 10–12 km from the mining site to the north of Krivoy Rog. The analysis in part of this data by MVDA was produced with a PLS application (Partial Least Square Projections to Latent Structures) in order to demonstrate interactions between various parameters [4]. The process of birth and formation of dust-gas clouds during the operation of explosives in quarries is a complex, powerfully fusty and flowing process. At birth, a dust-gas cloud (DGC) is an ordinary object, with a high-density space and temperatures, having a large stock kinetic and thermal energy, with insignificant geometrical parameters (up to several hundred $m^3$). At the development stage, a DGC is an object with a polydisperse (dust-gas air) space, with insignificant density and temperatures equal to ambient temperatures. In its final stage of development, a DGC has significant geometrical parameters (from several hundred thousand up to ten million $m^3$). A DGC's powerful potential is a result of the sum of dynamic and thermal potentials produced by the explosive energy. Data from more than twenty videos of mass explosions were taken into account while studying the formation process of a DGC.

DGC development during a mass explosion in an atmospheric setting consists of a sequence of three basic processes:

1. process of DGC birth (time interval 0–560 ms);
2. process of DGC formation (time interval 560–5000 ms);
3. process of DGC distribution in the quarry atmosphere (time interval 5–30 s).

The research on DGC formation during mass explosions in the quarries utilizing experimental videos was useful in defining a temporary period during which it is necessary to take technical measures for active DGC epicenter suppression. This period is defined by the time of formation of thermal and dust epicenters. An important characteristic of the process of DGC development in the atmosphere is its change in height during formation in the thermal and dust epicenter of the cloud.

It was established that in this period the change in DGC height does not depend on meteorological factors.

For the last decade, the annual average emission of industrial dust was 1.25 million tons for the Krivbass District. The amount and the type of emissions in different areas of the Dnepropetrovsk region depend on the type of industrial enterprises involved. In particular, iron-ore mining metallurgy in Krivbass results in high levels of industrial dust, sulphur oxides, carbon and nitrogen. Heavy metal composition content in quarry dust is presented in table 2.

**Table 2.** Heavy metal composition content in quarry dust mg/kg (1NHCL)

| Distance-m | Co | Ni | Pb | Mn | Zn | Cu | Fe | Cd | Cr |
|---|---|---|---|---|---|---|---|---|---|
| 50 | 2.3 | 11.0 | 11.2 | 545 | 13.0 | 4.6 | 9750 | 2.02 | 2.1 |
| 105 | 2.3 | 13.3 | 11.2 | 545 | 9.7 | 4.3 | 11500 | 1.54 | 2.4 |
| 132 | 3.86 | 18.1 | 9.6 | 545 | 7.3 | 6.3 | 2700 | 1.83 | 2.0 |

The heavy metal content is radically different even within one hundred meters from the source of the emission.

The volume of the dust-gas cloud is in the range of 10–20 mln $m^3$ and disperses over distances of 12–15 km [2]. The results of the investigation of the distribution of heavy metals, monitored in two districts (Kryvorozhsky and Shirokovsky) of the Dnepropetrovsk Region where iron ore is mined, is presented in part in Figure 6.

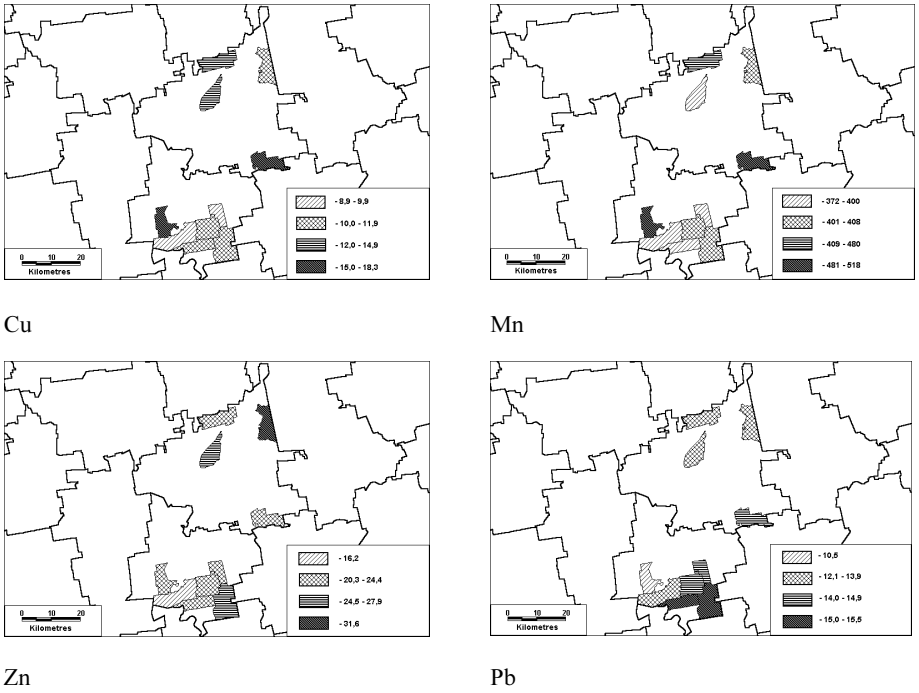



**Figure 6.** Heavy metals spreading with dust clouds around the open-pit mines in the Kryvbass mining region, mg/kg.

There are five iron-ore mining enterprises and 200 open-pit mines within the two mentioned districts. The area of each quarry can be up to 3 $km^2$. It should be noted that each open-pit mine is an emission source affecting residential areas and other quarries. However, different heavy metals obey different spreading patterns.

## 6. Technical Aspects of Open-Pit Mines

The technical aspects of quarry aerology involve the development and practical usage of devices, mechanisms, installations and machines to provide environmentally/health safe open-pit mines [1,5].

This is directly or indirectly related to the choice of dust-gas cloud suppression and quarry ventilation methods.

### *6.1. Dust-Gas Cloud Suppression and Quarry Ventilation Methods*

There are several methods for decreasing and controlling the spread of dust-gas clouds, catching, neutralization, and recycling of harmful substances in quarries:

- well drilling with special chisels in optimum modes;
- dust suppression by water (in the summer), water-emulsion mixtures, aerated solutions, solutions of chemical substances and their mixtures, foam;
- dust suppression by hot water (in the winter), air-hot water mixture, solutions of chemical substances, their mixtures and foam with low congealing and hardening temperatures;
- applications of gravitational and inertial dust catching;
- neutralization of toxic gases in the termdrill torch, etc;
- measures used at the stage of auto transport with diesel engines (replacement of liquid fuel by natural gas; application of engines with smaller toxicity; application of liquid catalysts and ardent neutralizations; supplementary applications to fuel; the combined methods of limiting exhaust gases; conditioners in automobiles cabins, etc.).

Measures at the stage of blasting:

- neutralize supplementations to the blocking and explosive substance;
- covering the explosive block by foam or snow;
- intensive irrigation of the explosive block and its artificial ventilation, application of robotized technical systems of dust recycling in the quarry.

The possibilities of using methods developed for defense applications for air protection and improvements of the working conditions of miners in open-pit mines:

- development of technical systems and devices for active dust suppression based on military techniques (T-55 tank or other);
- design of operative systems to control meteorological and dust-gas conditions in open mines and in the environment by means of chemical investigation.

## 7. Conclusions

The main objective of this paper was to provide a discussion on air/soil monitoring in order to create a basis for designing new approaches for the estimation and mitigation of the blast impact on the spreading of dust-gas clouds in the iron ore-mining region of Ukraine. Physical, heuristic and combined approaches based on separate mathematical interpretation (statistics, numeric integration, etc.) were considered. Approaches to converting military tools into mitigation equipment have also been proposed. One of the possible directions of future research is to investigate the benefits of certain data fusion methods (e.g. remote sensing) to the process of blast impact assessment in the iron-ore region of Ukraine.

## References

[1] Zberovsky, A.V., 1997, Air protection in the quarry – environment – human system. Dnepropetrovsk, p. 136 (in Russian).

[2] Protection from dust in the mining quarries/ Mikhaylov V.A., Beresnevich P.V., Borisov V.G., Loboda A.I., – Moscow, Nedra, 1981, 262 p.

[3] Kharytonov M., Gritsan, N., and Anisimova, L. Environmental problems connected with air pollution in the industrial regions of Ukraine. In: Global atmospheric change and its impact on regional air quality. Barnes, I. (Ed.). *NATO Science Series, IV: Earth and environmental sciences.* Kluwer Academic Publishers, 2002b; 16: 215–222.

[4] Ruth, E., Kharytonov, M., Integrated approach for Assessment of Polluted Areas. In: Social and Environmental Impacts in the North: Methods in evaluation of socio-economic and environmental consequences of mining and energy production in the Arctic and Sub-Arctic. Rasmus Ole Rasmussen and Natalia E. Koroleva (Ed.). *NATO Science Series, IV: Earth and environmental sciences.* Kluwer Academic Publishers, 2003; 31; 57–66.

[5] Kharytonov, M., Zberovsky, A., Drizhenko A., Babiy, A. Blasting impact assessment of the dust-gas clouds spreading in the iron-ore mining region of Ukraine, Data fusion for Situation Monitoring, Incident Alert and Response Management, NATO ASI, August 18–29 2003, Armenia, CD-ROM.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Information and Spatial Data Processing in Risk Management

## *Building a System from the Regional Scale*

Urbano FRA PALEO
*Department of Geography and Spatial Planning,University of Extremadura, Campus universitario. 10071 Caceres. Spain, upaleo@unex.es, +34 927 257000 ext 7722, +34 927 257401 fax*

**Abstract.** This paper examines the process of implementation of an emergency preparedness and response system in Spain and its effect in data and information processing and management. In certain cases, emergency management systems have developed into risk management systems by expanding the number of processes involved. However, for the system to be complete it is also necessary to incorporate mitigation and recovery phases. Emergency management comprises both assessment and policy implementation phases, which differ in data and information analysis and processing. As the phases advance, the weight of data processing declines with the increase of the importance of information. While there have been more developments in data collection and physical simulation models, less attention has been paid to the role of stakeholders and their integration in the decision-making process. As a result, inefficiencies in the communication between agents occur, in turn producing system dysfunction and an increase in the uncertainty of data and information. In order to avoid these problems, it is necessary to define more precisely the rules of integration of system components, and in this regard, system theory offers useful principles. The process of regionalization has conditioned the scale and structure of the national emergency system. Thus, the region becomes the operating domain unless factors of geographical scale, magnitude and class of event divert the event to other administrative levels. The dynamic nature of plans allow effective opportunities for the shift from emergency plans to risk plans. This contributes to an increase in data and information volume, and introduces spatially-based systems. This paper addresses the issues of identifying critical steps in the transition from emergency to risk management systems while studying the prospective directions outlined above.

**Keywords.** Risk management, emergency preparedness and response, spatial information, data processing, GIS

## 1. Introduction

Risk management is a broad and complex field, covering several phases and subphases of hazard and vulnerability assessment and policy implementation. Main flows and transfers of data among agents take place when the assessment component prevails, whereas when implementation is predominant, information transfer among participating agents is more important. However, risk management has not been sufficiently understood as a whole because given that some important components work independ-

ently, there is a need for a continuous adjustment of roles, plans, processes and participation. Emergency preparedness and response, the core subsystem of a risk management system (figure 1), was the first component built; this subsystem is working with more or less success in many countries. The mitigation and recovery phases are only partially incorporated into the system. Inventory, analysis and rehabilitation are, to some extent, linked to the former two phases, but reconstruction and vulnerability reduction are kept as separate policies assigned to different departments, or as missing policies. Therefore the risk management system is not elaborated, complete, and integrated; only an emergent and evolving emergency management system in a pre-mature state has materialized.

Data flows are, in the first place, conditioned by the way information flows within the risk management system and between this and other systems, creating problems of data completeness, redundancy, transparency, access or connectivity. When a component has to make decisions, it does not have access to all the information due to incomplete design or inefficient functioning of the system. The language of communication between components is based on decisions and commands from officials, and opinions and reactions from experts, the public and communities, and its grammar is the set of established protocols.

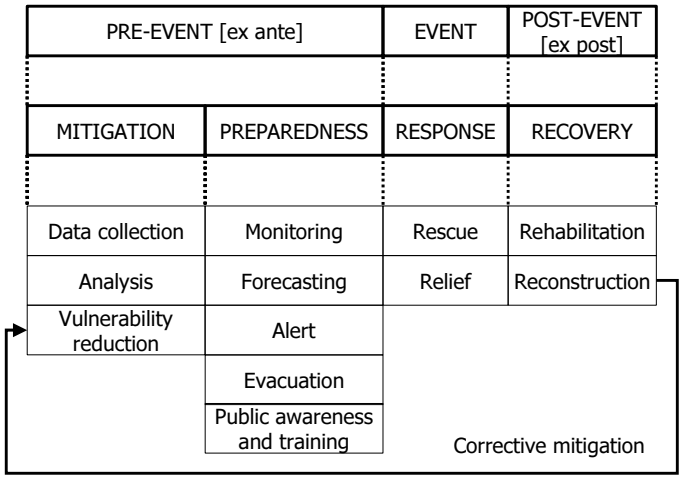



**Figure 1.** Phases in risk management.

The value of information changes along the risk management process. The high data dependency characterizes the phase of mitigation; it declines as the process advances, reaching a minimum in response, and increases again in the phase of recovery. Conversely, the role of information is more important during the response phase when small quantities of data are being processed. In a sense, spatial planning is comparable to mitigation planning since both integrate assessment and policy implementation, and follow processes with an increasing information component and a decreasing data component. The steps of this process are as follows:

- inventory: database construction;
- analysis: study of constituents and their relationships;
- diagnosis: valuation and identification of classes;

– prognosis: the prospect or determination anticipated from/for the course of the case.

Jones [1] identifies the steps and the derived information flow for risk assessment in the field of climate change that, with small changes, may be applicable to general risk assessment, and where the last two steps consist of a relevant policy implementation component:

– identify the key variables in the area;
– create scenarios for the variables;
– assess the relationship between change and impact;
– identify the impact thresholds to be analyzed for risk with stakeholders;
– carry out risk analysis;
– evaluate the risk of exceedance of given impact thresholds;
– analyze proposed adaptations of stakeholders and recommend planned adaptation options.

Identifying the key variables or indicators of hazard and vulnerability and collecting data to build the inventory is an intensive task at the early stages and, if its purpose is enhancing awareness, the inventory must be continuously updated. The process is retroactive since it facilitates the identification of needs, the values of chosen indicators, and the discovery of the public perspective. Risk analysis, a result of the integration of natural hazard and vulnerability assessment based on collected data, examines the probabilities that a given hazardous event will occur and also the expected effects on population, economy and resources. Simonovic and Carson [2] observe that more consideration is commonly given to economic impact and much less to environmental and social impact.

This paper analyses the implementation of the emergency and preparedness system within the Spanish and European framework and its implications in data and information processing and management.

The first part revises the related legislation, assignment and differentiation of responsibilities at the different administrative levels and the formation of decision-making procedures. The characteristics of emergency and preparedness plans as key components in implementation, and their integration into risk management are analyzed. Hazard and risk assessment and decision making are based on the flow of data and information through the processes, adopting various roles along the risk management phases. However, the emergency system is at times inefficient so that their causes and implication in data quality are revised. The construction of thresholds is a key factor in the reduction of dysfunctions. The objective of streamlining the administrative system has led to regionalization, and this process has also become a constituent of the risk management system, contributing to identifying the region as the optimal scale of work.

Finally, a regional emergency system in Spain is presented as a case study for examining the operation and performance of a calling center. Several models are analyzed and classified according to the kind of policies and event class managed. The role of the calling center within the risk management system is examined along with the contribution and structure of a spatially-driven database as the core of the system.

## 2. Policy Implementation

Emergency preparedness and response has remained a centralized system in Spain for some 30 years, but as a result of the progressive administrative regionalization of the country, and the passing of the Civil Protection Law 2/1985, which assigns regional responsibilities, a change to a multi-level system has taken place. It was the first step in structuring the emergency preparedness system, followed by the Regulation on Civil Protection, Decree 407/1992, that defines the objectives and structure of various emergency plans. Consequently, events with interregional coverage, nuclear emergencies, and events of a higher magnitude are the responsibility of the national government. Regions are assigned the responsibility of making special plans for forest fires, floods, seisms and volcanic events; as such they must elaborate a sectoral plan as well as an integrative territorial plan for the entire region (figure 2).

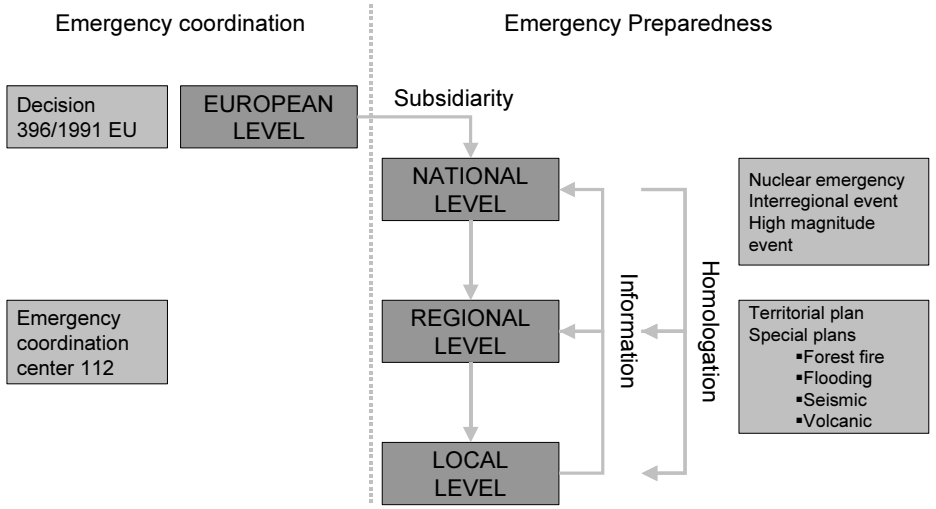



**Figure 2.** Legal framework and transfers in emergency management.

This legislation structures the emergency system by defining (1) components of the system: administrations, plans and classes of events, and (2) relationships within the system: roles and procedures. However, practice has shown that, where information processing is concerned, some dysfunctions arise in this kind of emergency model because some degree of fuzziness is introduced in the response.

