Studies in Computational Intelligence 704

Amir Nakib
El-Ghazali Talbi *Editors*

Metaheuristics for Medicine and Biology

Springer

Studies in Computational Intelligence

Volume 704

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the worldwide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

Amir Nakib · El-Ghazali Talbi
Editors

Metaheuristics for Medicine and Biology

Springer

Editors

Amir Nakib
Laboratoire Images, Signaux et Systèmes Intelligents
Université Paris Est Créteil
Créteil
France

El-Ghazali Talbi
INRIA Lille Nord Europe Parc Scientifique de la Haute Borne
Laboratoire d’Informatique Fondamentale de Lille (UMR CNRS 8022)
Villeneuve d’Ascq
France

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-662-54426-6 ISBN 978-3-662-54428-0 (eBook)
DOI 10.1007/978-3-662-54428-0

Library of Congress Control Number: 2017933440

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer-Verlag GmbH Germany
The registered company address is: Heidelberger Platz 3, 14197 Berlin, Germany

To my parents (Amir Nakib)

Preface

The metaheuristics appeared in the eighties. These global optimization algorithms are stochastic and can be applied to any problem, at the condition it is formulated as a mono-objective or multiobjective optimization problem. They are called nature inspired because their origin comes from the observation of natural behavior: analogy with physics (simulated annealing, microcanonical annealing), with biology (evolutionary algorithms) or with ethology (ant colonies, particle swarms). They also can be extended, particularly to multiobjective optimization. Algorithms, techniques, and methods based on metaheuristic paradigm have been successfully applied to a wide range of complex problems. From the perspective of science development, metaheuristics is an emerging interdisciplinary area between natural sciences, biology, sociology, and computer science. Its rapid growth is a natural product of the rapid development of interdisciplinary research today.

Medical imaging has established itself as a very important research area primarily due to the rapid development of sensors, communication technologies, databases, processors, etc. The phenomenal growth in the technologies and applications for medical imaging has allowed for many interesting results concerning, Medical image acquisition, Medical image processing, and Telemedicine.

This book "Metaheuristics for Medicine and Biology" aims at providing a review for researchers interested in the advances and applications of metaheuristics to biomedical engineering. The book is oriented towards both theoretical and applications aspects of metaheuristics to biomedical imaging.

Paris
December 2016

Amir Nakib
El-Ghazali Talbi

Acknowledgements

The Guest Editor, would like to thank:

- Dr. R. Blanc, co-head of the Department of Interventional Neuroradiology, Fondation A. De Rothschild, Paris, France.
- Pr. P. Decq Head of the Department of Neurosurgery at CHU Henri Mondor, Créteil, France.

Contents

Contributors

M. Ali Department of Paper Technology, Indian Institute of Technology Roorkee, Roorkee, India

Ismahan Baghli Laboratory of GBM, University of Tlemcen, Tlemcen, Algeria

Thibaut Bernard Laboratory of LISSI, Vitry-sur-Seine, France

Amitava Chatterjee Electrical Engineering Department, Jadavpur University, Kolkata, India

S. Corniglion DataTech Institute (DSTI), Paris, France

Narjes Dogaz Laboratory of LISSI, Créteil, France

A.M.T. Gouicem Laboratory of Image and Signal Processing (LISP), Welding and NDT Research Center CSC, Chéraga, Algeria

Atef Hamouda LIPAH, FST, University Tunis El Manar, Tunis, Tunisia

Nurhan Karaboga Sivas Vocational School, Cumhuriyet University, Sivas, Turkey; Department of Electrical and Electronics Engineering, Erciyes University, Kayseri, Turkey

Serdar Kockanat Sivas Vocational School, Cumhuriyet University, Sivas, Turkey; Department of Electrical and Electronics Engineering, Erciyes University, Kayseri, Turkey

Amira Kouzana Laboratory of LISSI, Créteil, France

Madhubanti Maitra Electrical Engineering Department, Jadavpur University, Kolkata, India

Devraj Mandal Electrical Engineering Department, Jadavpur University, Kolkata, India

Sarra Naffakhi Laboratoire LISSI, Université Paris Est, Créteil, France; LIPAH, FST, University Tunis El Manar, Tunis, Tunisia

Amir Nakib Laboratoire LISSI, Université Paris Est, Vitry-sur-Seine, France

M. Pant Department of Paper Technology, Indian Institute of Technology Roorkee, Roorkee, India

P. Siarry Laboratoire LISSI, Université Paris Est, Champs-sur-Marne, France

A. Taleb-Ahmed LAMIH UMR CNRS UVHC 8201, Valenciennes, France

E.-G. Talbi Laboratoire d'Informatique Fondamentale de Lille (UMR CNRS 8022), Villeneuve d'Ascq, France