Haque [3] studied institutional responsibilities in emergency preparedness in Canada and the Province of Manitoba, with a configuration comparable to that in Spain. The Emergencies Act (1988) and the Emergency Preparedness Act (1988) were aimed at developing an effective civil emergency preparedness. The legislation sought to facilitate cooperation in emergency preparedness and response among administration departments at various administrative levels, to delineate the chain of responsibilities and assign initial action to individuals and corporations. A level – nation, region or local government – intervenes when the emergency clearly lies within its jurisdiction. Higher levels intervene only when requested by the immediate level or when the latter cannot cope or respond effectively; this is usually regulated by a memorandum of un-

derstanding. Haque also observed how most emergencies occur at the regional scale, perhaps due to maximum resource optimization at this scale according to availability and coverage.

The process of differentiating responsibilities implies a definition of intervening variables: (1) event location, a municipality and a region, (2) the nature of the event, and (3) its magnitude (figure 3). Thus, decisions are taken at different levels depending on specific responsibilities. Subsequently, plans elaborated by the regions must be homologated by the national administration, and local plans by the regional authority, involving an upward transfer of information through the levels. However, information is scattered across multiple departments with very little communication, as such it becomes highly dispersed, redundant and loosely managed by each regional subsystem. Integrating information from the 17 regions in the national system is therefore a difficult challenge. Simonovic and Carson [2] suggest the implementation of a model of virtual and distributed database management, in which individual databases may remain in their home agencies provided they can be integrated into the national system.

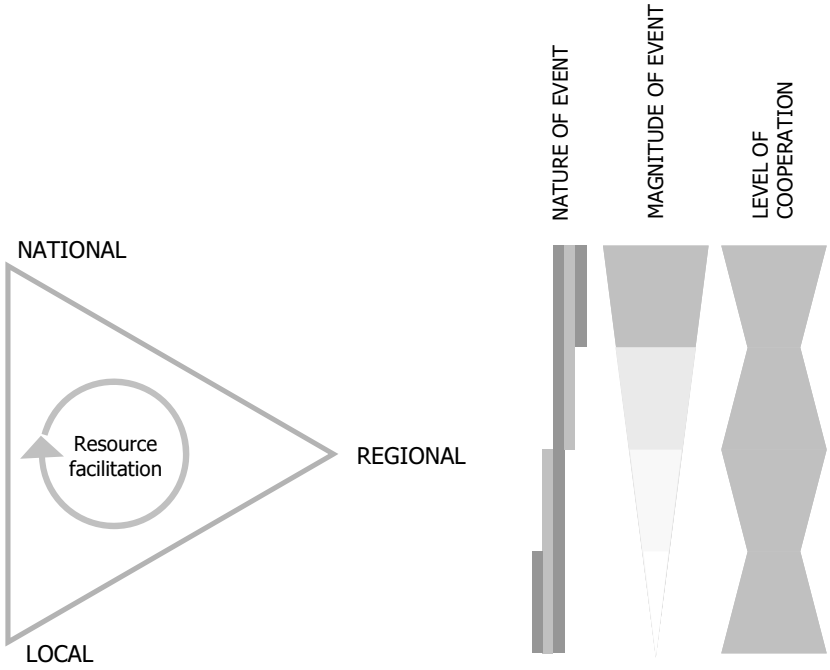



**Figure 3.** Scales of action at the different levels and variables.

Haque [3] studied the effectiveness of legislative and operational measures implemented in the 1997 flood of the Red River Valley, Manitoba. He concluded that mitigation measures and institutional interventions were not fully effective despite the fact that organizational preparedness and mobilization were considerable. Among other factors, there was a lack of communication and understanding between institutions and minimal public involvement in the emergency decision-making process, due to the dominance of the principle of authority over cooperation, co-decision and community involvement in governance. The command-control top-down approach has shown how

conflicts among stakeholders principally arise as a result of communication between experts, administration and the public. This condition produces an urban bias in policy design and implementation, neglecting an operative response in rural areas. Haque [3] suggests the use of more effective means of communicating alerts in order to break the one-way process of assigning people a role, or an interactive exchange of information [4] to give a sense of partnership in managing the situation. Perceptions of risk by stakeholders are diverse; people distrust messages because of the associated uncertainty, lack of credibility, or selective use of information. This calls for [3] a clearer definition of the role, relationships and structure of all participants, and an assessment of the risk associated with communication.

Jones [1] also found that risk assessment and management involves greater formal social participation of stakeholders in risk analysis and the implementation of mitigation measures; their participation ensures that their needs are identified at the early stages. This participation consists of establishing thresholds and an assessment of probabilities of exceedance, which make it possible to introduce risk perception, and ensure that hazard mitigation is promoted and sustained [5].

The process of top-down transfer and assignment of emergency management responsibilities among administrations is based on the principle of subsidiarity (figure 2). This entails a reassignment of duties and the assumption that the regional administration is the appropriate level to manage emergencies. Consequently, there is an event scaling process and an administration is designated to assume the proper functions, although in some cases, a change in event scale implies a change in event class (figure 3). A large flood may affect more than one region or result in a multiple, complex risk, causing side effects like landslides and reservoir collapse. At the lower scale, local governments assume responsibility over the development of local plans to identify vulnerability in the district, to define the structure, available resources, and the early warning system. Nevertheless, this local level implementation exhibits some shortcomings [3], such as lack of completeness, since some local administrations have only basic plans, which are limitedly known or poorly implemented. However, Reddy [5] claims that enforcement at higher levels is more expensive while local authorities can carry out risk management at a relatively low cost, and that there is a link between traditional procedures and new policies.

It is necessary to define more accurately the fundamental principles ruling the relationships among administrations and levels and translate them into protocols, which govern the decision-making process and prevent dysfunctions. Systems theory provides some valuable concepts that can be applied to information flow to define the processes taking place.

- **integration**: parts are components of the system, so that levels are subsystems, and plans at different levels work as a single plan;
- **suboptimization**: optimizing a subsystem independently will not lead, in general, to an optimization of the system and, in some cases, may disrupt the system. Difficulties arise when the interaction between levels evolves to competence, as benefit is conceived not for the optimization of the system but for a particular organizational structure;
- **co-responsibility**: parts pledge to take joint and separate action;
- **coordination**: parts organize themselves in a process of communication from one (several) levels to one (several) levels to act in concert;

- **complementarity**: no level has quantitatively and qualitatively enough resources to efficiently cope with risk management. Parts do not take action simultaneously;
- **redundancy of potential command**: leadership is in the part where the event takes place;
- **subsidiarity**: action is taken at the closest level to where the problem occurs. A part does not take action unless it is more effective than actions taken at the appropriate level, or is within its competence;
- **solidarity**: parts contribute with means and resources to complete those of other administrations. This guarantees that every citizen has the same opportunities.

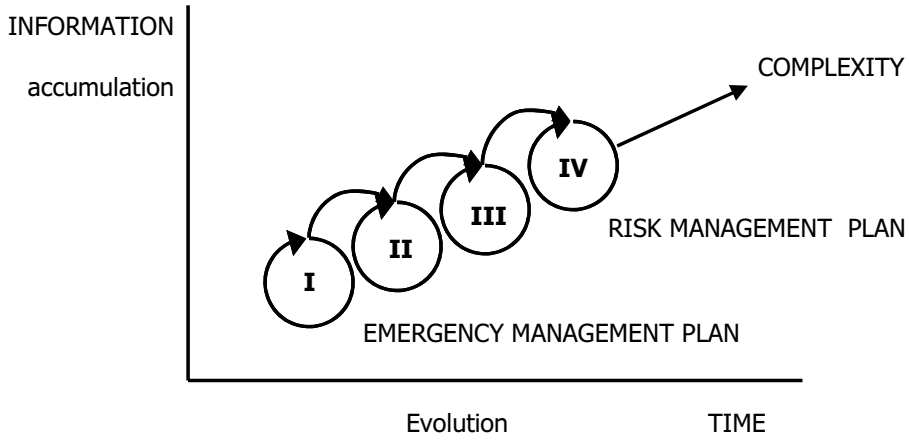



**Figure 4.** Dynamics of emergency plans.

Plans should be dynamic since conditions change with time, particularly those relating to vulnerability and available resources, so that after their lifetime a new plan must be developed (figure 4). Land use changes with new developments, industries, infrastructures and facilities. This implies that the data on which emergency plans were based has changed (figure 5), and plan implementation may fail to accomplish its goal to prevent losses and damages as well as to address specific objectives. Also, plan implementation gives rise to dysfunctionalities or insufficiencies, generating additional information. A new plan must be developed taking into account these factors, while adopting a more comprehensive perspective and increasing in complexity, thus shifting from emergency plans to risk plans.

Moreover, data may be unreliable or insufficient, which introduces uncertainty [6]. King [7] observes some problems, measuring vulnerability with commonly available data. Census data is the most easily accessible, but administrative boundaries must be recognizable and constant so as to enable comparison, and aggregate and derive indicators. Aggregation of data has its drawbacks since details can be lost, resulting in the disappearance of individual differences in the population, so that linkages between population characteristics and property structures are categorically lost. It becomes difficult to identify and locate those sectors of population which are particularly vulnerable, e.g. people from lower socioeconomic backgrounds, the elderly, single parent

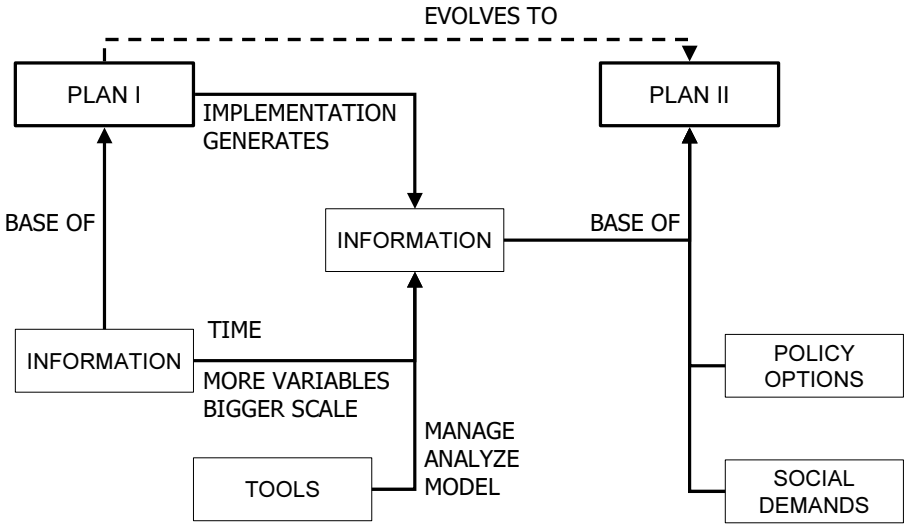



**Figure 5.** Plan dynamics.

families or indigenous people. During events and recovery phases, emergency managers need specific information on people requiring assistance for special intervention. Flynn et al. [8] suggest that, once risk analysis has identified communities at risk, community support for mitigation programs and projects must be sought, particularly when events take place and media attention is still high. These windows of opportunity [8] emerge as processes of communication and interaction with very specific stakeholders. King [7] criticizes the usefulness of Census data, which is considered incomplete, because it only concerns some aspects of the public and the community, and is inconsistent since socioeconomic data decays with time due to mobility and social change.

Jones [1] highlights the effect of uncertainty in climate change projections propagated throughout impact assessment resulting in a limited use of scenarios. Results of modeling tend to be too generic, since they rely on or are measured in terms of thresholds, which are defined as the points at which any minimal modification of a variable may result in a change in state, and may be used in delimiting both the accepted impact of a particular hazard or the resources needed to develop a significant response.

The thresholds used for planning and adaptation options are uncertain, so that decisions are based on incomplete knowledge. Impact thresholds have become a critical component of risk assessment [1], given that risk is evaluated as a function of thresholds.

## 3. Emergency Management

In parallel, the introduction of a centralized emergency telephone number at the European scale has also had implications in the organization of emergency management. The EU Commission Decision 91/396/CEE, on the European System of Emergency Management, selected 112 as the emergency number for all services. The national transposition Decree 903/1997 recognizes emergency management as a regional responsibility.

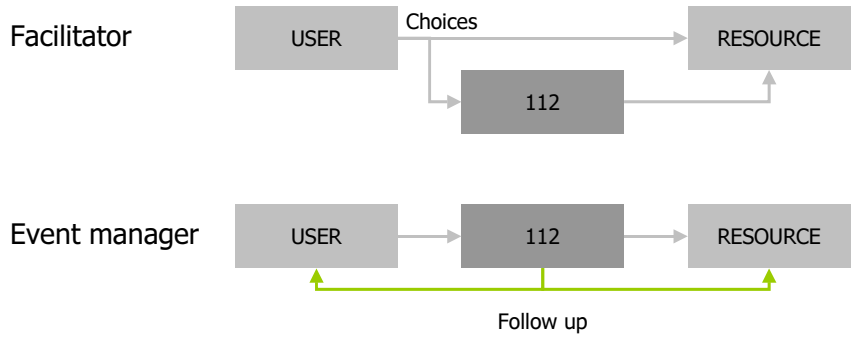



**Figure 6.** Types of emergency management centers according to policy.

The top-down transfer of responsibility has brought a proliferation of emergency management units and a heterogeneous set of perspectives when structuring the regional systems. The models commonly used are based on two criteria: policy and event class. According to the adopted policy we may find two models of an emergency center (figure 6):

- facilitator: 112 transfers the call to the appropriate resource and has no further involvement in the event. The user has the choice of calling directly, thus rendering the system unessential;
- event manager: in this approach the user has the same choice – calling the resource directly or dialing 112 – but subsequent developments in the latter case are not the same, since the manager in the emergency center will take part in the decision-making process. The manager will dispatch resources, check event circumstances, make sure resources are mobilized and get feedback from the user to ensure right return of demand.

Depending on the event class managed we may differentiate among three types of emergency centers (figure 7):

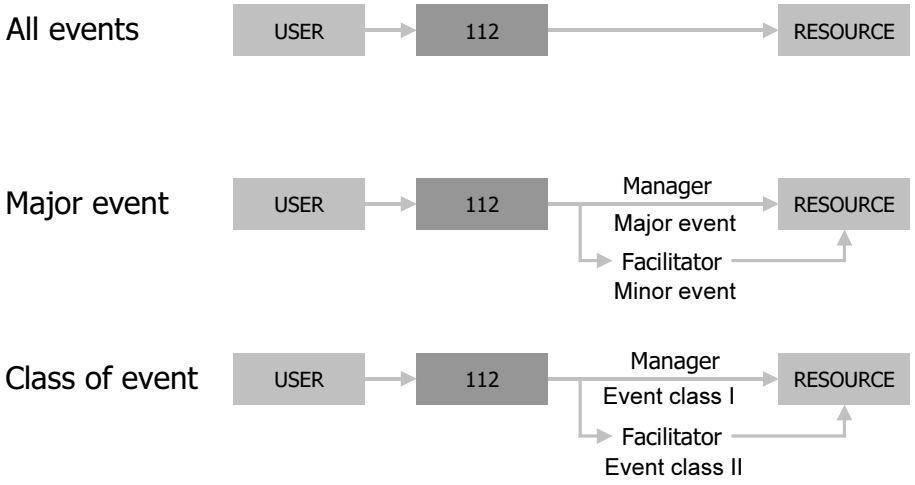



**Figure 7.** Types of emergency management centers according to event class.

- center for the management of extreme events;
- center for the management of emergencies;
- center integrating all/some sectors.

### *3.1. Emergency Management in Extremadura*

The region of Extremadura, located in the central western part of Spain, has adopted the universal phone number 112 following the Decree 137/1998, and has applied a model in which a center for the management of emergencies for every class of events is established. Until July 2002 a formal agreement with the public health system was not established, and its own emergency system – 061 – was integrated in August of the same year. The transfer of the health system from the national government to the region facilitated this process of integration and completion, since in previous years it had not been possible to reach an agreement between national and regional administrations. As a result, five different areas are covered: civil protection, fire extinction, rescue, public security and health emergency. The response entails resource mobilization of four classes and scales: preventive service, programmed service, emergencies and prioritized emergencies. The procedure is based on the adaptive response to emergency events, which depends on the category of the episode, the technical criteria of the officers in charge, and the protocols developed to face each situation (figure 8). The adaptive response system assigns a service operator the task of making decisions under varying circumstances. These decisions are the following: when an emergency call is received, the technician has to disregard those considered to be inappropriate due to various causes, mainly false calls or inappropriate demands. An accepted call then un-

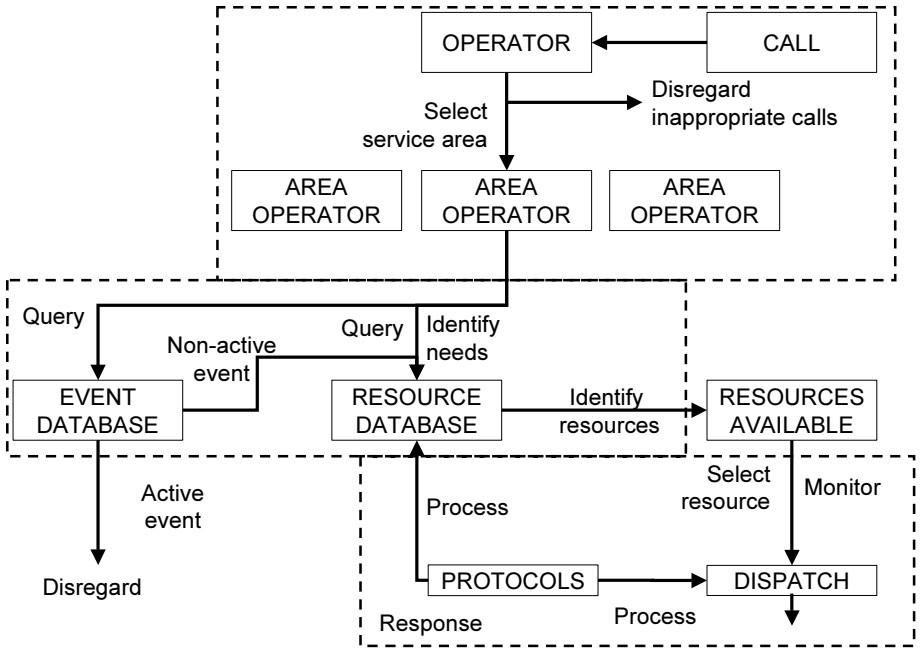



**Figure 8.** Emergency response process.

dergoes a thorough inquiry in order to locate the event and identify the variables on which the forthcoming decisions will be based. Once this is done, the event is transferred to the service area operator. The event database is queried to check whether the event has already been serviced and, if this is not the case, the resource database is queried to search for the nearest appropriate available resources. At this time, event needs and defined protocols are taken into account for decision making (figure 9). Resources are selected and dispatched according to the protocols, since certain resources may belong to different administrations. Finally, according to the adopted model, the operator actively monitors resource deployment.

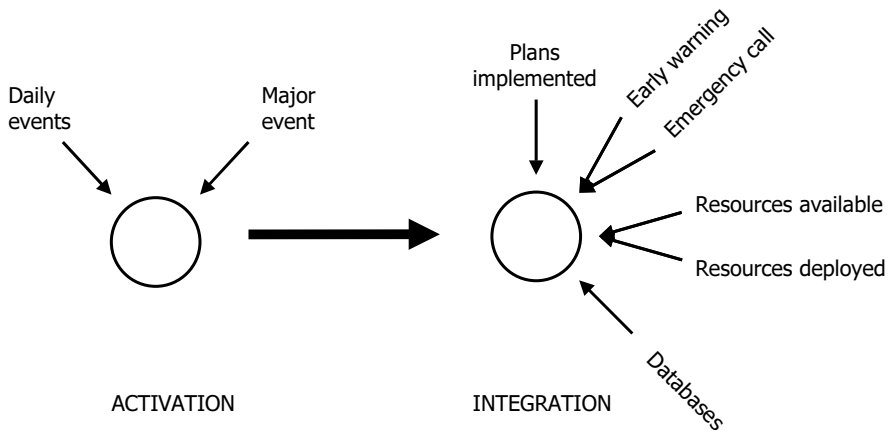



**Figure 9.** Data integration when the system is activated.

Some distinctive events with specific needs are further transferred to and monitored by a specialized operator, who assesses the case, identifies the resources deployed and calculates response times for future improvements. Thus, the basic functions of the call center are:

- registration of calls and their locations by community and/or geographical coordinates;
- classification of events and resource dispatch. A steering questionnaire allows the operator to select a proper response and a proper resource as a function of the event characteristics related to public security: health, protection, rescue and fire extinction;
- event management, which is defined by classes of events: ordinary, extraordinary and resource forecast;
- construction of resource databases;
- construction of the risk analysis database.

The standard technology in use allows for a better adaptation to changes, as compared to the use of technology which requires advanced training, and its modular structure facilitates upgrades based on the same technology. The center receives about 1200 calls a day, and this number increases every year as new resources join the system and demands increase.

### *3.2. From Calling Center to Emergency Preparedness Center*

It is likely that a subsequent adjustment in the system will be the incorporation of emergency plans created to deal with natural and technological hazards in the acting protocols, which require the following courses of action:

- shifting from a management based on non-spatial databases to a management based on spatial databases;
- integrating the volume of information generated by the management process;
- integrating data at the regional scale.

Apart from data management tools, new criteria must be developed to support the decision making process of resource dispatch. This new criteria mainly accounts for spatial parameters, specifically proximity to and connectedness of the events. Until recently, paper maps at a scale of 1:10000 with an overlaid grid of quadrats was used for resource location and dispatch. A geographic information system has now been implemented as a tool for data integration and for the optimization of resource assignment.

Data input is dynamic as information is generated continuously and comes from very diverse sources, which requires importing already existing databases, creating new ones and updating those previously structured. Thus the database management system (DBMS) becomes the core of the information system, although it is not necessarily designed to incorporate a spatial dimension. The GIS supplies the spatial dimension as well as several other capabilities: the organization of information in thematic layers, the spatial-thematic querying, and the analysis and modeling capabilities (figure 10). The following problems arise while integrating spatial information, thus making it a critical and time-consuming process:

- varied data formats;
- geo-referenced information or arbitrary coordinate systems;
- cartographic plane systems for large areas;
- organization in tiles;
- varied accuracy and precision of the different data sources;
- topological structuring of vector data;
- separate thematic and spatial databases;
- combination of vector and raster information. Some categories of information are available only as a raster, like satellite imagery and aerial photography, used commonly in assessment, monitoring, rescue and recovery.

As a result of the implementation of the GIS within the emergency preparedness and response system, the following two approaches are merged into one adaptive spatially-based emergency approach:

- the adaptive response applied to emergent events based on protocols and emergency plans;
- the spatially-based decision making process with a structured geo-referenced database, and the availability of spatial analysis tools supported by a geographic information system.

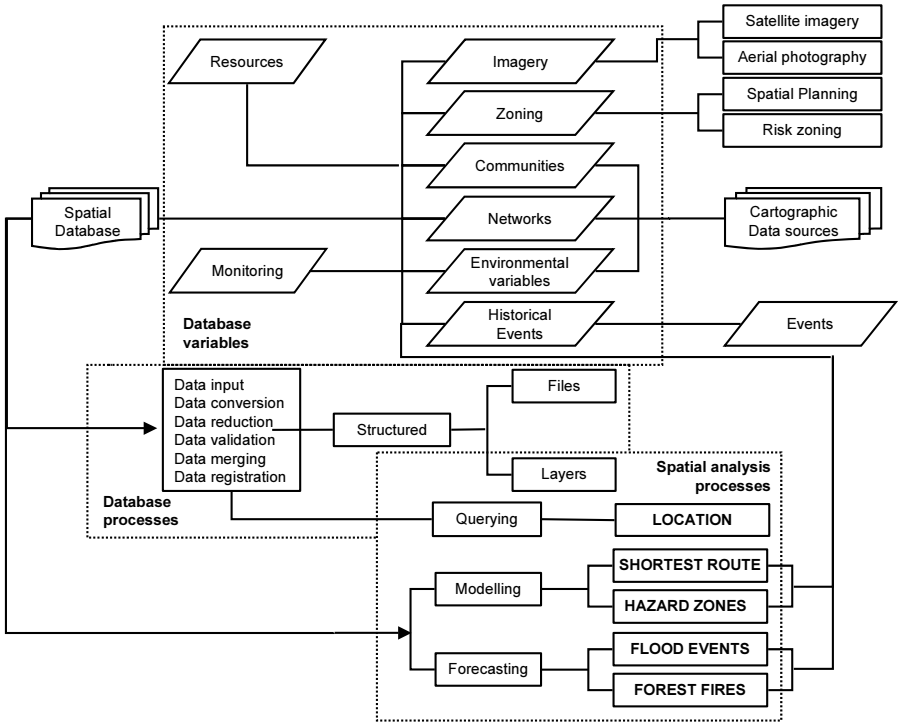



**Figure 10.** Database structure of the spatially-based emergency preparedness system.

## 4. Conclusions

Risk management has not been sufficiently organized and integrated in a system in Spain due to various factors such as the recent implementation of the policy of preparedness and response to hazards, the process of regionalization, and the adoption of an incremental emergency system perspective instead of a risk-based approach. This organization calls for the identification of administrations, departments, agencies and stakeholders that will have a function in risk management, and the relationships among them. These relationships should be ruled by transparent standards based on principles and implemented as protocols. This implies an evolution from emergency plans to risk management plans. The establishment of emergency management centers has induced advancements in this direction. The principles applied to information management and those ruling general systems are applicable to the structure and more efficient organization of the risk management system. The success of plan development and implementation depends on how information is processed by stakeholders, facilitating the flow of information from experts and administration officers, as well as contemplating their perspective of vulnerability in order to minimize the effects of data uncertainty.

The application of geographic information systems in the modeling and support of decision-making requires the use of databases giving rise to the problems of repository location, transferring data from lower to higher levels and, finally, agreement on a model of data distribution. Data processing in GIS encounters difficulties when dealing with uncertainty, lack of completeness, as well as with specific problems of spatial data

analysis. There is a need to standardize data quality, formats and processes, and review the current practice of constructing tiled databases in a national cartographic organization – following traditional mapping practices – that have to be re-processed to compile a customized database when studying a certain area.

## References

[1] Jones, R.N. 2001. An Environmental Risk Assessment/Management Framework for Climate Change Impact Assessments. Natural Hazards 23: 197–230.
[2] Simonovic, S. and Carson, R. 2003. Flooding in the Red River Basin. Lessons from Post Flood Activities. Natural Hazards 28:345–365.
[3] Haque, C.E. 2000. Risk Assessment, Emergency Preparedness and Response to Hazards: The Case of the 1997 Red River Valley Flood, Canada. Natural Hazards 21: 225–245.
[4] Kasperson, R.E. and Stallen, P.M. (eds.): 1991, Communicating Risks to the Public: International Perspectives. Kluwer Academic Publishers, Dordrecht.
[5] Reddy, S.D. 2000. Factors Influencing the Incorporation of Hazard Mitigation During Recovery from Disaster. Natural Hazards, 22:185–201.
[6] Alexander D. 2002. Quo vadis emergency preparedness?. Environmental Hazards 3: 129–131.
[7] King, D. 2001. Uses and Limitations of Socioeconomic Indicators of Community Vulnerability to Natural Hazards: Data and Disasters in Northern Australia. Natural Hazards 24: 147–156.
[8] Flynn, J.P. Slovic, Mertz, C.K. and Carlisle, C. 1999. Public Support for Earthquake Risk Mitigation in Portland, Oregon. Risk Analysis, 2: 205–216.

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Forest Fire Detection by Infrared Data Processing

## *General System and Decision Fusion Algorithms*

Luis VERGARA and Ignacio BOSCH
*Dpto. Comunicaciones, Universidad Politécnica de Valencia, Valencia, Spain*

**Abstract.** We consider the problem of designing an automatic detector for early warning of forest fires, based on the use of infrared cameras or sensors. A general overview of the system is given; we then concentrate on detection algorithms with special emphasis on the fusion of different decisions in order to exploit both the short term persistence and the long term increasing of an uncontrolled fire. Simulated and real data experiments are included to illustrate the algorithms.

**Keywords.** Infrared signal processing, fire detection, subspace matched filter, decision fusion

### 1. Introduction

It is a challenge to increase protection against problems that have an enormous impact on the environment e.g. early forest fire automatic detection. Human surveillance must be complemented or even replaced by automatic systems, preserving on one hand the (in general) reliable human response, but allowing for improved capabilities like area coverage or night surveillance, and all other capabilities that elude limited human performances. Cost-effective systems are required to allow the deployment of many of them to cover very wide areas.

The Signal Processing Group of the Polytechnic University of Valencia (SPAIN) has been working on this topic during the last four years [1–3]. The general system is composed by one or more infrared cameras covering the area under surveillance. Infrared energy in a given pixel of the image is expected to increase in a significant manner when a fire affects the area corresponding to that pixel. The images are locally processed to generate local decisions, which are sent to a base station that integrates decisions from different places. A Geographical Information System (GIS) places the alarms on a map to facilitate the adoption of adequate strategies. Two such systems are currently in a pilot phase in Valencia, where different experiments are demonstrating the viability of the technique.

Although the general system may be considered to be composed of three different parts, namely 1) data acquisition and processing, 2) communication among the local and the base stations, and 3) GIS aspects, data processing, besides being more attractive from a scientific point of view, is by far the method that gives rise to the most difficulties and challenges.

The rate of false alarms must be under strict control in order to insure reliability of the system. But in general, inside the area covered there can be many different sources

of false alarms, including not only background noise, but also occasional effects, such as cars, pedestrians, varying climatic conditions, houses, etc. and any other phenomena that may produce changes in the captured infrared energy. Hence, it is important to design detectors that are able to exploit any particular characteristic of an actual fire that can help to distinguish it from other infrared energy sources. This leads to an interesting application where background noise prediction, fire signature extraction and decision fusion schemes play an important role.

Multiple decisions may come from different sensors covering the same area. However the sensor (typically an infrared camera) is by far the most affecting factor in the final cost of the overall system. Thus, it is not very realistic to assume coverage redundancy. An interesting method to consider is monosensor decision fusion. We could implement different detectors on the same data by exploiting different features of the fire, and make the final decision by integrating all available decisions.

This lecture starts with some general considerations about monosensor decision fusion. We then focus on a detection scheme oriented towards the early warning of forest fires. Some simulations illustrate the proposed scheme. Finally some real data experiments are shown.

## 2. Monosensor Decision Fusion

Monosensor decision fusion implies applying different detectors to the observations which have been collected in a given sensor, to then make a final decision by integrating all available decisions. See figure 1, where vector **x** represents the observations, and $\mathbf{u} = \left[u_1 ....u_L\right]^T$ is the vector of decisions, $u_i = 1$ or $0$. In general each detector works on a vector $\mathbf{z}_i$, obtained after preprocessing the observation vector **x**.

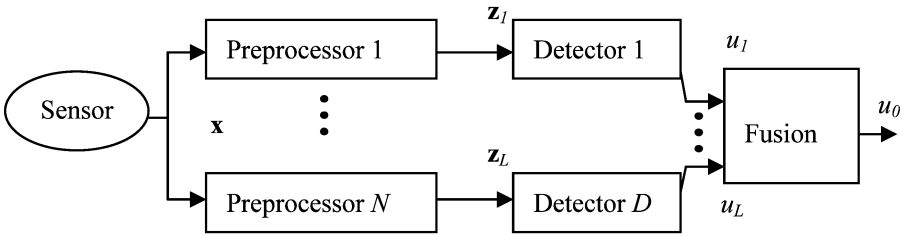



**Figure 1.** Basic scheme for monosensor decision fusion.

The interest of the basic scheme of Figure 1 is that it could be easier to make the design of every detector plus the fusion rule, rather than obtaining an optimum unique detector. Designing an optimum unique detector implies implementing the decision rule

$$R(\mathbf{x}) = \begin{cases} 1 & if \quad L(\mathbf{x}) > \lambda \\ 0 & if \quad L(\mathbf{x}) < \lambda \end{cases} \qquad L(\mathbf{x}) = \frac{f(\mathbf{x}/H_1)}{f(\mathbf{x}/H_0)}, \tag{1}$$

where $f(\mathbf{x} \mid H_j)$ is the probability density function (*pdf*) of the observations, conditioned to the hypothesis $H_j$ ($j$=0 or $j$=1). $L(\mathbf{x})$ is the likelihood ratio and $\lambda$ is a threshold to meet some design constraint. For example if we fit $\lambda$ to obtain a given probability of false alarm $PFA = Probability(R(\mathbf{x})=1|H_0)$, the rule (1) achieves maximum probability of detection $PD = Probability(R(\mathbf{x})=1|H_1)$. This is the Neyman-Pearson criterion.

Finding the likelihood ratio is in general a difficult problem, unless we assume some simplifying models about the observation *pdf* under both hypotheses. However the overall detector could be made from simpler detectors, where simple models are possible, in conjunction with a decision fusion rule to make the final decision. This justifies the use of monosensor decision fusion.

For example, in the classical problem of signal in a noisy background, we could have different types of information about the signal. While integrating all these prior informations in a unique model can be impossible, whereas designing a detector for every type of prior information could be simpler. These ideas have been considered for the specific problem of automatic detection of forest fires.

## 3. Automatic Detector for Early Warning of Forest Fires

First of all let us consider the definition of the observation vector **x** in the context of fire detection. Our problem is to detect an "uncontrolled fire," which is the kind of fire that causes a continuous increase of temperature in a length of time. This type of fire should produce true alarms, unlike any other effects that might generate false alarms. At this point, we consider it adequate to distinguish between false alarms due to occasional effects and false alarms due to background infrared noise. An occasional effect is the one liable to produce a specific pattern in the infrared level evolution, in a given area, when observed for some time. For example, a car crossing the area may produce a significant, but very brief, increase in the infrared level corresponding to that area. On the other hand, background noise refers to a more regular statistical distribution (usually but not necessarily Gaussian). We will refer to false alarms produced by occasional effects as undesired alarms; while we preserve the term false alarms for those produced by background noise. Consequently, we are interested in a system that, for a certain probability of false alarm (*PFA*), can maximize the probability of detection (*PD*) of an uncontrolled fire, while minimizing the probability of detecting undesired alarms (*PDU*).

We assume that the area under surveillance is divided into different cells of spatial resolution (range–azimuth). In a given instant, the sensor or camera will collect an infrared level (sample) to be associated to every given cell. A simple possibility could be to compare the sample level with a predetermined threshold. However, if we consider an isolated sample, we will not be able to differentiate true alarms from undesired alarms: *PD* could be similar, and even lower than *PDU*. Considering that there must be distinctive characteristics on the fire time evolution in a given cell, when compared with the evolution of occasional effects, we can make the detections by using various samples related to the same cell in instants of consecutive scans. We order the consecutive data samples related to each cell in a vector **x** (signature), in which we should try to detect the possible presence of an uncontrolled fire (see figure 2). Hence what follows will refer to the processing of every pixel in the collected infrared image.