M. Yahi Laboratory of Image and Signal Processing (LISP), Welding and NDT Research Center CSC, Chéraga, Algeria

Chapter 1
Design of Static Metaheuristics for Medical Image Analysis

Amir Nakib

1.1 Introduction

Medical images, such as Computed Axial Tomography (CAT), Magnetic Resonance Imaging (MRI), Ultrasound, and X-Ray, in standard DICOM (Digital Imaging and Communications in Medicine) formats are often stored in Picture Archiving and Communication Systems (PACS) and linked with other clinical information in clinical management systems. Since 70's research efforts have been devoted to processing and analyzing medical images to extract meaningful information such as volume, shape, motion of organs, to detect abnormalities, and to quantify changes in follow-up studies. Automated image segmentation, which aims at automated extraction of object boundary features, plays a fundamental role in understanding image content for searching and mining in medical image archives. A challenging problem is to segment regions with boundary insufficiencies, i.e., missing edges and/or lack of texture contrast between regions of interest (ROIs) and background. To address this problem, several segmentation approaches have been proposed in the literature, with many of them providing rather promising results.

From the literature algorithms designed medical image segmentation are application dependent, imaging modality and type of body part to be studied. For example, requirements of brain segmentation are different from those of the thorax. The artifacts, which affect the brain image, are different partial volume effect is more prominent in brain while in the thorax region it is motion artifact which is more prominent. Thus while selecting a segmentation algorithm one is required to consider all these aspects. The problems common to both CT and MR medical images are:

- Partial volume effect
- Different artifacts: example motion artifacts, ring artifacts, etc.
- Noise due to sensors and related electronic system.

A. Nakib (✉)
Laboratoire LISSI, Université Paris Est, 122 Rue P. Armangot, 94400 Vitry-sur-Seine, France
e-mail: nakib@u-pec.fr

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_1

It is well known that there is no a standard algorithm for the segmentation of all medical images. Each imaging system has its own specific limitations. For example, in MR imaging (MRI) one has to take care of bias field noise (intensity inhomogeneities in the RF field). It is obvious that some methods are more general as compared to specialized algorithms, and can be applied to a wider range of data. A brief survey of three generations of medical image segmentation techniques can be found in [24].

The motivation of using metaheuristics is to design a new image segmentation techniques, that combine the flexibility of fitness functions with the power of metaheuristics for searching vast search spaces, in order to find the optimal solution. In our work, metaheuristics were improved to solve the continuous and combinatorial optimization problems.

1.2 Image Segmentation as an Optimization Problem

In this section, we show that the segmentation of an image can be reduced to an optimization problem, usually NP-hard [14]. Hence the need to use a metaheuristic. The segmentation of an image I using a homogeneity feature A is usually defined as a partition $P = R_1, R_2, \ldots, R_n$ of I, where:

1. $I = \bigcup R_i, i \in [1, n]$
2. R_i is convex $\forall i \in [1, n]$
3. $A(R_i) = \mathit{True}, \forall [1, n]$
4. $A\ (R_i \cup) = \mathit{False}, \forall i \in [1, n]$ for all connected regions$\big(R_i, R_j\big)$.

One can notice that the uniqueness of the segmentation is not guaranteed by these four conditions. Indeed, the segmentation results depend not only on the information contained in the image, but also on the method used to process these information (method used to take a decision looking to the segmentation result). Generally, to reduce the problem of non-uniqueness of the solution, the segmentation problem is regularized by adding an optimization constraint function F characterizing the quality of a good segmentation. Then, a fifth condition is added to the first four:

5. $F(P^*) = \underset{P \in P_A(I)}{Min} F(P)$ where F is a decreasing function and $P_A(I)$ is the set of all possible partitions of I.

It is obvious that condition 5 does not entirely solve the problem of uniqueness of the segmentation. There are still cases where multiple segmentations can have the same optimal value. This explains the need to implement algorithms based on metaheuristics.

1.3 Metaheuristics' Enhancement for Medical Image Segmentation

1.3.1 Hybrid Ant Colony System (ACS)

In this work, a new formulation of the image thresholding problem as a shortest path problem was proposed. Then, a hybrid ant colony system based algorithm to solve was used to solver it. Moreover, a new segmentation criterion was proposed, named the biased survival exponential entropy. This criterion considers the cumulative distribution of the gray level information and takes into account spatial quality of the segmentation result. It is well known that the ant colony optimization algorithms are slow; in order to solve this problem we enhanced the classical ant colony system by hybridizing it with a local search algorithm. The original Ant Colony System (ACS) [6] was applied in different real world applications. Since the formulation of the problem as a graph optimization problem, ACS can be applied by associating two measures to each arc: the closeness $\tau(i, j)$, and the pheromone trail $\eta(i, j)$. ACS uses a mechanism based on three main operations: (1) the state transition rule provides a direct way to balance between exploration of new edges and exploitation of a priori and accumulated knowledge about the problem. (2) The global updating rule is applied only to edges that belong to the best ant tour. (3) While ants construct a solution, a local pheromone updating rule (local updating rule, for short) is applied.