We will assume that the noise background is $N\left(0,\sigma^2\mathbf{I}\right)$. Gaussianity guarantees optimality of the different implemented detectors. On the other hand the noise variance $\sigma^2$ must be periodically estimated in a calibration step to normalize the observation vector. In the following we assume that **x** has been already normalized.

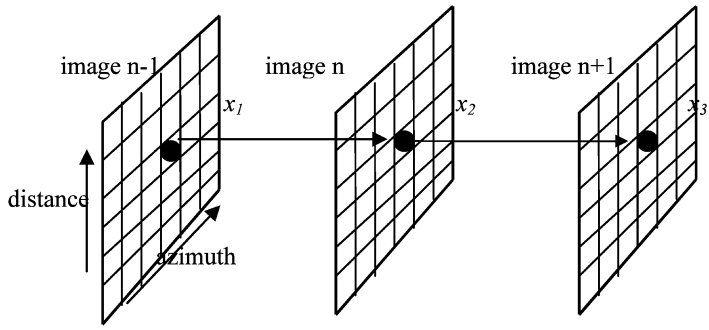



**Figure 2.** Definition of the observation vector **x** (in this case *M*=3).

To define the distinctive features of an uncontrolled fire that will be exploited in designing the detector, it seems convenient to imagine what kind of situation would produce a look out demand in a human vigilante watching the area under protection. There are two main properties that will produce a warning from the human vigilante: short term persistence and long term increasing. Short term persistence is a distinctive feature of something which is not occasional, like a car or any other impulsive phenomena. On the other hand an uncontrolled fire should exhibit an increasing trend if it is really a fire that we should be worried about.

Let us call *T* the time duration available to make a decision, $T = M \cdot T_s$, where *M* is the number of elements of the observation vector **x**, and $T_s$ is the interval between two consecutive images. The value *T* depends on the particular constraints about coverage and/or user delay requirements for early warning.

Short term persistence is implemented by dividing **x** into smaller non-overlapping segments and assuming that the fire signature along each one of these segments is inside a "low-pass" subspace having projection matrix **P**. In agreement with the notation of figure 2 (assuming that no other preprocessing has been done on the observation vector) let us call $\mathbf{z}_1 \ldots . \mathbf{z}_L$ by the *L* segments in which **x** has been divided. Then a matched subspace detector (*MSD*) can be implemented for every segment. The *MSD* compares the energy of the vector $\mathbf{z}_i$ inside the subspace **P**, with a threshold $\lambda$

$$\mathbf{z}_i^{\mathbf{T}}\mathbf{P}\mathbf{z}_i \underset{<}{>} \lambda \tag{2}$$

The threshold can be easily calculated to fit a desired *PFA*, because $\mathbf{z}_i^{\mathbf{T}}\mathbf{P}\mathbf{z}_i$ has a $\chi_p^2$ *pdf* (chi-square *pdf* with *p* degrees of freedom), where *p* is the subspace dimension. In [1,2], the problem of establishing **P** is considered in detail. Basically **P** defines a "low-pass" subspace where the cut-off frequency and the dimension of the subspace

can be established. Occasional effects having large energy outside the band-pass will have low *PDU*, while a significant *PD* of uncontrolled fires may be preserved.

The increase detector can be implemented by looking for increasing trends in the energy vector $\mathbf{z}_E = [E_1 ... E_L]^T$. To this end, vector $\mathbf{z}_E$ is transformed by a difference matrix of order *n*, and then matched to a DC vector $\mathbf{s}_n$

$$\frac{\mathbf{z}_E^T \mathbf{Q}^{(n)T} \left(\mathbf{Q}^{(n)} \mathbf{Q}^{(n)T}\right)^{-1} \mathbf{s}_n}{\left(2p \mathbf{s}_n^T \left(\mathbf{Q}^{(n)} \mathbf{Q}^{(n)T}\right)^{-1} \mathbf{s}_n\right)^{\frac{1}{2}}} \underset{<}{\overset{>}{\cdot}} \lambda_0 \tag{3}$$

where $s_n = \underbrace{[1 ..... 1]}_{L-n}^T$, the difference matrix is defined by

$$\mathbf{Q}^{(n)} = \mathbf{Q}_{L-n+1} \cdots \mathbf{Q}_{L-1} \cdot \mathbf{Q}_L$$

$$\mathbf{Q}_L = \underbrace{\left.\begin{bmatrix} -1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & -1 & 1 & \ldots & \ldots & 0 \\ & & \vdots & & & \\ 0 & 0 & 0 & \ldots & -1 & 1 \end{bmatrix}\right\} L-1}_{L} \tag{4}$$

To understand equation (3), note that it represents a correlation between the transformed vector $\mathbf{Q}^{(n)}\mathbf{z}_E$ and the DC vector $\mathbf{s}_n$. Prewhitening is necessary because the transformed noise in vector $\mathbf{Q}^{(n)}\mathbf{z}_E$ has an autocorrelation matrix $\mathbf{Q}^{(n)} 2p\mathbf{I}\mathbf{Q}^{(n)T} = 2p\mathbf{Q}^{(n)}\mathbf{Q}^{(n)T}$, $2p$ is the variance of a $\chi_p^2$ random variable. On the other hand $\mathbf{Q}^{(n)}\mathbf{z}_E$ may be considered to be multivariate Gaussian because every element in it is obtained by successive differences of random variables. The first difference vector $\mathbf{Q}_L\mathbf{z}_E$ will have marginal *pdf's,* which are the convolution of a $\chi_p^2$ *pdf* with a $-\chi_p^2$ *pdf*, and this will be approximately Gaussian. Successive differences will increment Gaussianity due to the central limit theorem. Thus, we may consider that the statistic in (3) is *N*(0,1), so that *PFA* can be easily met.

In figure 3 we show a detector scheme that exploits persistence and increasing features. Persistence is determined depending on the number of detections generated along the *L* segments $\mathbf{z}_1 ..... \mathbf{z}_L$. If at least *k* out of *L* decisions are positive, the persistence detector decides "1." Then, if the increasing detector also decides "1," an alarm is generated. We devote the next section to the fusion problem: conditions for optimality, and fitting of the *PFA* and *PD*.

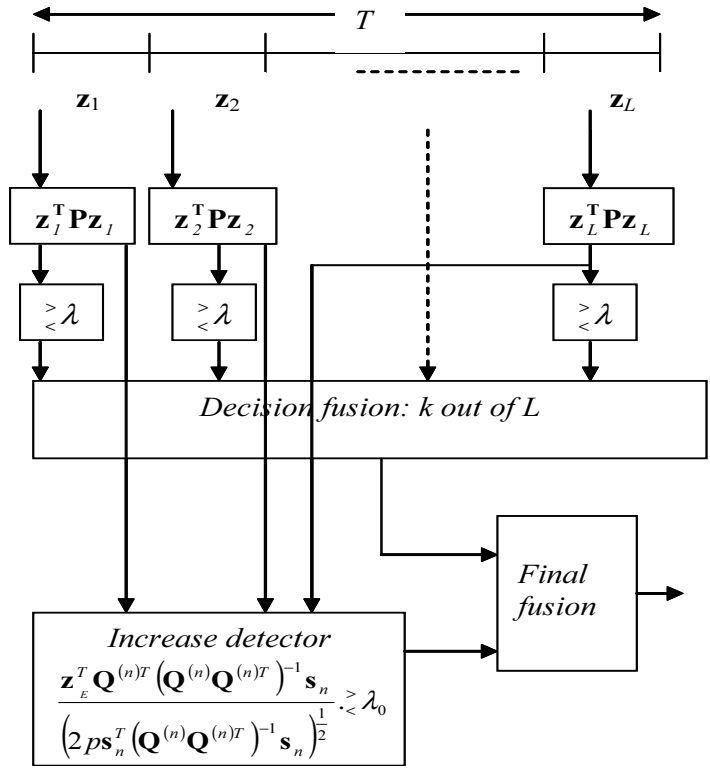



**Figure 3.** Automatic detector scheme.

## 4. Decision Fusion

### *4.1. Persistence Detector*

We look for the optimum decision fusion rule to combine the decisions obtained from the different segments into which we have divided the observation vector **x**.

Let us call $\mathbf{u} = \left[u_1 ... u_L\right]^T$ the vector of decisions. The optimum decision fusion is

$$R_{opt}(\mathbf{u}) = \begin{cases} 1 & if \quad T(\mathbf{u}) > t \\ 1 \text{ with probabilty } \gamma & if \quad T(\mathbf{u}) = t \\ 0 & if \quad T(\mathbf{u}) < t \end{cases} \tag{5}$$

where $T(\mathbf{u}) = \text{Prob}(\mathbf{u} \mid H_1) / \text{Prob}(\mathbf{u} \mid H_0)$ is the likelihood ratio, but as in [3,4], if we assume statistical independence among the decisions

$$T(\mathbf{u}) = \prod_{i=1}^{L} \left( \frac{PD_i}{PFA_i} \right)^{u_i} \left( \frac{1-PD_i}{1-PFA_i} \right)^{1-u_i} \tag{6}$$

where $PD_i$ and $PFA_i$ are the probabilities of detection and false alarm at the *i*-th segment. Normally, both probabilities should be constant $PD_i = PD_0$ and $PFA_i = PFA_0$ for all *i*. Then

$$T(\mathbf{u}) = \prod_{i=1}^{L} \left( \frac{PD_0}{PFA_0} \right)^{u_i} \left( \frac{1-PD_0}{1-PFA_0} \right)^{1-u_i} \tag{7}$$

Now (following the approach of [4]) we must order $T(\mathbf{u})$ for the different values of **u**. In practice $PD_0 > PFA_0$, then

$$\begin{aligned}
& T(\mathbf{u}_1) = T(\mathbf{0}) = \left( \frac{1-PD_0}{1-PFA_0} \right)^{L} \\
& < \underbrace{T(\mathbf{u}_2) = \ldots\ldots = T\left( \mathbf{u}_{\binom{L}{1}+1} \right)}_{\binom{L}{1}\ cases\ having\ 1\ one} = \left( \frac{1-PD_0}{1-PFA_0} \right)^{L-1} \frac{PD_0}{PFA_0} \\
& < \underbrace{T\left( \mathbf{u}_{\binom{L}{1}+2} \right) = \ldots\ldots = T\left( \mathbf{u}_{\binom{L}{1}+\binom{L}{2}+1} \right)}_{\binom{N}{2}\ cases\ having\ 2\ ones} = \left( \frac{1-PD_0}{1-PFA_0} \right)^{L-2} \left( \frac{PD_0}{PFA_0} \right)^{2} \\
& \cdot \\
& \cdot \\
& < T\left( \mathbf{u}_{2^L} \right) = T(\mathbf{0}) = \left( \frac{PD_0}{PFA_0} \right)^{L}
\end{aligned} \tag{8}$$

Now let us assume that we select a threshold $t = \left( \frac{1-PD_0}{1-PFA_0} \right)^{L-nu} \left( \frac{PD_0}{PFA_0} \right)^{nu}$

Taking into account (8), the optimum decision fusion rule (5), can be expressed in the form

$$R_{opt}(\mathbf{u}) = \begin{cases} 1 & \text{if} \quad \text{number of ones in } \mathbf{u} > nu \\ 1 \text{ with probabilty } \gamma & \text{if} \quad \text{number of ones in } \mathbf{u} = nu \\ 0 & \text{if} \quad \text{number of ones in } \mathbf{u} < nu \end{cases} \tag{9}$$

The final PD and *PFA* obtained for the persistence detector are

$$PD_P = \sum_{k=nu+1}^{L} \binom{L}{k} PD_0^k (1-PD_0)^{L-k} + \gamma \binom{L}{nu} PD_0^k (1-PD_0)^{L-k} \tag{10}$$

$$PFA_P = \sum_{k=nu+1}^{L} \binom{L}{k} PFA_0^k (1-PFA_0)^{L-k} + \gamma \binom{L}{nu} PFA_0^k (1-PFA_0)^{L-k} \tag{11}$$

*4.2. Final Decision Fusion*

Let us call $u_P$ to a binary random variable representing the decision given by the persistence detector (*PED*), and $u_I$ to the corresponding binary random decision of the increase detector (*ID*). The proposed decision fusion rule is

$$R(\mathbf{u}) = \begin{cases} 1 & \text{if} \quad \mathbf{u} = [1 \;\; 1] \\ 0 & \text{otherwise} \end{cases} \qquad \mathbf{u} = [u_P \; u_I] \tag{12}$$

The optimum fusion rule is the same expressed in (5), but now with the new two-element vector **u**.

We assume that the *PED* threshold has been fitted to have a certain $PFA_P$, and the *ID* threshold to have a certain $PFA_I$. Let us call $PD_P$ and $PD_I$ the corresponding probabilities of detection. Now we make use of Theorem 1 in [4] to evaluate the likelihood ratio $T(\boldsymbol{u})$ for our particular two-decision fusion case. It is easy to arrive to arrive at the following results

$$T(\mathbf{u}) = \begin{cases} \dfrac{1-PD_P-PD_I+P_1(1,1)}{1-PFA_P-PFA_I+P_0(1,1)} & \mathbf{u} = [0 \;\; 0] \\ \dfrac{PD_P-P_1(1,1)}{PFA_P-P_0(1,1)} & \mathbf{u} = [1 \;\; 0] \\ \dfrac{PD_I-P_1(1,1)}{PFA_I-P_0(1,1)} & \mathbf{u} = [0 \;\; 1] \\ \dfrac{P_1(1,1)}{P_0(1,1)} & \mathbf{u} = [1 \;\; 1] \end{cases} \tag{13}$$

The practical use of equation (13) is very limited because the joint probabilities $P_1(1,1)$ and $P_0(1,1)$ are difficult to evaluate in general, except if statistical independence is assumed between the decision random variables $u_P$ and $u_I$. Let us then assume independence, i.e.,

$$\begin{aligned} P_1(1,1) &= PD_P \cdot PD_I \\ P_0(1,1) &= PFA_P \cdot PFA_I \end{aligned} \tag{14}$$

Substituting (14) in (13) we obtain

$$T(\mathbf{u}) = \begin{cases} \dfrac{(1-PD_P)\cdot(1-PD_I)}{(1-PFA_P)\cdot(1-PFA_I)} & \mathbf{u} = [0\ \ 0] \\ \dfrac{PD_P\cdot(1-PD_I)}{PFA_P\cdot(1-PFA_I)} & \mathbf{u} = [1\ \ 0] \\ \dfrac{(1-PD_P)\cdot PD_I}{(1-PFA_P)\cdot PFA_I} & \mathbf{u} = [0\ \ 1] \\ \dfrac{PD_P\cdot PD_I}{PFA_P\cdot PFA_I} & \mathbf{u} = [1\ \ 1] \end{cases} \tag{15}$$

To complete the optimum test we need to select the threshold *t* and the probability $\gamma$. It should be clear that any pair of values $t-\gamma$ would implement an optimum test, with the only constraint $0 \le \gamma \le 1$. Let us use the notation $T(\boldsymbol{u}) = T_{u_S u_I}$. Noting that in practice $PD_P > PFA_P$ and $PD_I > PFA_I$, we have that

$$T_{00} < \begin{matrix} T_{01} \\ T_{10} \end{matrix} < T_{11} \tag{16}$$

where $T_{01} \le T_{10}$ or $T_{01} \ge T_{10}$ depending on the particular values of $PD_P, PD_I, PFA_P, PFA_I$. For example, if $PFA_P = PFA_I$ and $PD_P \ge PD_I$ then $T_{01} \le T_{10}$, and vice versa. Now we select the pair $t = T_{11}$ $\gamma = 1$, from (32), the optimum decision rule will be

$$R_{opt}(\mathbf{u}) = \begin{cases} 1 & if \quad T(\mathbf{u}) > T_{11} \\ 1 & if \quad T(\mathbf{u}) = T_{11} \\ 0 & if \quad T(\mathbf{u}) < T_{11} \end{cases} \tag{17}$$

and the corresponding *PFA*

$$\begin{aligned} PFA &= P_0\left(T(\mathbf{u}) > T_{11}\right) + 1 \cdot P_0\left(T(\mathbf{u}) = T_{11}\right) = \\ &= 0 + P_0\left(u_S = 1\right) \cdot P_0\left(u_I = 1\right) = PFA_S \cdot PFA_I \end{aligned} \tag{18}$$

because, deducting from (38), $T(\boldsymbol{u})$ can never be greater than $T_{11}$. However, rule (39) is equivalent to deciding "1" when both detectors decide "1," i.e., when $T(\boldsymbol{u}) = T_{11}$ otherwise decide "0." This is the proposed fusion decision rule (12): optimum criterion is implemented when both decisions may be considered statistically independent. It is possible to arrive at the same conclusions by selecting $t = T_{10} \quad \gamma = 0$ if $T_{10} \geq T_{01}$, or $t = T_{01} \quad \gamma = 0$ if $T_{01} \geq T_{10}$.

## 5. A Report on the Warning System

The above detection scheme is the core of a surveillance system which is currently operating in Valencia (Spain). It is based on several automatic lookout posts linked to an Alarm Central Station (ACS) through a communication system.

The lookout posts consist of: thermal sensors, automatic detection unit, and communication subsystem. The sensors are 4 motorized infrared cameras. The lookout posts are strategically set up in order to watch over wide areas. The detection unit is the system kernel and it consists of a thermal image processor subsystem that processes every pixel of the infrared image using the aforementioned algorithms. This reduces false and undesired alarms while maintaining a high detection capability. The ACS receives alarms from the lookout posts; it then places the alarms in a Geographic Information System (GIS) that allows the operator to monitor alarms on a map. We include here some reports about the system performance.

From June 2002 to March 2003, the system was in a pilot phase for correcting and debugging problems of the different subsystems. During this period two real fires were detected at 542 m (camera 1) and 978 m (camera 2), which required the intervention of firefighters. No significant fires passed undetected, but the system still produced too many false or undesired alarms. On March 1, 2003, the system was considered completely debugged and the normal operation phase started. Here we have the corresponding report. In this report, we call false alarms those from an unknown origin and undesired alarms the ones that are due to a known source apart from a real fire.

- period: from March 1 to May 27, 2003;
- number of false alarms: 14;
- number of undesired alarms: 2 (one due to a car and the other one due to an aerostatic balloon);
- number of real fires detected: 3 (two due to barbecues and the other one caused by the burning of wooden figures in the traditional festival of Valencia, called Fallas, in March 19).
- number of undetected real fires: 0

A raw estimate of the obtained *PFA* can be made by the formula

$$PFA = \frac{Number\ of\ false\ alarms}{Number\ of\ total\ processed\ pixels} \tag{19}$$

The image resolution is 320x240 pixels, an image is processed every 2 seconds and there are 4 cameras in operation, we have

$$\begin{aligned} &Number\ of\ total\ processed\ pixels = \\ &Number\ of\ total\ processed\ images \times 320 \times 240 = \\ &= \frac{Effective\ \ number\ of\ \ seconds\ \ \ available\ por\ detection}{2} \times 4 \times 320 \times 240 = \\ &= \frac{Total\ seconds\ in\ one\ day \times number\ of\ \ days \times (UT1 + UT2)}{2} \times 4 \times 320 \times 240 \end{aligned} \tag{20}$$

*UT*1 and *UT*2 are, respectively, the useful percentages of time available for detection in one steady camera and in one scanning camera. In our case *UT*1=110/120 (10s devoted to calibration every 120 s), and *UT*2=45/60 (10 s devoted to calibration every 60s, plus 5 more seconds needed to turn the camera toward a new orientation.

Using (20), we have estimated $PFA = 7.2 \times 10^{-12}$ meanwhile the persistence detector and the increase detector were adjusted to a final $PFA = PFA_P \cdot PFA_I = 10^{-13}$. Consequently, the estimated *PFA* can be considered a good value, taking into account that the span of time is short for reliable estimates of the order of $10^{-13}$. Besides, most of the supposed false alarms will probably be undesired alarms due to occasional effects, which we were unable to determine, rather than to infrared noise background.

Finally, it should be noted that an alarm of any type does not always result in firefighter intervention, since a video surveillance focused on the zone where the alarm appears helps to make the final decision.

Overall, the system is generating an increasing level of user confidence.

## References

[1] Vergara, L., Bernabeu, P., Automatic signal detection applied to fire control by infrared digital signal processing. *Signal Processing*, 80, nº 4, 2000, pp. 659–669.

[2] Vergara, L., Bernabeu, P., Simple approach to nonlinear prediction. *Electronics Letters,* 37, nº 13, 2001, pp. 926–928.

[3] Bernabeu, P.,Vergara, L., Bosch I., A prediction/detection scheme for automatic forest fire surveillance. Submitted to *Digital Signal Processing: A Journal Review.*

[4] Drakopoulos E., Lee C.C., Optimum multisensor fusion of correlated local decisions, *IEEE Trans. On Aerospace and Electronic Systems*, 27, nº 4, 1991, pp. 593–606.

[5] Kam M., Zhu Q., Gray W.S., Optimal data fusion of Correlated Local Decisions, *IEEE Trans. On Aerospace and Electronic Systems*, 28, nº 3, 1992, pp. 916–920.

# APPENDICES

This page intentionally left blank

*Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*
*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*



# Organizing Committee

## *Appendix A*

Elisa Shahbazian
Director Research and Development
R&D Department
Lockheed Martin Canada
6111 Royalmount ave.
Montréal, QC, H4P 1K6
Tel: (514) 340-8343
elisa.shahbazian@lmco.com
CANADA

Pierre Valin
Defence Research and Development
Canada, Valcartier
2459 Pie-XI Blvd.
North Val-Bélair, Quebec G3J 1X5
pierre.valin@drdc-rddc.gc.ca
Tel: 418-844-4000
CANADA

Ashot Akhperjanian
Yerevan Physics Institute
Alikhanian Brothers Street 2, 375036,
Yerevan-36
Tel: 374-1-34-23-74
ashot@jerewan1.yerphi.am
ARMENIA

Galina L. Rogova
Encompass Consulting
9 Country Meadows Drive
Honeoye Falls, NY 14472
Tel: 585-624-1364
rogova@rochester.rr.com
USA

E. Shahbazian et al. (Eds.)
IOS Press, 2005


# Lecturers and Authors

## *Appendix B*

Sarkis Abrilian
Laboratory of Computer Science for Mechanical and Engineering Sciences
LIMSI-CNRS, BP 133, F-91403 Orsay Cedex
Tel.: +33.1.69.85.81.04
sarkis.abrilian@limsi.fr
FRANCE

Kiril Metodiev Alexiev
Central Laboratory for Parallel Processing
Bulgarian Academy of Sciences
Acad. G.Bonchev Str., bl.25 A
1113 Sofia
Tel.: (+359 2) 979 6620
alexiev@bas.bg
BULGARIA

Yannick Allard
R&D Department, Lockheed Martin Canada
6111 Royalmount ave., Montréal, QC, H4P 1K6
Tel.: (514) 340-8310 x-7776
yannick.allard@lmco.com
CANADA

Armen Atoyan
Research Associate
Centre de recherches mathématiques
Université de Montréal
C.P.6128, Succ. Centre-ville
Montréal, QC H3C 3J7
atoyan@crm.umontreal.ca
Tel. (CRM): (514) 343-6111 #4730
CANADA

Andriy Babiy
Institute for Nature Management Problems and Ecology
National Academy of Sciences of Ukraine
6, Moskovskaya Street,
Dnepropetrovsk, 49000
dea@email.dp.ua
UKRAINE

Louise Baril
R&D Department, Lockheed Martin Canada
6111 Royalmount ave., Montréal, QC, H4P 1K6
Tel.: (514) 340-8310 x-8487
louise.baril@lmco.com
CANADA

Niels Ole Bernsen
Natural Interactive Systems Laboratory (NIS)
University of Southern Denmark – Odense
Campusvej 55 DK-5230 Odense M
Tel.: +45 65 50 35 44
nob@nis.sdu.dk
DENMARK

Supratik Bhatacharjee
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721-0104
Tel.: (520) 621-2953
USA

Erik Blasch
Air Force Research Laboratory
2010 Fifth Street
Dayton, Ohio 45433
erik.blasch@wpafb.af.mil
Tel.: 937-255-1115 x3432
USA

David Boily
Dept. of Electrical and Computer Engineering, McGill Univesity
3480 University, Montreal, QC
McConnell Engineering Building 427
dsboily@fastmail.ca
CANADA

Ignacio Bosch
Dpto. Comunicaciones - Universidad Politecnica de Valencia
Camino de Vera, s/n. E-46071 Valencia
Tel.: 0034-9963877111
ibosch@upvnet.upv.es
SPAIN

Éloi Bossé
Defence Research and Development Canada, Valcartier
2459 Pie-XI Blvd.
North Val-Bélair, Quebec G3J 1X5
eloi.bosse@drdc-rddc.gc.ca
CANADA

Richard Breton
Defence Research and Development Canada, Valcartier
2459 Pie-XI Blvd.
North Val-Bélair, Quebec G3J 1X5
richard.breton@drdc-rddc.gc.ca
Tel.: 418-844-4000
CANADA

Herman Bruyninckx
Dept. of Mechanical Engineering, Production Engineering, Machine Design & Automation (PMA)
Katholieke Universiteit Leuven
Afd. PMA, Celestijnenlaan 300B
B-3001 Heverlee
Tel.: (+32)16 322480
Herman.Bruyninckx@mech.kuleuven.ac.be
BELGIUM

Victor S. Chernyak
Moscow Aviation Institute
State Technical University
31-1-12, Volgina ul., Moscow, 117437
v.chernyak@g23.relcom.ru;
chernyak@ieee.org
RUSSIA

Bogdan Constantinescu
Applied Nuclear Physics Department
Institute of Atomic Physics
POB MG-6
76900 Bucharest-Romania
Fax: 401.423.17.01
bconst@sun3vme.nipne.ro
ROMANIA

Paolo Corna
Via Silvio Pellico, 4
20030 Seveso (MI)
paolo.corna@libero.it
ITALY

Andrea Corradini
Natural Interactive Systems Laboratory (NISLab)
University of Southern Denmark – Odense
Campusvej 55, DK-5230 Odense M
Phone: (+45) 65 50 36 98
andrea@nis.sdu.dk
DENMARK

Joris De Schutter
Dept. of Mechanical Engineering, Production Engineering, Machine Design & Automation (PMA)
Katholieke Universiteit Leuven
Afd. PMA, Celestijnenlaan 300B
B-3001 Heverlee
Tel.: +32 16 322479 or +32 16 322480
Joris.DeSchutter@mech.kuleuven.ac.be
BELGIUM

Hugues Demers
R&D Department, Lockheed Martin Canada
6111 Royalmount ave., Montréal, QC, H4P 1K6
Tel.: (514) 340-8310 x-8673
hugues.demers@lmco.com
CANADA

Dany Dionne
Centre for Intelligent Machines
McGill University
3480 University Street
Montreal, H3A 2A7, Canada
ddionne@cim.mcgill.ca
CANADA

Alessandro Farinelli
Dipartimento di Informatica e Sistemisica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma
Tel.: +39-06-4991 8477
farinelli@dis.uniroma1.it
ITALY

Lorella Fatone
Dipartimento di Matematica Pura ed Applicata
Università di Modena e Reggio Emilia,
Via Campi 213/b, 41100 Modena (MO)
fatone.lorella@unimo.it
ITALY

Gian Luca Foresti
Department of Mathematics and Computer Science
University of Udine
via delle Scienze 206, 33100 Udine
foresti@dimi.uniud.it
ITALY

Urbano Fra Paleo
Department of Geography and Spatial Planning
University of Extremadura
Campus universitario. 10071 Caceres
upaleo@unex.es
Tel.: +34 927 257000 x7722
SPAIN

Matteo Gandetto
Biophisycal and Electronic Engineering Department (D.I.B.E.)
University of Genoa
Via Opera Pia 11/a
Genova 16145
Tel.: +39-010-3532674
gandetto@dibe.unige.it
ITALY

Vladimir Gorodetsky
St. Petersburg Institute for Informatics and Automation of the Russsian Academy of Sciences
39, 14th Liniya, St.-Petersburg, 199178, Russia
Tel.: +7-812-2323570
gor@mail.iias.spb.su
RUSSIA

Ion Grama
Universite de Bretagne-Sud,
Laboratoire SABRES
56000 Vannes
ion.grama@univ-ubs.fr
FRANCE

Marco Guainazzo
Biophisycal and Electronic Engineering Department (D.I.B.E.)
University of Genoa
Via Opera Pia 11/a
Genova 16145
Tel.: +39-010-3532189
guainazzo@dibe.unige.it
ITALY

Evgueni Haroutunian
Institute for Informatics and Automation Problems, Armenian National Academy of Sciences
1 Sevak Street
Tel.: + 374 2 281 950
evhar@ipia.sci.am
ARMENIA

Luca Iocchi
Dipartimento di Informatica e Sistemisica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma
Tel.: +39-06-4991 8332
iocchi@dis.uniroma1.it
ITALY

S.S. Iyengar
Robotics Research Laboratory
Department of Computer Science
298, Coates Hall
Louisiana State University
Baton Rouge, LA-70802
Tel.: (225)578-1495
iyengar@bit.csc.lsu.edu
USA

Arsen Javadyan
State Engineering University of Armenia
Computer Software Division
105 Teryan Str.
Yerevan
arsen@arm.hpl.com
ARMENIA

Alexandre Jouan
Defence Research and Development Canada, Valcartier
2459 Pie-XI Blvd.
North Val-Bélair, Quebec G3J 1X5
ajouan@ieee.org;alexandre.jouan@drdc-rddc.gc.ca
Tel.: 418-844-4000
CANADA

Anne-Laure Jousselme
Defence Research and Development Canada, Valcartier
2459 Pie-XI Blvd.
North Val-Bélair, Quebec G3J 1X5
anne-laure.jousselme@drdc-rddc.gc.ca
Tel.: 418-844-4000
CANADA

Oleg Karsaev
St. Petersburg Institute for Informatics and Automation of the Russsian Academy of Sciences
39, 14th Liniya, St.-Petersburg, 199178, Russia
Tel.: +7-812-2323570
ok@mail.iias.spb.su
RUSSIA

Uri Kartoun
Faculty of Engineering
Ben-Gurion University
P.O. Box 653, Be'er-Sheeva 84105
Fax: +972-8-6472958
Tel.: +972-8-6477550
kartoun@bgumail.bgu.ac.il
ISRAEL

Khanik Kerobyan
Institute for Informatics and Automation Problems, Armenian National Academy of Sciences
1 P. Sevak st. 375014 Yerevan
khanik_kerobyan@hotmail.com;
kerobyan@yandex.ru
ARMENIA

Mykola Kharytonov
State Agrarian University
Voroshilov st.25, Dnepropetrovsk, 49027
mykola_kh@yahoo.com
UKRAINE

N. Kostanyan
State Engineering University of Armenia
Terian St. 105,
Yerevan, Armenia, 375009
Tel.: +3741-520-521
ARMENIA

Tine Lefebvre
Dept. of Mechanical Engineering, Production Engineering, Machine Design & Automation (PMA)
Katholieke Universiteit Leuven
Afd. PMA, Celestijnenlaan 300B
B-3001 Heverlee
Tel.: (+32)16 322533 or (+32)16 322480

Tine.Lefebvre@mech.kuleuven.ac.be
BELGIUM

Eric Lefebvre
R&D Department, Lockheed Martin
Canada
6111 Royalmount ave., Montréal,
QC, H4P 1K6
Tel.: (514) 340-8310 x-8715
eric.lefebvre@lmco.com
CANADA

Carlos Lollett
Department of Computer Science
State University of New York at Buffalo
Dept. CSE 136 Bell Hall
Buffalo NY 14260-2000
clollett@eng.buffalo.edu
USA

Minggen Lu
Department of Biostatistics
The University of Iowa
Iowa City
C22 General Hospital
200 Hawkins Drive
Iowa City, IA 52242-1009
Tel.: (319) 384-5016
USA

Mikhail B. Malyutov
Department of Mathematics
Northeastern University
360 Huntington Avenue
Boston, MA 02115
Tel.: (617) 373-5650
MLTV@neu.edu;
mmltv2000@yahoo.com
USA

Luca Marchesotti
Biophisycal and Electronic Engineering
Department (D.I.B.E.)
University of Genoa
Via Opera Pia 11/a
Genova 16145
marchesotti@dibe.unige.it
ITALY

Jean-Claude Martin
Laboratory of Computer Science for
Mechanical and Engineering Sciences
LIMSI-CNRS, BP 133, F-91403
Orsay Cedex
Tel.: (+33) 1 69 85 81 04
Montreuil Computer Science Institute
LINC-IUT, 140 rue de la Nouvelle
France, F-93100 Montreuil
Tel.: (+33) 1 48 70 37 14
martin@limsi.fr
FRANCE

Patrick Maupin
Defence Research and Development
Canada, Valcartier
2459 Pie-XI Blvd.
North Val-Bélair, Quebec G3J 1X5
patrick.maupin@drdc-rddc.gc.ca
Tel.: 418-844-4000
CANADA

Manish Mehta
Natural Interactive Systems Laboratory
(NIS)
University of Southern Denmark –
Odense
Campusvej 55 DK-5230 Odense M
Tel.: +45 65 50 36 97
manish@nis.sdu.dk
DENMARK

Eric Ménard
R&D Department, Lockheed Martin
Canada
6111 Royalmount ave., Montréal,
QC, H4P 1K6
Tel.: (514) 340-8310 x-7544
eric.menard@lmco.com
CANADA

Hannah Michalska
Dept. of Electrical and Computer
Engineering
3480 University, Montreal,
QC, McConnell 514
Tel.: (514) 398-3053
michalsk@cim.mcgill.ca
CANADA

Guy Michaud
R&D Department, Lockheed Martin Canada
6111 Royalmount ave., Montréal,
QC, H4P 1K6
Tel.: (514) 340-8310 x-8798
guy.michaud@lmco.com
CANADA

Lyudmila Mihaylova
Dr. Lyudmila Mihaylova
Department of Electrical and Electronic Engineering
University of Bristol
Merchant Venturers Building
Woodland Road, Bristol BS8 1UB
Tel.: + 44 (0) 117 3315072
mila.mihaylova@ieee.org;
mila.mihaylova@bristol.ac.uk
UK

Maristella Musso
Biophisycal and Electronic Engineering Department (D.I.B.E.)
University of Genoa
Via Opera Pia 11/a
Genova 16145
Tel.: +39-010-3532774
musso@dibe.unige.it
ITALY

Daniele Nardi
Dipartimento di Informatica e Sistemisica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma
nardi@dis.uniroma1.it
ITALY

Vincent Nimier
ONERA Centre de Châtillon
29 avenue de la Division Leclerc
BP 72. 92322 Châtillon
nimier@onera.fr
FRANCE

Stéphane Paradis
Defence Research and Development Canada, Valcartier
2459 Pie-XI Blvd.
North Val-Bélair, Quebec G3J 1X5
stephane.paradis@rdrc-rddc.gc.ca
Tel.: 418-844-4000
CANADA

Jiri Patera
Membre régulier du CRM
Office 4441, Pavillon André Aisenstadt
Université de Montréal
Tel. (CRM): 514-343-6419
patera@CRM.UMontreal.CA
CANADA

Timothy J. Peterson
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721-0104
Tel.: (520) 621-2953
USA

Stefano Piva
Università di Ferrara
Dipartimento di Ingegneria
Via Saragat, 1
44100, Ferrara
spiva@ing.unife.it
Tel.: 0532/293825
ITALY