- *ACS state transition rule*

In ACS the state transition rule is as follows: an ant positioned at the node r chooses the city v to move to by applying the rule given below.

$$v = \begin{cases} ArgMax_{u \in J_k(r)} \left\{ [\tau(r,u)] \cdot [\eta(r,u)]^b \right\} & q \le q_0 \\ s & Otherwise \end{cases} \tag{1.1}$$

where q is a random number uniformly distributed in [0, 1], q_0 is a parameter $0 \le q_0 \le 1$. It determines the relative importance of exploitation versus exploration: when an ant in node r has to choose a node s to move to, it samples a random number $0 \le q \le 1$. b is a parameter that determines the relative importance of pheromones versus distance. s is a random variable selected according to the probability distribution. It is given by:

$$p_k(r,s) = \begin{cases} \frac{[\tau(r,s)][\eta(r,s)]^b}{\sum_{u \in J_k(r)} [\tau(r,s)][\eta(r,s)]^b} & s \in J_k(r) \\ 0 & Otherwise \end{cases} \tag{1.2}$$

$J_k(r)$ is the set of the neighborhood solutions to the current ant r. The state transition rule resulting from Eqs. (1.1) and (1.2) is called *pseudo-random proportional* rule. This state transition rule, as with the previous *random-proportional* rule, favors transitions towards nodes connected by short edges and with a large amount of pheromone. If $q \leq q_0$, then the best edge according to (1.1) is chosen (exploitation), otherwise an edge is chosen according to (1.2) (biased exploration).

- *ACS global updating rule*

In ACS, only the globally best ant is allowed to deposit pheromone. The pheromone level is updated by applying the following global updating rule:

$$\tau(r, s) = (1 - \mu) \cdot \tau(r, s) + \mu \cdot \Delta\tau(r, s) \tag{1.3}$$

where

$$\Delta\tau(r, s) = \begin{cases} \left(L_{gb}\right)^{-1} & \mathit{if}(r, s) \in gb \\ 0 & \mathit{Otherwise} \end{cases} \tag{1.4}$$

$0 < \mu < 1$ is a pheromone decay parameter, and L_{gb} is the length of the globally best tour (gb) from the beginning of the trial.

- *ACS local updating rule*

After having crossed an edge (i, j) during the tour construction, the following local update rule is applied:

$$\tau(i, j) = (1 - \xi) + \xi \cdot \tau_0 \tag{1.5}$$

where $0 < \xi < 1$, and τ_0 are two parameters. The value for τ_0 is suited to be the same as the initial value for the pheromone trails. In order to apply the EACS to solve the segmentation problem, it must be reformulated as a shortest path problem. Then, we define a stochastic rule of local choice of transition to carry out the good path research in this graph. It is also needed to fix the strategy of the deposit and the use of the different traces of pheromone. In our case, the graph is related and balanced. The nodes represent the various possible thresholds (255 thresholds). The weights will be placed on the arcs. The weight on an edge represents the value of the BSEE for the thresholds T bound by this edge (see Fig. 1.1). Then, we define:

- A set of components $C = T$.
- The whole $L = T \times T$, either a total, simple and non controlled interconnection between the objects.
- The function of transition cost $J(i, j) = p_{ij}$.
- The set of the solutions, that correspond to all possible thresholds that allow to segment the image, without violation of the sort constraint.

In order to solve the image segmentation problem, we specify the behavior of the set of the colony guided by the ACS to minimize the segmentation criterion. Initially, N ants are placed randomly on N nodes of the construction graph. Thus, each ant adds, in an incremental way, the threshold that minimizes the total segmentation criterion. Then, every ant moves to its neighborhood using a stochastic transition rule. This rule depends on the quantity of pheromone and heuristic information locally valid. In other terms, an ant having a constructed solution s chooses to move toward a node j according to the rule (1.1). As we consider that the timeliness of

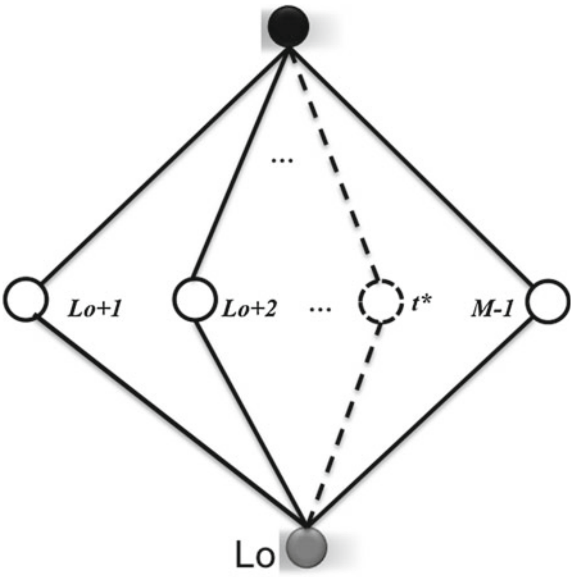


Fig. 1.1 Illustration of the problem formulation where M is the maximum *gray*-level in the image, Lo is the lowest *gray*-level and t^* is an example of an optimal threshold. The *dashed lines* correspond to the shortest path

a solution depends on all the solutions (segmentation thresholds) already found, the heuristic information and the pheromone represent a total relation between the current solution and all the found solutions from the start S (all visited nodes).

$$\tau_S(j) = \sum\nolimits_{i \in S} \tau(i, j) \tag{1.6}$$

The heuristic information corresponds to the value of the segmentation criterion. Then, the candidate is more desirable with the decrease of its fitness function.

The proposed algorithm (Fig. 1.2) consists in applying the proposed EACS (hybridization of ACS with Tabu Search (TS)). The different steps of the proposed segmentation algorithm are presented in the algorithm in Fig. 1.2. Indeed, the EACS principle consists in applying a local search based on TS for 30% of the ant colony. The aim of this local search is to accelerate the convergence of the algorithm and to avoid the local optima. Indeed, the tabu search is used for the intensification procedure as a local search and ACS is used with a specific fitting that allows a diversification.

1.3.1.1 Segmentation Criterion: Biased Survival Exponential Entropy

In this section, the information measure of random variables, called Survival Exponential Entropy (SEE), is recalled. Here, we show how using a biased version of this measure can allow to segment images.

```
The EACS algorithm
Data: image histogram, number of classes ($N$)
for each ant do
    Choose the first threshold randomly
    for $i = 2$ to $N - 1$ do
        Build a list of the candidates thresholds
        Choose a threshold that minimizes the segmentation criterion
        $r = rand$
        if $r < 0.3$ then
            Apply a Tabu search in the neighbor of the current threshold
        end
        Local update of the pheromone
    end
    Global update of the pheromone
end
Retrun The best ant
```

Fig. 1.2 The EACS algorithm

Let $X = (X_1 \ldots X_m)$ be a random vector in $\Re^m$. Denoted by $|X|$, the random vector with components $|X_1|, \ldots, |X_m|$ and use the notation $|X| > x$ to mean that $|X_i| > x_i \; for \; x_i \geq 0, i = 1, \ldots, m$.

The multivariate survival function $\overline{F}_{|X|}(x)$ of the random vector $|X|$ with an absolutely continuous distribution with probability density function $f(x)$ is defined by:

$$\overline{F}_{|X|}(x) = P\left(|X_1| > x_1, \ldots, \; |X_m| > x_m\right) \tag{1.7}$$

where $x \in R_+^m$. For a discrete distribution, the survival cumulative distribution function can be expressed as:

$$\bar{F}(x) = 1 - \sum_{i=0}^{x} p(i) \tag{1.8}$$

For the random vector X in $\Re_+^m$, the survival exponential entropy of order α is defined by:

$$M_\alpha(X) = \left(\int\limits_{R_+^m} \overline{F}_{|X|}^{\alpha}(x)\, dx \right)^{1/1-\alpha} \tag{1.9}$$

for $\alpha > 0$ and $\alpha \neq 0$, where m denotes the number of dimensions for X. The SEE uses the density function with the cumulative distribution that is more regular than the density function. The SEE has several advantages over the Shannon entropy and differential entropy (extension of Shannon entropy to the continuous case): it is consistently defined in both the continuous and discrete domains, it is always nonnegative and it is easy to compute from sample data. Whereas the Shannon entropy is based on the density of the random variable, that may not exist in some cases and, when it exists, must be estimated. For the application in image segmentation, we use the well known property of measures of entropy, which states that the entropy of the joint distribution is equal to the sum of entropies of the marginal distributions under the assumption of independence.Thus the proposed biased survival exponential entropies (BSEE) associated with different image classes' distributions are defined below:

- the BSEE of the class $m-1$ can be computed through:

$$BM_{\alpha}^{(m-1)} = \omega_{(m-1)} \cdot \left(\sum_{j=t_{n-1}}^{t_n -1} \left(\overline{F}(i, j)\right)^{\alpha} \right)^{1/(1-\alpha)} \tag{1.10}$$

- the BSEE of the class m can be computed through:

$$BM_{\alpha}^{(m)} = \omega_{(m)} \cdot \left(\sum_{j=t_{n}}^{t_{n+1} -1} \left(\overline{F}(i, j)\right)^{\alpha} \right)^{1/(1-\alpha)} \tag{1.11}$$

where $\omega_{(m)} = log(N_i)$, so that the first term $\left(\omega_{(m)} \cdot M_{\alpha}^{(m)}\right)$ is high for non-homogeneous regions (typically, the large ones), while N_i denotes the number of pixels in the class i, and $\omega_{(1)} = 1$. For the convenience of illustration, two threshold values $t_0 = 1$ and $t_N = 255$ were added, where $t_0 < \ldots < t_N$. Then the total BSEE is:

$$BM_{\alpha}^{(T)} = \sum_{i=0}^{N-1} BM_{\alpha}^{(i+1)} \tag{1.12}$$

According to the minimum survival exponential entropy principle that corresponds to the maximum Shannon entropy principle, the optimal vector ($t_0^* < \ldots < t_N^*$) should meet:

$$BM_{\alpha}^{(T)} = \text{ArgMin} \left\{ BM_{\alpha}^{(T)} \right\} \tag{1.13}$$

where $1 < t_1 < \ldots < 255$. In the case of one threshold ($N = 2$), the computational complexity for determining the optimal vector t^* is $O(L^2)$ where L is the total number of gray level. However, it is too time-consuming in the case of multilevel thresholding. For the n-thresholding problem, it requires $O(L^{n+2})$. In this work, we used a the hybrid ACS, previously presented, for solving the problem formulated in (1.13) efficiently.

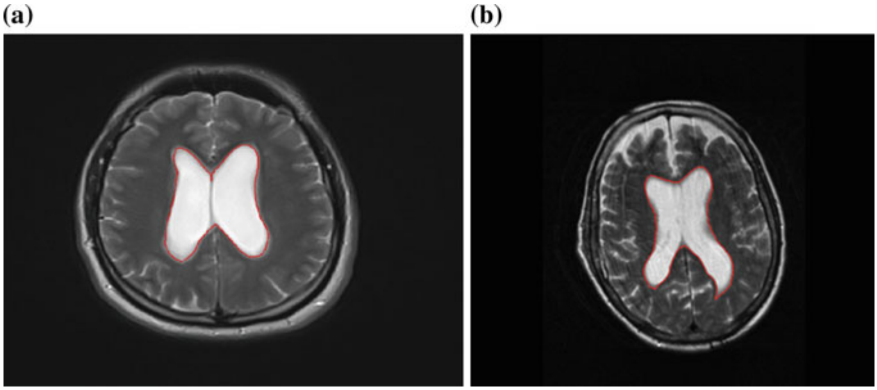


Fig. 1.3 Illustration of the segmentation results using BSEE-EACS on two different MRI (*a*) and (*b*)

1.3.1.2 Results and Discussions

In our experiments, the value of α was equal to 100, in order to have stable performances. The different values of the parameters of EACS algorithm were fixed empirically as follows: Population size = 100, Number of Cycles = $2 \times N$, $\mu = 0.5$, $b = 2$, $T_L = 3 \times N$, $S_n = 10$. The different images were acquired using Siemens Avento 1.5T, the resolution was 256×256 and the field of view equal to 250. The Fig. 1.3 presents an illustration of a segmentation results obtained on two different pathologic MRI cases. Indeed, in this paper the pathology studied the hydrocephalus, that consists in atrophy in the ventricular system. Then, our goal was to calculate the volume of the cerebrospinal fluid (CSF) inside only the ventricle. The proposed algorithm was used as a pre-processing step to calculate the volume. The obtained result through the application of the segmentation algorithm when $N = 2$, shows that the use of the BSEE allows to segment satisfactorily the brain MR images.

1.3.2 Enhanced BBO for Image Segmentation

Biogeography is known as a popular method of studying geographical distribution of biological organisms, whose earliest works can be traced back to the days of Alfred Wallace and Charles Darwin [19]. The mathematical models of biogeography are available which describe the governing laws about migration of specifies from one island to another island, the arrival of new species and the extinction of some existing species. However, only very recently a population based optimization technique has been proposed employing the basic nature of biogeography and it has been named biogeography based optimization (BBO) [19].

The mathematical models of biogeography are available which describe the governing laws about migration of specifies from one island to another island, the arrival of new species and the extinction of some existing species. However, only very recently a population based optimization technique has been proposed employing the basic nature of biogeography and it has been named biogeography based optimization (BBO) [19]. In biogeography models, the fitness of a geographical area (called "island" or "habitat") is judged on the basis of habitat suitability index (called HSI). A habitat with a high HSI indicates that it is more suited for species to reside here. Similarly a habitat with a low HSI indicates that it is less suited for species to reside there. It is natural that higher the HSI of a habitat, it is likely that more number of species will be present there. The variables that characterize habitability, e.g. rainfall, vegetation, temperature etc., are called suitability index variables (SIVs). The dynamics of the movement of the species among different habitats is mainly governed by two parameters, called immigration rate (λ) and emigration rate (μ) and these two parameters are functions of the species count in a habitat.