Edward Pogossian
Academy of Science of Armenia
Institute for Informatics and Automation Problems
24 Marshall Bagramyan Ave.
Yerevan
epogossi@aua.am
Tel.: +374-1-26-31-01
ARMENIA

Carlo S. Regazzoni
Biophisycal and Electronic Engineering Department (D.I.B.E.)
University of Genoa
Via Opera Pia 11/a
Genova 16145

Tel.: +39-010-3532792
carlo@dibe.unige.it
ITALY

Fabio Roli
Department of Electrical and Electronic Engineering
University of Cagliari
Piazza d'Armi, 09123 Cagliari
roli@diee.unica.it
ITALY

Vardan Sahakian
Yerevan Physics Institute
Alikhanian Brothers Street 2,
375036, Yerevan-36
Tel.: 374-1-34-23-74
sahakian@mail.yerphy.am
ARMENIA

Vladimir Samoilov
St. Petersburg Institute for Informatics and Automation of the Russsian Academy of Sciences
39, 14th Liniya, St.-Petersburg,
199178, Russia
Tel.: +7-812-2323570,
Fax: +7-812-3280685,
samovl@mail.iias.spb.su
RUSSIA

Shivakumar Sastry
Department of Electrical and Computer Engineering
University of Akron
Akron OH 44325-3904
ssastry@uakron.edu
USA

Peter Scott
Department of Computer Science
State University of New York at Buffalo
Dept. CSE 136 Bell Hall
Box 602000
Buffalo NY 14260-2000
peter@cse.buffalo.edu
USA

Giuliano Scotti
Sal. Alla chiesa di Fegino 33A
16161 Rivarolo GENOVA
Tel.: 3803220412
ITALY

Armin Shmilovici
Faculty of Engineering
Ben-Gurion University
P.O. Box 653, Be'er-Sheeva 84105
Fax: +972-8-6472958
Tel.: +972-8-6477550
armin@bgumail.bgu.ac.il
ISRAEL

Reetu Singh
Biophisycal and Electronic Engineering Department (D.I.B.E.)
University of Genoa
Via Opera Pia 11/a
Genova 16145
Tel.: +39-010-3532060
reetu@ginevra.dibe.unige.it
ITALY

Alexander Smirnov
Head of Computer Aided Integrated Systems Laboratory
St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
SPIIRAS, 39, 14th Line BO,
St. Petersburg 199178
Tel.: +7 (812) 328-2073 or 328-8071
smir@mail.iias.spb.su
RUSSIA

Lauro Snidaro
Department of Mathematics and Computer Science
University of Udine
via delle Scienze 206, 33100 Udine
snidaro@dimi.uniud.it
ITALY

Alan Steinberg
Space Dynamics Laboratory

Utah State University
1675 North Research, Parkway
North Logan, Utah 84341-1947
asteinberg@sdl.usu.edu
USA

Helman Stern
Faculty of Engineering
Ben-Gurion University
P.O. Box 653, Be'er-Sheeva 84105
Fax: +972-8-6472958
Tel.: +972-8-6477550hel
man@bgumail.bgu.ac.il
ISRAEL

Malur K. Sundareshan
Professor of Electrical and Computer Engineering
Director, Information Processing and Decision Systems Lab,
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721-0104
Tel.: (520) 621-2953
Fax: (520) 626-3144
USA

Alexandre B. Tsybakov
Probability Lab.
Paris Universite-6
175 rue du Chevaleret 75013 Paris
Tél. : +33 1 44 27 53 19
FRANCE

Daniel Turgeon
R&D Department, Lockheed Martin Canada
6111 Royalmount ave., Montréal,
QC, H4P 1K6
Tel.: (514) 340-8310 x-8522
daniel.turgeon@lmco.com
CANADA

Khatchik Vardanyan
Yerevan State University
1 Alex Manoogian Street
375049 Yerevan
Republic of Armenia
Tel.: (374.1)55 4629 (374.1)55 0612
ARMENIA

Pramod K.Varshney
Electrical Engineering and Computer Science Department,
335 Link Hall, Syracuse University
Syracuse, NY 13244
Office: 379 Link Hall
Tel.: (315) 443-4013
Fax: (315) 443-2583, 443-4441
varshney@syr.edu
USA

Luis Vergara
Dpto. Comunicaciones - Universidad Politécnica de Valencia
ETSIT-Camino de Vera s/n, 46023
Valencia
lvergara@dcom.upv.es
SPAIN

Yee Chin Wong
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721-0104
Tel.: (520) 621-2953
USA

Ronald R. Yager
Machine Intelligence Institute
Iona College
New Rochelle, NY 10801
yager@panix.com
USA

Alexandr Zberovsky
National Mining Academy, K. Marx Av. 19, Dnepropetrovsk, 49027
gis@creator.dp.ua
UKRAINE

Francesco Zirilli
Dipartimento di Matematica
"G. Castelnuovo"
Università di Roma "La Sapienza"
Piazzale Aldo Moro 2, 00185 Roma
f.zirilli@caspur.it
ITALY

This page intentionally left blank

*E. Shahbazian et al. (Eds.)*
*IOS Press, 2005*


# Subject Index

active robotic sensing, 129, 131, 141
adaptive, 96, 252, 281, 368, 540, 616, 638, 649, 683
    estimation (see estimation)
    relaxation, 451, 456, 457, 460, 462
    response, 769, 771
agent
    -based systems, 284, 291, 662
    cooperating, 661
    heterogeneous autonomous, 659
    intelligent, 105, 283
    multi-agent
    Oriented Programming, 660
    software, 290, 491, 492, 494, 660
        architecture, 284, 294, 303, 309, 313, 323, 336
        data and information fusion, 336
        information fusion (MAS IF), 308, 309, 312, 319, 323–337
        paradigms, 313
        software tools and platforms, 329
        system (MAS), 104–108, 119, 125–127, 143, 284, 302, 303, 308, 309, 312, 318, 319, 323–337, 659–668, 708
        technology, 312, 326, 669
alignment, 13, 103, 255, 259, 368, 528, 530
algorithm
    Canny, 458
    detection (see detection algorithm)
    EM, 74, 452, 462, 577, 586–588, 596
        generalized (GEM), 575–577, 586
    genetic, 127, 143, 297, 353, 357, 358, 364, 387
    Gerchberg-Papoulis, 454
    iterative restoration, 460, 463
    maximum a posteriori (MAP), 471, 541, 728, 732, 739
    maximum likelihood (ML), 459–461
    minimax, 688, 699, 702, 705
    Projection-Onto-Convex-Sets (POCS), 448, 454–461
allegiance, 92, 94, 555, 556
Ambient (AmI) intelligence, 520–523, 526, 538
ambiguity, 18, 19, 66, 67, 69, 75, 92, 102, 232, 504, 651, 680, 683
architecture
    blackboard-based, 351
    centralized fusion, 662
    communication, 480
    distributed, 662, 663

fusion (see fusion architecture)
hybrid, 25
logical, 480, 488, 489, 492
network, 439, 490, 491
open, 225, 233
parallel, 25, 456
physical, 489
Serial, 25
system (see system architecture)
tree, 257, 260
artificial intelligence, 23, 67, 79, 143, 709
association, 3, 5, 7, 8, 10, 11, 18, 64, 100, 235, 249, 255–257, 259, 261, 319, 341, 349, 354, 355, 358–361, 363, 364, 368, 377, 503, 526, 528, 532, 533, 540, 557, 559, 564–567, 569, 574, 632, 634, 639, 653, 744
autonomous robots, 129
Bayesian (Bayes), 7, 99, 128, 129, 133, 462, 479, 503, 540, 552, 615, 618, 676
classifier, 24, 97, 99, 100, 672, 676
decision rule, 401
distributed detection, 508
estimation, 448
image restoration, 453
inference, 681
naïve classifier, 672, 676
networks, 136, 321
reasoning, 95
risk function, 509
rule, 99, 676
theory, 503
behaviour, 174, 177, 340, 363
knowledge space (BKS), 30, 34
belief
attitude, 64
degree of, 75, 551
function, 21, 70, 75–77, 109
justified (true), 65, 66
propositional, 64
biometric systems, 225, 232
boundary conditions, 174, 175, 717, 718
calculus
incidence, 71, 78, 80
of variations, 171, 172, 176, 180, 188
probability (see probability calculus)
cellular network, 500
Certainty Equivalence Principle (CEP), 599
classification
centralized, 314
distributed (see distributed classification)
gesture, 223–233
meta-, 31, 315–318, 320, 321, 323, 334–337

tree, 315, 317–319, 324, 328, 333
classifier(s)
base, 314, 315, 317–320, 323, 324, 334, 335
Bayesian (see Bayesian classifier)
combination, 34, 394
competence, 320
competent, 323
crisp, 29
decision theoretical (see decision theoretical classifier)
Dempster-Shafer (see Dempster-Shafer)
ensemble of, 30, 394, 403
fusion, 23, 31
hierarchical SAR, 91
image (see image classifiers)
k-nearest neighbours, 25
moment based, 394, 395, 399, 402
multiple, 23–25, 32, 33
system (MCS), 25–27, 32, 34
naive Bayes (see Bayes naïve classifier)
NN classifier, 21, 98
pattern (see pattern classifier)
SAR (Synthetic Aperture Radar), 98
ISM, 99–101
template based (see template based classifier)
closed circuit TV (CCTV), 542
clustering, 60, 107, 256, 342, 353–355, 363, 364, 382, 432, 446, 556
clutter, 237, 239, 242–245, 248, 249, 430, 436, 466, 555, 575, 576, 578, 582, 584–586, 588, 596, 651
cognitive
Agent Development Toolkit, 659, 665
Applied-Work Analysis (ACWA), 215–221
hierarchy, 62, 63
perception, 543
System Engineering (CSE), 63, 205, 206, 208–210, 214, 221, 222
Task Analysis (CTA), 208, 209, 215, 218, 221
Work Analysis (CWA), 17, 209, 214, 215, 220, 222
Work Requirements (CWR), 216, 218, 219
combination rule, 33, 74, 95, 394, 400, 403, 434
Command and Control (C2), 11, 22, 24, 62, 79, 205–207, 209, 210, 215, 221, 341, 343, 347, 367
Command and Control System (CCS), 341, 343, 683
Common Planning and Dynamic Testing, 685, 686
Common Referencing, 259, 262
Communications, 11, 94, 116, 206, 250, 260, 267, 272, 274, 276, 277, 279, 301, 342, 347, 348, 481, 487, 489, 493, 498, 501, 503, 519, 668
architecture (see architecture communication)
bandwidth, 342, 343
capabilities, 493
cooperative mode, 481

delay, 18, 107, 662
electromagnetic (see electromagnetic communications)
load, 662
Manager (CM), 348
network management, 492
network, 265, 367, 481, 492, 493, 501, 627
paths, 347
personalized devices, 480
protocol, 280
technology, 662
complementarity, 23–26, 91, 98, 230, 399, 766
complexity, 3, 18, 24, 26, 106, 108, 141, 143, 144, 149–152, 165, 167, 172, 205, 218, 235, 250, 257, 264, 427, 441, 446, 481, 482, 486, 489, 491, 495, 524, 642, 654, 660, 661, 663, 677, 680, 750, 766
computational, 95, 132, 236, 250, 466, 575
Kolmogorov conditional, 677
confidence, 3, 6, 7, 8, 11, 13, 18, 20, 24, 29, 31, 68, 75, 93, 99, 114, 115, 125, 227, 256, 257, 343, 386, 387, 390, 391, 399, 504, 565, 633, 650, 744, 784
degree of, 43, 76, 81, 82, 633
level, 93, 99, 388, 400–402, 467
conflict, 7, 12, 17, 19, 37, 38, 42, 47, 51–54, 69, 75, 77, 81, 83–85, 91, 94, 95, 97, 127, 206, 360, 740
confusion matrix, 320, 399, 400
consequence prediction, 342
consistency, 52, 54, 331, 337, 343, 345, 372, 575, 609, 666
context(ual), 3, 5, 11–13, 15, 18, 19, 51, 62, 79, 81, 94, 99, 126, 129, 132, 143, 144, 159–161, 182, 186, 210, 212, 219, 223–225, 227, 256, 260, 268, 275, 283, 329, 340, 345, 350, 353, 364, 376, 381, 397, 402, 404, 440, 451, 464, 480, 481, 484, 486, 488, 498, 499, 501–505, 511, 518, 520, 523, 526, 527, 543, 544, 552, 614, 628, 659, 663, 691, 710, 776
aware(ness), 483, 486, 487, 491, 492, 499
constraints, 226
estimation, 341
representation, 481, 482, 489, 492, 493
convergence, 126, 162–164, 172, 173, 404, 405, 409, 413, 417, 424, 453, 455–457, 459, 466, 576, 577, 586–588, 592, 593, 612, 626
correlation, 3, 5, 8, 18, 121, 248, 255, 341, 342, 344, 354, 362, 366–369, 605, 617, 646–648, 653, 720, 730, 731, 778
auto- matrix, 778
Project -, 255
cost function, 16, 144, 149, 150, 152, 154, 155, 357, 599
counteraction, 685–688, 703, 709
credibility, 36, 45, 47–49, 81–89, 765
crisp, 29, 37, 39–41, 50, 51, 55, 56, 58, 76, 471–473, 545, 549
cross section, 382
cross-covariance, 342, 344
cryptography, 673, 679–681, 684
DARPA Agent Markup Language (DAML), 283, 296, 304, 305, 307, 333
DAML+ Ontology Inference Layer (OIL), languages, 333

data
combination, 3, 4, 341
incest, 345, 346, 369
fusion (DF), 3, 4, 7, 16, 20, 48, 49, 61, 65, 88, 91, 104, 129, 142, 171, 176, 180, 186, 187, 205, 234, 256, 257, 261, 285, 313–318, 341–353, 362–365, 371, 375–426, 446, 480, 481, 486, 487, 488, 492, 493, 495, 498, 501, 503, 504, 518–520, 526–530, 627–630, 632, 634–638, 641, 645, 661, 685, 708, 714, 719–722, 729, 733
architecture, 340, 351, 464
Engineering Guidelines, 255, 260, 261, 264
levels, 340
multi-modal platform, 347, 350, 351
multi-sensor (source) data fusion (MSDF), 130, 255, 348–351, 367, 369, 372, 376, 381, 519, 573, 639, 784
nodes, 258, 260
process, 81, 256, 503
quality, 486
system engineering, 257, 260, 261
systems, 255, 260, 263, 264, 320, 366, 708, 729
tree, 256, 259, 260, 262, 264
optical satellite, 171
processing, 11, 79, 129, 214, 235–238, 247–250, 311, 313, 343, 404, 416, 425, 652, 660, 710, 760, 774
database, 6, 15, 91, 105, 237, 238, 296, 314, 323, 330, 333, 365, 396, 402, 461, 522, 552, 630, 636, 692, 708, 761, 762, 764, 770–773
management system (DBMS), 289, 764, 771
decentralised system, 342
decision
aid tool, 92
combination, 308, 314, 316, 317, 320–323, 324, 334, 337, 338, 400
distributed --theory, 503
fusion (see fusion decision)
hard, 342, 344
local, 271, 503, 508, 518, 774, 784
makers, 3, 41, 346, 363
optimization, 32, 400
requirements, 218, 219
rule, 115, 400, 491, 509, 510, 514, 597, 600, 775, 782, 783
soft, 343
structure, 320
support system (DSS), 143, 205, 215, 217, 219, 220
theoretical classifier, 500
trees, 27, 226, 318, 319, 324, 338
decision-making, 4, 7, 10, 11, 20, 23, 41, 129, 141, 206, 207, 209, 210, 217, 218, 266, 273, 760, 762, 764, 765, 768, 772
distributed, 308, 328, 335
dynamic human decision making, 205, 207
Intuitive decision making, 209
deformable templates, 394

defuzzifier, 545, 549
degree of confidence, 43, 76, 82, 633
Dempster, 74, 82, 88, 110, 112, 344, 394, 401, 403, 452, 462, 576, 586
Dempster-Shafer (DS)
    classifier, 97
        reasoning, 101, 102
    theory, 7, 13, 44, 54, 59, 75, 87, 91, 100, 321, 381, 387, 388, 390, 396, 397, 401, 402
    truncated, 103, 344, 352
Desktop-based interfaces, 223
detection
    algorithms, 630, 774
    Bayesian distributed (see Bayesian detection)
    characteristics, 651
    coherent, 514
    decentralized (or distributed), 642
    edge, 188, 220, 458, 463
    envelope, 650
    event, (see event)
    false, 249, 602
    fire, 774, 776
    incident (see incident detection)
    Intrusion, 310
    maneuver, 446, 597, 599, 607, 612, 613
    multi-sensor target (see multi-sensor target detection)
    object (see object detection)
    parallel distributed (see distributed parallel detection)
    performance, 18, 504, 518
    probabilities, 432, 643
    theory, 17, 507, 654
    threat (see threat detection)
deterrence function, 542
development methods, 261
diffraction, 246, 448–450, 477
diffusion coefficient, 175, 177
Dirichlet integral, 174, 176, 179
disaster response, 661
display, 5, 8, 11–13, 20, 62, 219, 220, 222, 232, 236–238, 242, 350, 361, 364, 489, 551, 555
distance function, 400
distributed
    approach, 518, 662, 667
    attacks, 310
    Bayesian -detection (see Bayesian distributed detection)
    classification, 312, 331
    coordinated - attacks, 310
    coordination protocol, 667
    data fusion, 340, 344, 351, 366
    data mining protocol, 334, 335

data mining, 308, 335
decision making (see decision making distributed)
decision-theory (see decision distributive theory)
designers, 336
detection theory, 498, 515
fusion nodes, 351
fusion, 125, 351
identification, 515, 518
knowledge representation, 318
parallel -detection, 508
sensor network (DSN), 265, 266, 268–280, 464, 627–629, 637, 638
smart spaces, 480
system, 11, 105, 273–275, 277, 279, 280, 284, 342, 465, 749
Doppler frequencies, 642, 645
dynamic
environment, 104, 106, 107, 125, 139, 141, 142, 300, 665, 666
expert system, 546, 548
fuzzy expert system, 543, 550, 551
human decision making (see decision making dynamic)
knowledge, 484, 485
model, 542, 543, 546, 551
properties, 484
earthquake, 659, 664, 667
edge-preserving constraints, 452
effectiveness, 6, 27, 33, 236, 251, 259, 467, 561, 635, 659, 665, 668, 669, 686, 727, 764
efficiency, 104, 130, 144, 226, 288, 319, 415, 432, 438, 440, 446, 460, 461, 463, 466, 500, 501, 503, 518, 629, 667, 668, 685, 702, 742
electromagnetic
communications, 481
scattering data, 171, 179, 182, 186–188
signal, 93, 182
waves, 186
Electronic Intelligence (ELINT), 353, 355, 361, 362
emergency, 663, 667, 669, 746, 760–768, 771, 772
management, 760, 763, 765, 767, 769, 772
plans, 663, 760, 763, 766, 771, 772
response, 761, 763
preparedness, 760, 761, 763, 771–773
system, 760, 762, 763, 769, 772
emitter, 92–95, 101
entropy, 4, 18, 52, 60, 132–134, 136, 139–143, 162, 163, 165, 225, 385, 390, 462
analysis, 519
binary, 680
cross, 677
Maximum, 452, 683
environmental conditions, 392, 449, 470
contamination, 740
degradation, 745, 749, 755

protection, 749
stress, 745
threat, 745, 746
ergonomics, 208
estimation
adaptive, 447, 597, 599, 600, 610, 626
Bayesian (see Bayesian estimation)
error, 597, 602, 620
ID, 91, 342, 344
integrated, 597, 599
integrated adaptive, 611
M-, 575, 579, 585, 586
ML-, 577, 579
moving window (kernel), 588
non-parametric (regression), 587, 590
performance, 440, 601, 619
position, 341, 504, 511, 512, 644
set-theoretic (see set-theoretic estimation)
state (see state estimation)
theory, 654
Euclidian distance, 398
equation
diffusion, 175, 177, 753
Fredholm integral, 187
heat, 175, 177
partial differential, 171, 172
event
detection, 542, 627
prediction, 342
evidence (ial), 35, 59, 70, 74, 75, 96, 105, 107, 108, 110, 126, 214, 225, 228, 231, 233, 262, 387–390, 400, 671, 672, 674, 682, 706
reasoning, 91, 100, 381, 388
theory, 71, 76, 104, 108, 126, 670
expert system, 59, 542, 543, 545, 546, 550, 551
false alarms, 235, 249, 538, 685, 686, 774, 776, 783, 784
feature
-based semantic analysis, 543
extract(ion/or), 29, 31, 92, 130, 499, 542, 543
filter(ing), 15, 20, 62, 172, 178, 180, 188, 271, 275, 289, 342, 349, 374, 415, 420, 472, 614, 615
Inverse, 368, 450, 451
matched, 606, 650, 774
nonlinear, 613
Kalman, 7, 13, 18, 23, 133, 138, 249, 366, 368–370, 426–429, 431, 434, 436–439, 445, 446, 556, 560, 601, 604, 605, 607, 611, 617, 619, 621, 631, 632, 639
Inverse Kalman, 366, 369–372, 374, 375
filtration theory, 654
fingerprints, 226
finite differences approximations, 185

Forward Looking Infrared (FLIR), 91, 92, 94, 96, 98, 103–105, 394, 403, 411, 412, 418–424
Fourier, 249, 404–409, 411–413, 415, 417, 419, 424, 425, 430, 462
Functional Abstraction Network (FAN), 216–219
fusion
  application, 309, 313, 327, 330
  architecture, 25, 340–347, 350, 351, 366, 367, 427, 439, 441–443, 446, 448, 464, 465, 467, 474, 478, 638, 662
  audio-video, 225
  contextual, 226
  data (see data fusion)
  decision, 314, 317, 774
  decision-level, 32, 427
  feature level, 224, 662, 666
  higher level fusion, 342, 351, 493, 659
  identification fusion, 95
  image (see image fusion)
  information (see information fusion)
  knowledge (see knowledge fusion)
  knowledge based (KB), 317
  Learning System, 312
  Level 0, 11, 14, 92, 342
  Level 1, 91, 341, 344, 348, 351
  Level 2–4, 9, 11, 15, 26, 92, 207, 342
  Level 3, 9, 11, 15, 26, 92, 342
  Level 4, 8, 11, 16, 26, 342, 348
  macrotemporal, 226
  measurement-level, 29, 31, 394
  meta-model, 324, 328, 331, 333, 334
  network, 527
  performance, 4, 5, 20, 342, 467
  physical level, 488, 494
  pixel/signal level, 11, 342
  process, 3, 8, 10, 17, 81, 83, 86, 88, 89, 185, 256, 258, 349, 353, 366, 368, 388, 392, 470, 627, 633, 642, 661, 662
  rank level, 29, 31, 394
  rule, 28, 29, 31, 493, 775, 779–781
  semantic level, 224, 542
  sensor (see sensor fusion)
  signal level, 225, 498, 576, 661
  symbolic level, 662
  system, 3, 5–7, 9, 11, 12, 16, 19, 20, 65, 66, 260, 264, 309, 312, 342–344, 366, 465, 487, 661, 662, 708, 729
  template-based, 394, 395, 399, 400, 402
  temporal, 226
  tree, 256, 257, 260
  video signal, 642
fuzzy(ness), 59, 66, 70, 76, 763
  belief function, 76

expert system, 542, 543, 545, 547–549
fuzzification, 91–93
fuzzifier, 545
fuzzy inference, 545, 551
logic, 35, 73, 78, 80, 92, 472, 545, 547, 549, 552
membership function, 69, 93, 546
rule, 545, 547, 549
set theory, 59, 76, 79
set, 34, 35, 37, 39, 42, 52, 58, 59, 70, 71, 76–78, 321, 472, 545, 547
system, 60, 545, 549, 550, 551
G-Network, 708, 710, 723, 725–727
game, 223, 227, 228, 598, 599, 603, 612, 741
model, 685–688, 690, 692, 693, 703, 706
Optimal Strategy Provision (OSP), 685
tree, 685–690, 696, 702
Gaussian, 137, 245, 410, 431, 433, 506, 507, 510–512, 576, 588, 599, 606, 618, 776, 778
Additive White - Noise (AWGN) (see noise)
functions, 225, 413, 547
jump-- linear system, 599
noise, 411, 424, 575, 598, 601, 603, 612
process, 576
white -noise (WGN) (see noise)
Gaming Based Dynamic Protection (GBDP), 690–692, 696–701
Geographic Information System (GIS), 237, 387, 389, 663, 664, 760, 771, 772, 774, 783
Global
context representation, 488, 489, 492, 493
MSDF, 349, 367–369, 376
Positioning System (GPS) 164, 500
Resource Management (RM), 263
Tactical Picture (GTP), 348, 349, 366, 367
gradient decent, 173–176, 178
Graphical User Interface (GUI), 219, 223, 227, 236, 238, 239, 251, 659
GUI- Windows, Icons, Menus and Pointers (WIMP), 223
group tracking, 21, 555, 556, 559, 563, 565, 567, 572
IMM Tracking (GRIT), 555, 556, 558, 564, 568, 572, 573
guidance, 258, 449, 597–600, 603, 610–612, 626, 744
Guided Propagation Networks, 230
HALIFAX Class frigates, 340, 343, 346, 350, 351
Heaviside function, 178
heterogeneous data, 302, 312, 481, 520, 536
Hierarchical OS (HOS), 95
Huber, 576, 580, 581, 583–586
Human
cognition, 218, 222
cognitive tasks, 205, 206
Computer Interface (HCI), 12, 224, 226, 487, 489
Computer Interaction, 221, 223, 234

factor, 205, 206, 208, 210–212
Factors Engineers, 219
perception, 129, 220, 224
hypothesis
generation, 256, 258, 259
selection, 256, 258
testing, 189, 192, 201, 613
Identification (ID), 12, 13, 17, 21, 91–102, 189, 192, 195–198, 200, 201, 333, 344, 350, 354, 355, 358, 518, 524, 525, 555, 558, 563–569, 573, 574, 634, 636, 683, 781
estimation (see estimation ID)
mode identification, 498, 500, 503, 521
Identification Freiend or Foe (IFF), 91, 92, 93, 95, 103, 372
ignorance, 19, 65–67, 70, 71, 76, 91, 95, 102, 344, 401, 403
image
classifier(s), 91, 92, 96, 394
compression, 404, 405, 411, 412, 418, 422, 466, 552
degraded, 456, 459
enhancement, 452, 466
exploitation, 449
fusion, 176, 179, 181, 183, 185, 187, 426, 448–451
hyperspectral, 381, 386, 392
infrared (see infrared image)
medical, 17, 452
optical, 171, 181, 183, 185
passive millimeter-wave (PMMW), 460
Positron Emission Tomography (PET), 452
processing, 404, 417, 424, 425, 448, 450, 451, 454, 459, 461, 466, 540, 545
reconstruction, 141, 354, 364, 409, 413, 417, 450, 454, 462, 540, 596, 665, 761
registration, 478
restoration, 448, 450–455, 457, 459, 462, 463
retrieval, 466
satellite, 182
segmentation, 99, 172, 177, 395, 402, 430, 446, 542, 543, 545, 546, 550, 633
SPOT, 182–184, 187
Support Module (ISM), 92, 96, 100, 101
Synthetic Aprture Radar (SAR) 100, 101, 115, 182–184, 381, 382, 387, 461, 477, 479
tactical, 451, 456, 458, 460
thermal, 783
understanding, 542, 543
imaging, 91, 92, 94, 96, 103, 244, 250, 382, 386, 404, 417, 426, 428, 430, 431, 433, 434, 441, 446, 448–450, 453, 454, 458, 462, 464–466, 477, 542, 564, 655
medical, 171, 451, 454
system, 250, 426, 449, 450, 458
impact
assessment, 12, 15, 21, 311, 341, 486, 660, 749, 759, 767
blast impact, 749, 759
incident(s), 186, 187, 544, 597, 599, 600, 602, 603, 740, 744, 745, 747
detection, 448, 464, 626, 740

management, 740
nuclear, 747
radiological, 744, 747
incompleteness, 66, 67, 69
information
a priori information, 7, 86, 95, 98, 106, 237, 387, 389, 452, 454, 458, 464–467, 470, 608
contextual, 3, 15, 17, 274, 381, 392, 480–484, 490–492, 495, 522
dynamic, 483, 484, 664
fusion, 3, 4, 11, 20, 21, 23, 35, 61, 91, 108, 109, 205, 235, 237, 248, 308, 309, 311, 313, 314, 323, 333, 334, 336, 340, 627, 642–644, 653, 654, 659, 660, 665, 669
Design Toolkit, (IFDT), 309, 327, 329, 331–333
fusion centre (IFC), 642, 652, 653
Generalized -Theory, 75
Geographic Information System (GIS), 783
Geographic Information System (GIS), 783
grid, 347
heterogeneous, 107, 302, 312, 480
historical, 545
incomplete, 95, 134
integration, 285, 641, 642, 667
loss, 552, 641, 642
management (IM), 284, 348, 349, 367, 387, 772
modeling, 458, 466
processing, 11, 143, 208, 214, 236, 238, 247, 250, 260, 309, 313, 337, 480, 519, 642, 653, 760, 762, 763
redundant, 224, 342
system, 11, 221, 237, 255, 284, 312, 353, 365, 771, 772
technologies, 709, 730
theory, 70, 200
Value (IV), 102, 352, 464, 467, 470–475, 477, 759
map(s), 470, 471, 473–475, 477
value, 446, 464, 465, 467, 468, 470–475, 477, 478, 484
Information/Relationship Requirements (IRR), 216, 218, 219
Infrared (IR), 394, 396, 449, 466, 627, 630, 631, 633, 635, 774–776, 783, 784
camera, 774, 775, 783
energy, 774
image, 395, 448, 627, 628, 631, 633, 636, 776, 783
noise, 776, 784
sensors (see sensors infrared)
intent, 205–207, 209, 220, 235, 256, 308, 312, 490
Interface, 21, 222, 233, 275, 361, 489, 496, 639
interpolation, 404, 409, 410, 417–420, 422, 424, 472
ISP, 151
JBPDA(F), 555, 559–561, 563, 564, 567, 568, 571, 573
Jonker-Volgenant-Castanon (JVC), 344, 352, 398
knowledge
- based rules, 98, 99

- based system (KBS), 102, 285, 289, 294–296, 303, 306, 313, 317, 318, 328, 347, 377,
acquisition, 284, 305, 665
a-priori, 85, 126, 207, 451, 452, 475, 484, 491, 494
base, 90, 98, 99, 283, 284, 313, 314, 317, 337, 347
expert, 108, 118, 119, 126, 291, 545
fusion, 283, 286, 287, 289, 291, 296, 298, 303
management, 283, 284, 306
mining, 320
non-propositional, 65
prior, 70, 104, 197, 544, 547, 575
representation, 3, 23, 63, 106, 284, 284, 287, 290, 298, 304, 314, 318, 330
space, 320
Knowledge Query and Manipulation Language (KQML), 304, 324
Knowledge Source Network (KSNet), 283, 285, 287, 289, 289–297, 299, 301–303, 306
Landau symbol, 175
Laplace-Stieltjes, 708, 711, 720–722, 737
Laplacian, 176
learning, 16, 23, 25, 34, 74, 104–112, 116–118, 121, 122, 127, 224, 226, 233, 312–314, 319, 320, 322–326, 328, 331, 333, 337, 338, 439, 440, 489, 521, 524, 536
ensemble, 23, 28
inductive, 320
Inferential Theory, 320, 337, 338
meta-, 320, 323
referee, 322
reinforcement, 104, 105, 107, 108, 111, 112, 114, 116, 118, 125–127
least squares, 134, 248, 440, 580, 606
lethality, 94, 206
Lie group, 404–406, 410, 411, 413, 415, 417, 425
linguistic variables, 545, 547, 548
LMS, 576, 578, 583, 585
local
area network, 277, 491
Area Picture (LAP), 348, 367
MSDF, 348, 349, 367, 369, 372, 374, 375
position estimates, 504
logic(s)
classical, 72–74, 78
fuzzy (see fuzzy)
modal, 70, 71, 73, 74, 79
multi-valued, 73, 74
non-monotonic, 71, 73, 74
probabilistic, 78
Least Squares (LS), 576, 583, 585
Maritime Air Area Operations (MAAO), 100
Markov(ian)
Arrival Process (MAP), 732
chain, 201, 614, 677, 715, 716
distributions, 677

networks, 710
model, 226, 577, 676, 677
Random Fields (MRFs), 453
Maximum Likelihood (ML), 191, 677
algorithm, 452, 453, 459–461
blind algorithm, 453
estimate(s), 453, 459, 579, 577, 607, 609
estimation, 225
iterations, 459, 460
restoration, 448
measure
Counter Measures, 95, 100, 649
Electronic Support Measures (ESM), 91, 92, 95, 100–103, 372
mitigation, 764, 765
of performance (see performance measure)
of efficiency, 668
membership functions, 76, 91, 95, 473, 549
Method of Parallel Projections (MOPP), 456, 459, 460
Multiple Validation Gate Model (MVGM), 559, 560
Mining, 673, 747, 749, 750, 754, 756, 757, 759
data, 4, 32, 312, 319, 328, 336
rule, 319, 320
model
auto regressive, 97, 121
Digital Elevation, 183
dynamic (see dynamic model)
game (see game model)
Hidden Markov, 576
JDL (Joint Director of Laboratories), 8, 309, 311, 336, 341, 352, 486
Markov (see Markov model)
meta (see fusion meta-model)
multiple (see multiple model)
parametric, 225
queue model, 708, 729, 733
uncertainty (see uncertainty model)
Modulation
identification, 505
recognition, 499
transfer function (MTF), 458, 466
multi-
frequency electromagnetic scattering, 186
modal associative map, 226
modal behaviors, 230
modal grammar, 225
modal operators, 230
modal user interfaces, 224
scan, 343, 345
sensor tracking, 252, 257, 474, 520, 526, 573, 574, 638

sensor (source) data fusion (MSDF), 130, 255, 348–351, 367, 369, 372, 376, 381, 519, 574, 639, 784
sensor target detection, 257, 478
site Radar Systems (MSRS), 640–643, 647–649, 651–654
trajectory estimation (MTE), 575, 578, 581
multiple
classifiers, 23–25, 27–29, 31, 34, 322, 399, 401
distributed sources, 312
model, 556, 562, 574, 613, 614, 616, 623, 626
site Radar System, 640, 651, 654
target tracking, 235, 555
Validation Gate Model (MVGM), 555, 559–573
Natural Interactive Communication for Edutainment (NICE), 223, 224, 227–229, 232
natural language, 60, 74, 78, 82, 83, 542–545, 551
processing, 229
understanding, 226, 229, 232
naval warfare, 143, 169
network
architecture, 439, 491
-Centric Warfare (NCW), 94, 346, 351
context representation, 488, 489, 493
Manager (NM), 348
node, 709–711, 723, 724, 727
performance, 708, 709
productivity, 709, 709
resources, 709, 710
stability, 726
stationary distribution, 710, 725
topology, 256, 343, 345, 347, 350, 351, 491, 629
Neumann-type initial-boundary value problem, 174–176
neural net(works) (NN), 23–25, 27, 104, 105, 108, 110, 113, 126–128, 226, 233, 387, 394, 426, 427, 430, 432, 434–440, 443, 445, 446, 475, 494, 500, 519, 628, 728
Neyman-Pearson criterion, 776
NM, 350, 353
noise
Additive White Gaussian (AWGN), 506, 507, 510, 514
background, 344, 433, 466, 545, 775, 776
Gaussian (see Gaussian noise)
infrared (see infrared noise)
filtering, 445, 467
variance, 777
white-gaussian- (WGN), 468
nuclear power plant (NPP), 740, 745, 747, 748
numerical approaches, 74, 75, 77, 78
frameworks, 77
object detection, 381, 627
identification, 5, 344, 542, 543, 546, 547
location, 17, 484, 498, 499, 518