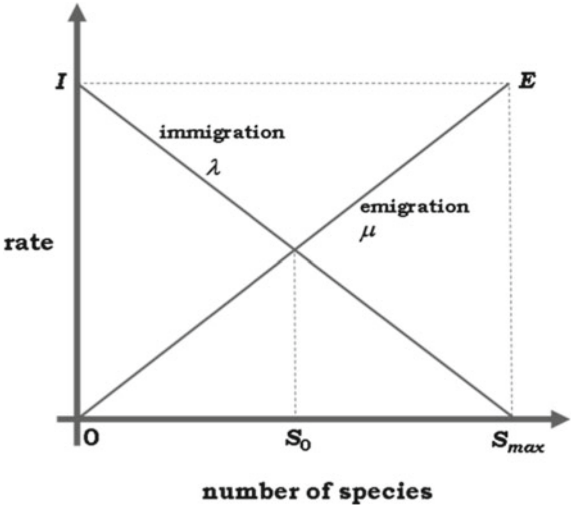
Figure 1.4 shows the species model of a single habitat [12, 19]. The curve shows a special case when maximum immigration rate (I) and maximum emigration rate (E) are equal. However, strictly speaking, there is no such constraint and the condition of equality can be easily relaxed. In this curve the maximum number of species that a habitat can host is considered to be S_{max}. When the number of species (S) is small, there is more possibility of immigration of species from neighboring habitats and less possibility of emigration of species from this habitat to neighboring habitats. With the increase in the population of species, the possibility of immigration to the habitat decreases and the possibility of emigration from the habitat increases. This is characterized by the two curves for immigration rate and emigration rate in Fig. 1.4. When $S = S_0$, the equilibrium condition is reached i.e. in a given time span, the same number of species immigrate to and emigrate from the habitat.

Let the probability that the habitat contains S species at time t be given by P_s, then, the dynamic equations of the probabilities of species count in the habitat can be defined using a matrix relation.

$$\dot{P}_s = \begin{cases} -(\lambda_s + \mu_s)\, P_s + \mu_{s+1} P_{s+1} & S = 0 \\ -(\lambda_s + \mu_s)\, P_s + \lambda_{s-1} P_{s-1} + \mu_{s+1} P_{s+1} & 1 \leq S \leq S_{max} - 1 \\ -(\lambda_s + \mu_s)\, P_s + \lambda_{s-1} P_{s-1} & S = S_{max} \end{cases} \tag{1.14}$$

As there is a possible maximum of S_{max} number of species in the habitat, one can obtain a matrix relation governing the dynamic equations of the probabilities of species count in the habitat:

Fig. 1.4 Curves for immigration and emigration rates in basic BBO algorithm



$$\begin{bmatrix} \dot{P}_0 \\ \dot{P}_1 \\ \vdots \\ \vdots \\ \dot{P}_{S_{max}} \end{bmatrix} = \begin{bmatrix} -(\lambda_0+\mu_0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \mu_n \\ 0 & \cdots & -(\lambda_n+\mu_n) \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ \vdots \\ P_{S_{max}} \end{bmatrix} \tag{1.15}$$

1.3.2.1 Enhancement of Biogeography Based Optimization (BBO) Algorithm

These basic ideas of biogeography have been utilized to design a population based optimization procedure that can be potentially used to solve many engineering and other optimization problems. As there are a lot of similarities between the mathematical model of biogeography and the population based optimization algorithms. The BBO algorithm proposed in [19] designate each habitat H as a potential $m \times 1$ decision variable vector, where $H \in SIV^m$ i.e. each habitat or solution comprises m $SIVs$. For each habitat H, its HSI corresponds to the fitness function in population-based algorithms. A habitat with a higher HSI indicates that it is a better candidate for the optimum solution. It is considered that the ecosystem has n habitats i.e. the ecosystem is H^n. In the context of population based metaheuristics, it means there are a total of n possible candidate solutions (i.e. the population size). The overall scheme of BBO is presented in Fig. 1.5. In the basic BBO algorithm, the immigration and emigration rates follow as linear variations of species count. They are described using the following equations:

```
BBO
Inputs: $ItMax$: maximum number of iterations
while $i < ItMax$ do
    Evaluate the $HSI$ (fitness) of each solution
    Compute $S, \lambda, and\mu$ for each solution
    Modify habitats (Migration) based on $\lambda$ and $\mu$
    Mutation based on probability
    Perform elitism to keep only the best solutions
end
```

Fig. 1.5 Classical biogeographic optimization algorithm

$$\lambda_S = I * \left(1 - \frac{S}{S_{max}}\right) \tag{1.16}$$

and

$$\mu_S = \frac{E * S}{S_{max}} \tag{1.17}$$

In our work, we proposed an enhancement of BBO by implementing nonlinear variations of immigration rate and emigration rate with the number of species in a habitat. Indeed, the basic spirit of these variations will not violate the original considerations i.e. the immigration rate should decrease and emigration rate should increase with number of species. Our proposed variations for these two rates can be described as:

$$\lambda_S = I * \left(1 - \frac{S}{S_{max}}\right)^{p1} \tag{1.18}$$

and

$$\mu_{S_{nlin}} = E * \left(\frac{S}{S_{max}}\right)^{p2} \tag{1.19}$$

Theoretically both p_1 and p_2 can be chosen in the range $[0, \quad \infty[$. The basic BBO algorithm is a special case of these improved variants proposed in (6) and (7), where $p_1 = p_2 = 1.0$. Figure 1.2 shows typical variations of these proposed immigration and emigration rates for the case, when their maximum permissible values are same i.e. $E = I$. However, this condition is not really a constraint and it can be relaxed, if needed. The proposed modifications allows to have more flexibility and, depending on the image(s) under consideration, one can choose an appropriate set accordingly.