Refinement, 91, 341
tracking, 501, 627
Observe-Orient-Decide-Act (OODA), 9, 11, 22, 206, 207, 210, 263, 340, 341
ontology, 12, 19, 283, 284, 287–289, 291–293, 298, 299, 301, 302, 304, 306, 307, 313, 317, 323
shared, 318, 324, 332
open system, 268
operational conditions, 19, 449, 465
operational/tactical picture, 345
Operations Research, 340
optimal(ity), 130, 132, 141, 190, 599, 668, 686, 777, 778
asymptotically optimal test, 189, 200, 201
test, 189, 195, 200
Strategy Provision (OSP), 685,
optimization, 16, 129, 131, 132, 134, 135, 137, 141–143, 149, 171, 172, 176, 181, 183, 185, 187, 188, 249, 251, 262, 352, 400, 403, 418, 424, 440, 479, 484, 549, 551, 764, 765, 771
coverage, 32, 400, 403
statistical, 451–455, 459, 477
unconstrained optimization, 172
orthogonal sum (OS), 95
pattern recognition, 23, 25, 33, 74, 127, 209, 226, 338, 401, 403, 499, 638
classification, 23, 32, 445
classifier, 23, 24, 33
perception, 3, 9, 35, 52, 60, 65, 71, 79, 129, 141, 207, 214, 220–222, 395, 481, 482, 484, 637, 668, 765
machine, 224
Perceptual User Interfaces (PUI), 224, 233
performance
analysis, 260, 395
detection (see detection performance)
estimation (see estimation performance)
evaluation, 660
evaluation, 236, 255, 342, 426, 436
fusion (see fusion performance)
measure, 98, 551, 660
real-time, 102
sensor (see sensor performance)
system (see system performance)
Phase-shift keying (PSK, BPSK, OQPSK, QPSK), 500, 505–508, 510, 514–518
plan, 11, 62, 146, 169, 205, 259–264, 363, 663, 666, 667, 752, 763, 765, 766, 772
action, 262
evaluation, 259
Execution/Resource Control, 259
generation, 259, 264
selection, 259
platform data base (PDB), 92–95, 101
plausibility, 44, 76, 87, 95, 671
Poisson, 452, 453, 459, 576, 614, 676, 711, 712, 728, 730, 733, 738

polarization, 381–384, 390, 649
possibility, 35–43, 49–60, 71, 75–77, 82, 87, 88, 95, 106, 242, 247, 248, 277, 321, 356, 358, 390, 400, 411, 413, 419, 424, 465, 483, 489, 502, 505, 518, 526, 562, 563, 602, 642, 652, 680, 776
    function, 76, 77
    theory, 59, 71, 75, 76, 86, 88, 321
preparedness, 760–764, 771–773
pulse repetition interval (PRI), 466
Presentation Design Concepts (PDC), 216, 219–221
process
    management, 348, 349
    refinement, 8, 11, 311, 348, 349, 660
probabil(ity/ities)(istic), 33, 49, 60, 78, 80, 86, 249, 431, 497, 540, 632
    a priori, 99, 401
    Basic-Assignment (BPA), 95
    calculus, 78
    conditional, 98
    density function, 133, 505, 507, 510, 512, 776
    detection (see detection)
    distribution, 49–54, 134, 135, 189–191, 459, 599, 620
    error, 189, 190, 192, 195, 197, 199, 201
    fuzzy, 75, 76
    joint, 782
    of detection, 249, 503, 514–517, 776, 779–781
        undesired alarms (PDU), 776, 778
    of false alarm (PFA), 538, 620, 776–778, 781, 783
    posterior, 226
    propositional, 78
    theory, 70, 71, 75, 76, 78, 86, 88, 89
    upper, 95
Probabilistic Data Association Filter (PDAF), 431
projection operators, 455
proximity measure, 109, 454, 455
Pulse Repetition
    Frequency (PRF), 94, 355
    interval, 354, 466
Quality of Service (QoS), 708, 729, 730, 732, 738, 739
quantitative analysis, 186
    methods, 465
radar, 5, 91, 94, 99, 100, 183, 184, 235, 237–241, 244–251, 362, 365, 369, 392, 402, 426–429, 433, 434, 438, 439, 441, 444, 464, 466, 526, 556, 573, 574, 639, 645
    bistatic, 642, 646, 649, 654
    Cross Section (RCS), 92, 94, 101, 651
    Doppler, 93
    Electro Optic (EO), 92, 94
    High Frequency Surface Wave (HFSWR), 353, 355, 361–363
    laser (LADAR), 449
    laser, 250, 449
    radar, 640, 643–654

radiometric, 92
Synthetic Aperture (SAR), 91, 92, 94, 96, 98–103, 171, 181–185, 188, 381, 387, 389–392, 448, 449, 451, 461, 462, 477, 479, 655
Range Doppler Profiling (RDP), 96, 652
Resource Description Framework (RDF), 283, 333
reasoning, 5, 7–9, 16, 35, 37, 42, 59, 60, 61, 64, 65, 68, 71–74, 78–80, 83, 91, 95, 98, 106, 130, 161, 200, 262, 348, 351, 387, 390, 484, 510, 750
evidential (see evidential reasoning)
received signal strength (RSS), 501, 512
reconnaissance, 402, 427, 448
recovery, 274, 277, 279, 455, 463, 728, 760, 761, 767, 771
recursive planning, 260, 262
redundancy, 18, 230, 342–345, 464, 465, 629, 681, 761, 766, 775
reference sensing information (RSI), 485, 491
refraction, 186
relaxation parameter, 456, 457
reliability, 18, 32, 62, 81–85, 89, 105, 126, 189, 193–196, 199, 224, 249, 251, 266, 268, 274, 321, 449, 464, 504, 533, 587, 596, 630, 652, 654, 668, 774
remote sensing, 6, 32, 171, 179, 182, 186, 171, 176, 182, 187, 232, 381, 427, 754, 759
rescue, 659, 663–669, 740, 769–771
resolution
angular, 449, 477, 647, 648
enhancement, 448, 450, 451, 460, 461, 467, 477
limitations, 449, 473
spatial, 18, 182, 183, 236, 449, 450, 504, 776
super-resolution, 448, 450–454, 458–463, 467, 477, 478
algorithms, 448, 450, 463, 467, 477
imagery, 448
processing, 461–463, 477
resource, 62, 117, 144, 218, 255, 256, 260–265, 267, 271, 273, 276, 285, 298, 300, 302, 341–344, 348, 490, 494, 573, 574, 659, 687, 692, 709, 729, 731, 733, 734, 764, 768, 769, 770
assignment, 258, 771
management (RM), 11, 92, 94, 143, 151–154, 158, 160–163, 165, 169, 170, 221, 255–264, 348–351, 490
tree, 257, 259, 260, 262
sharing, 729
risk, 15, 214, 218, 260, 263, 271, 344, 509, 516, 743, 746, 760
analysis, 15, 761, 765, 767, 770
assessment, 761, 762, 765, 767
management, 760, 772
Recognised Maritime Picture (RMP), 353, 363
robustness, 6, 100, 104, 130, 225, 250, 271, 308, 426, 464, 465, 484, 526, 597, 604, 667
Software Defined Radio (SDR), 502, 503, 513
semantic(s), 68, 75, 78, 79, 224, 225, 228–230, 232, 284, 285, 290, 486, 521, 523, 542, 552
hypotheses, 229
interpretations, 229

level (see fusion semantic level)
representation, 486, 521
sensor
acoustic, 464
activation, 543
active, 91, 529, 530
fusion, 3, 23, 32, 233, 427, 428, 435, 460, 464, 467, 555, 635
grid, 347
heterogeneous, 235, 250, 480, 481, 483, 522, 539, 630
imaging, 91, 94, 96, 103, 244, 426, 428, 430, 431, 433, 434, 441, 446, 448, 450, 453, 462, 464–466, 477
infrared, 466, 627
intelligent, 92, 236, 268, 495, 520
long-range active, 96
measurement, 662, 464, 467, 468
measurements, 5, 11, 16, 126, 138, 139, 248, 348, 434, 441, 445, 470, 475, 476
mono, 775
multiple, 12, 18, 27, 92, 224, 265, 274, 427, 486, 498, 526, 555, 573, 638
network topology, 342
non-imaging, 91, 441
on-board, 343
optical, 182, 627
performance, 95, 464, 465, 467, 478
phenomenology, 458, 466
PMMW sensor, 449, 451, 462
radio, 491
RF energy, 464
short-range passive sensor, 96
sonar, 94
suite, 343
systems, 235, 250, 255, 444, 445, 490, 491, 504
tactical, 451, 460, 462
thermal, 488, 783
vibration, 464
video, 488, 493, 495, 500, 526, 528, 530, 531, 538, 633
virtual, 481, 492, 627
set theory, 7, 74, 448, 454, 545
random set, 70, 71, 77, 79
rough, 70, 76, 77
set-theoretic
approach, 455
estimation, 451, 452, 454, 459
Shannon, 33, 52, 70, 132, 133, 142, 267, 408, 410, 416, 680, 681
shape
context, 397
descriptor, 396, 397
matching, 397
signal
assessment, 492

processing, 5, 236, 247, 271, 417, 427, 431, 454, 466, 467, 470, 477, 478, 500, 502, 504, 505, 653, 662
-to-interference-plus-noise ratios (SINR), 650, 651
-to-Noise Ratio (SNR), 96, 244, 248, 395, 402, 466, 500, 514, 515, 518
situation
analysis, 21, 61, 63, 72, 79
assessment (SA), 9, 11, 12, 61, 71, 79, 92, 93, 126, 205, 207–209, 257, 309, 311, 313, 336, 341, 348–353, 361, 362, 365, 377, 486, 638, 660
awareness, 9, 62–64, 205, 207, 208, 214, 221, 308, 353
monitoring, 105, 107, 125, 448, 464, 555, 626, 740
refinement, 91, 341
spatial
and spectral constraints, 454
coherence, 641
information, 760, 771
spectrum
microwave, 182
near-infrared, 182
spline, 398, 628, 634
Smart Spaces (SS), 481–483, 487, 490, 491, 493, 496, 498–505, 511, 512, 515, 518
STANAG 2022, 81, 82, 84, 88, 89
state
epistemic, 666
estimation, 11, 141, 236, 237, 250, 255, 259, 308, 344, 426–431, 436, 443, 445, 474, 475, 526, 528, 534, 559, 561, 572, 574, 597–600, 602, 603, 607, 610, 613–620
machine, 225, 278, 329, 330
network, 710, 724, 726
statistical
analysis, 13, 418, 545, 672, 675, 750
approach, 455, 682
independence, 401, 779, 782
Support Vector Machines, 677
surveillance
airborne, 91, 96, 103, 613
aircraft, 100, 355
camera, 542–544, 550
maritime, 353, 364
system, 235, 244, 249, 486, 526, 540, 543, 627, 628, 630, 637, 783
tactical, 448
video, 542, 551
survivability, 11, 251, 342, 343, 345, 449, 652, 654
symbolic, 8, 24, 33, 72, 74, 77, 78, 106, 126, 230, 662
approaches, frameworks, 72, 74, 77
-connectionist technique, 230
Symmetric Functions of Measurements (SFM), 575, 587, 588
system
architecture, 107, 108, 219, 255, 260, 291, 343–346, 445, 464, 471, 628, 662
engineering, 255, 260–262, 264, 309

hybrid, 23
intelligent, 6, 80, 313, 336, 495, 661
performance, 260, 464, 470, 698, 702, 783
TABU, 143, 152, 154, 167, 169, 170
tactical applications, 459, 460, 462
tactical imagery, 451, 456, 458, 460
tactical picture commonality, 343, 345, 346
tactical picture, 248, 342, 343, 345, 346, 351
tactical weapon system, 261
target
associator, 344
Constrained on Road Networks (TCORN), 555, 559, 563, 564, 572, 573
platform, 91
recognition, 6, 21, 104, 126, 381, 446, 460, 641, 648
tracking, 235, 236, 250, 377, 426–434, 465, 474, 526, 555, 560, 563, 569, 573, 626–654
task allocation, 667
template, 249, 259, 394, 395, 397–400, 402, 535, 632
matching, 249, 397, 402, 632
based (see fusion template based)
classifier, 394, 399, 400, 402
terrain, 183, 311, 381, 382, 387, 389, 390, 392, 559, 563, 568, 569, 573, 654, 663
texture, 383, 388, 466
threat
detection, 448
assessment, 92, 93, 348–352, 377
track(ing), 3, 6, 11, 13, 90, 92, 95, 142, 235–237, 239, 244, 245, 248, 249, 257, 264, 272, 343–346, 349, 353–376, 426–451, 462, 464, 465, 467, 468, 474–478, 486, 488, 491, 496, 501, 519, 521, 523–528, 530, 533, 534, 540, 551–573, 597, 608, 612, 614, 626–630, 632, 634, 635, 637, 639, 642, 646, 647, 653, 654, 732, 733
IMM, 250, 252, 344, 352, 368, 369, 555, 556, 558–561, 573, 574, 611, 613, 619–621, 623
Group, 21, 555, 556, 558, 559, 563, 565, 567, 572
management, 343, 345, 346
multi-sensor (see multi-sensor tracking)
object (see object tracking
target (see target tracking)
track maintenance, 427, 467
tracker, 92, 258, 344, 369, 375, 376, 446, 556, 558, 559, 560, 564, 565, 568, 569, 571
tracklet, 366, 367, 369, 372, 375
transition boundary, 471–473
uncertainty, 3, 5, 12, 13, 16, 18, 20, 35–43, 50, 52, 53, 58, 59, 61, 62, 66–77, 79, 84, 95, 104–108, 125, 129–134, 138, 140–142, 205, 206, 209, 213, 221, 224, 227, 258, 260, 263, 284, 316, 343, 354, 369, 372, 374, 375, 390, 401, 462, 465, 475, 477, 597, 601, 603, 604, 627, 680, 742, 744 , 747, 760, 765–767, 772
battlespace, 213
characterization, 61, 72
empirical, 68
inductive, 68

management, 79
models, 321
objective, 69
principles, 75
subjective, 67
uncertainty representation, 61, 71, 105
utility, 15, 17, 21, 108, 115–119, 121, 122, 125, 126, 146, 149–151, 154, 155, 167, 169, 255, 259, 260, 402, 462
vagueness, 66, 67, 69, 75, 76
video
analysis, 520, 522, 526–528, 552
camera, 250, 449, 481, 493, 503, 521, 523
visual
clues, 544, 545
perception, 394, 395, 420, 423
segmentation (VPS), 394, 395
visualization, 64, 220, 222, 235–238, 251, 404, 536, 665, 667
Voting, 320
majority, 25, 26, 29, 395
weighted, 29, 30, 320
vulnerability, 18, 342, 691, 746, 747, 760, 762, 765, 766, 772
wavelet, 475, 500, 519
Weapons Management, 348
weather, 244, 301, 311, 382, 387, 448, 449, 630, 637
wireless LAN (WLAN), 488, 490, 499, 500, 521, 530

# Author Index


Abrilian, S. 223
Akhperjanian, A. 404
Alexiev, K. 235
Allard, Y. 381
Atoyan, A. 404, 417
Babiy, A. 749
Baril, L. 340
Bernsen, N.O. 223
Bhatacharjee, S. 448
Blasch, E. 3, 555
Boily, D. 143
Bosch, I. 774
Bossé, É. 61, 205
Breton, R. 205
Bruyninckx, H. 129
Chernyak, V.S. 640
Constantinescu, B. 740
Corna, P. 171
Corradini, A. 223
De Schutter, J. 129
Demers, H. 394
Dionne, D. 597, 613
Farinelli, A. 659
Fatone, L. 171
Foresti, G.L. 627
Fra Paleo, U. 760
Gandetto, M. 498
Gorodetsky, V. 308
Grama, I. 587
Guainazzo, M. 498
Haroutunian, E. 189
Iocchi, L. 659
Iyengar, S.S. 265
Javadyan, A. 685
Jouan, A. 381
Jousselme, A.-L. 61
Karsaev, O. 308
Kartoun, U. 542
Kerobyan, K. 708, 729
Kharytonov, M. 749
Kostanyan, N. 708
Lefebvre, E. 353
Lefebvre, T. 129
Lollett, C. 104
Lu, M. 575
Malyutov, M.B. 575, 587, 671
Marchesotti, L. 520
Martin, J.-C. 223
Maupin, P. 61
Mehta, M. 223
Ménard, E. 366
Michalska, H. 143, 597, 613
Michaud, G. 340, 366
Mihaylova, L. 129
Musso, M. 498
Nardi, D. 659
Nimier, V. 81
Paradis, S. 205
Patera, J. 404, 417
Peterson, T.J. 464
Piva, S. 480
Pogossian, E. 685
Regazzoni, C. 480, 498, 520
Rogova, G. v, 104
Roli, F. 23
Sahakian, V. 404
Samoilov, V. 308
Sastry, S. 265
Scott, P. 104
Scotti, G. 520
Shahbazian, E. v, 340, 366
Shmilovici, A. 542
Singh, R. 480
Smirnov, A. 283
Snidaro, L. 627
Steinberg, A. 255
Stern, H. 542
Sundareshan, M.K. 426, 448, 464
Tsybakov, A.B. 587
Turgeon, D. 366
Valin, P. v, 91
Vardanyan, K. 708, 729
Varshney, P.K. 498
Vergara, L. 774

Wong, Y.C. 426
Yager, R.R. 35
Zberovsky, A. 749
Zirilli, F. 171