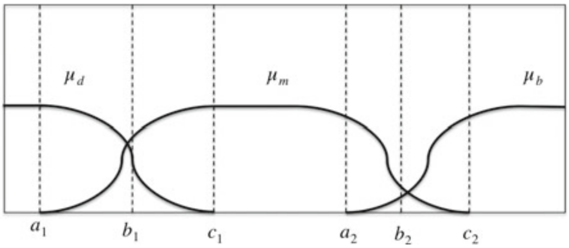


Fig. 1.6 Membership function graph

1.3.2.2 The Segmentation Criterion: Overall Probability Error

The maximum fuzzy entropy measure was proposed in the works of Tao et al. [21, 22], which were inspired by an earlier work of Zhao et al. [27], where an entropy function was used to measure the compatibility between the fuzzy c-partition (FP) and the probability partition (PP). In Taos work [21] the entire image is classified into three partitions of dark pixels, medium pixels and bright pixels and each partition is characterized by a fuzzy membership function (MF). The dark pixels are characterized by a Z-shaped MF, the medium pixels are characterized by a $\prod$-shaped MF and the bright pixels are characterized by an S-shaped MF. An example of three membership functions are shown in Fig. 1.6. In this method, the three classes are associated to three membership functions, respectively, μ_d function of the class dark, the function μ_m, and tbe μ_b function. Then, the segmentation problem consists in finding the optimal thresholds vector that allows to maximize the total fuzzy entropy.

1.3.2.3 Results and Discussions

The proposed segmentation method was used to segmentation CT-Scan images. These images were acquired from the face CT scan of a volunteer in "*Centre Hospitalier Universitaire (CHU) Henri Mondor, Créteil (France)*". The image acquisition system employed was the PHILIPS famous multi-slice CT scanner. The resolution was 0.4883 mm per pixel, each slice was 0.9 mm thick, and the spacing between the consecutive slices is 0.45 mm.

The obtained results are illustrated by the one in Fig. 1.7. More results can be found in [5]. In this work, we demonstrated that some form of the enhanced BBO could largely outperform both basic BBO and other metaheuristics for the posed problem. It is recommended that the choice of a specific enhanced BBO based variant should be carried out on a trial and error basis. However, our results have provided a thumb rule which can be adopted in providing a possible search direction for choosing the specific improved BBO based variant, where it is expected to obtain satisfactory segmentation performance.

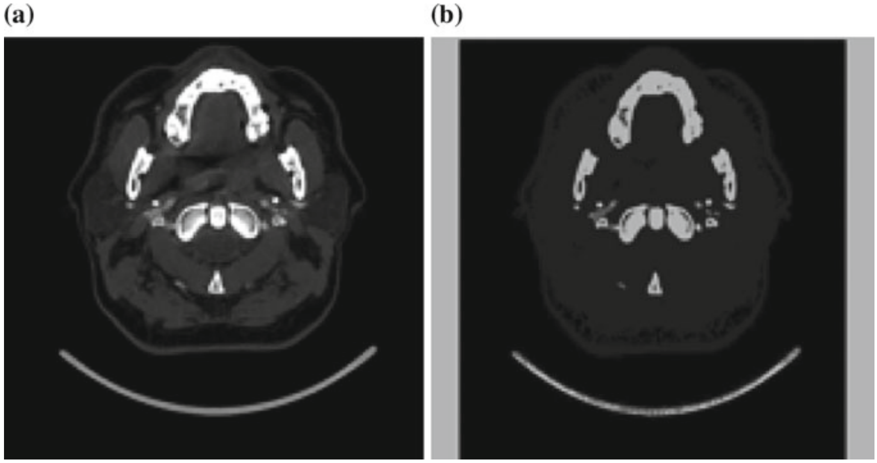


Fig. 1.7 Illustration of the segmentation results using Enhanced BBO. **a** Original CT-Scan slice, **b** Segmented image using Enhanced BBO

1.3.3 Enhanced DE for Image Thresholding

Differential Evolution algorithm (DE) [18] is one of the most popular metaheuristic for solving continuous optimization problems. Recently, it has gained much popularity in different kinds of applications, because of its simplicity and robustness in comparison with other evolutionary algorithms [23]. DE has very few parameters to adjust, making it particularly easy to implement for a diverse set of optimization problems [2, 4, 11]. This work proposes the development of a new optimal multilevel thresholding algorithm based on image histograms by employing an improved version of DE called EDE. After fitting the Gaussian curves using EDE, the optimal threshold is calculated by minimizing the overall probability error between these Gaussian distributions.

In this section we briefly describe the EDE, an enhanced version of basic DE. EDE uses the concepts of opposition based learning, random localization and has a one population set structure. The working of EDE is as follows.

Population initialization

EDE starts with a population $S = \{X_1, X_2, \ldots, X_N P\}$ of NP solutions: $X_i = \left(x_{(1,i)}, x_{(2,i)}, x_{(n,i)}\right)$ with $i = 1, \ldots, NP$, where the index i denotes the i^{th} solution of the population. For this we randomly construct a population $P1$ of NP solutions, using the following rule:

$$x_{i,j} = x_{\min,j} + rand(0,1) \times (x_{\max,j} - x_{\min,j}) \tag{1.20}$$

where $x_{min,j}$ and $x_{\max,j}$ are lower and upper bound for j^{th} component respectively and $rand(0, 1)$ is a uniform random number between 0 and 1. We construct another population P_2 of NP opposite solutions to those in the population P_1 using the following rule:

$$y_{i,j} = x_{\min,j} + x_{\max,j} - x_{i,j} \tag{1.21}$$

where xi, j is the component of solution Xi of population P_1. Now the initial population S is constructed by taking NP best solutions from union of P_1 and P_2.
Mutation
The mutation operator of EDE applies the vector difference between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three solutions $\{X_{r1}, X_{r2}, X_{r3}\}$ from the population corresponding to target solution X_i. A tournament is then held among the three solutions and the region around the best point is explored. That is to say if X_{r1} is the point having the best fitness function value, then the region around it is searched with the hope of getting a better solution. Assuming that $X_{tb} = X_{r1}$, the mutation equation is given as:

$$V_i = X_{tb} + F \times (X_{r2} - X_{r3}) \tag{1.22}$$

where $r_1, r_2, r_3 \in 1, \ldots, NP$ are randomly selected such that $r_1 \neq r_2 \neq r_3 \neq i$, and F is the control parameter such that $F \in [0, 1]$. This variation gradually transforms itself into search intensification feature for rapid convergence, when the points in S form a cluster around the global minima.

Crossover
The Crossover operator of EDE is same as that of DE. According to it, once the perturbed individual $V_i = \left(v_{1,i}, \ldots, v_{n,i}\right)$ is generated, it is subjected to crossover operation with target individual $X_i = \left(x_(1, i), x_{(2,i)}, x_{(n,i)}\right)$, that finally generates the trial solution, $U_i = \left(u_{1,i}, \ldots, u_{n,i}\right)$, as follows:

$$u_{j,i} = \begin{cases} v_{j,i} \text{ if } rand_j \leq C_r \vee j = k \\ x_{j,i} \;\; Otherwise \end{cases} \tag{1.23}$$

where $j = 1, \ldots, n, k \in 1, \ldots, n$ is a random parameter index, chosen once for each i. The crossover rate, $Cr \in [0, 1]$, is set by the user.

Selection The selection operator of EDE for new solutions is different from that of DE. After the generation of a new solution, selection operation is performed between it and its corresponding target solution by the following equation:

$$X'_i = \begin{cases} U_i \text{ if } f(U_i) \leq f(X_i) \\ X_i \quad Otherwise \end{cases} \tag{1.24}$$

If the new solution is better than the target solution then it replaces the old one in the current population. While in DE, the better one of these two solutions is added to an auxiliary population, two populations (current and auxiliary) are considered simultaneously in all the iterations, which results in the consumption of extra memory and CPU time. On the other hand, in EDE, only one population is maintained and the individuals are updated when a better solution is found. Also, the newly found better solution that enters the population instantly takes part in the creation of new solution.

1.3.3.1 Segmentation Criterion

The segmentation criterion used in this work is based on the approximation of the image histogram by a Gaussian mixture model. Indeed, over the years, many authors have proposed several algorithms to solve Gaussian mixture model for multi-level thresholding. Besides, Snyder et al. [20] presented an alternative method for fitting curves based on a heuristic method called tree annealing; we also proposed a fast scheme for optimal thresholding using simulated annealing algorithm [15, 16]; Zahara et al. [25] proposed a hybrid Nelder-Mead Particle Swarm Optimization (NM-PSO) method and more recently a hybrid method based on Expectation Maximization (EM) and Particle Swarm Optimization (PSO+EM) was proposed in [7]. All these metaheuristics based methods are efficient in solving the multi-level thresholding problem and could provide better effectiveness than the other traditional methods (local search and deterministic methods). However, curve fitting is usually time-consuming, which indicates that improved methods are yet needed to enhance the efficiency, while maintaining effectiveness, and these methods have many parameters that must be well fitted.

This segmentation approach consists in two optimization problems: the first is that of the Gaussian curve fitting problem. While the second is the minimization the overall probability error of miss classification of the pixels. The first problem is a continuous optimization problem, while the second is a discrete optimization problem.

1.3.3.2 Results and Discussions

The proposed algorithm EDE has only 4 parameters that must be well fitted. We have done preliminary testing for the purpose of getting suitable values of these parameters and results are listed in Table 1.1, and the initial population is generated

Table 1.1 Parameters of EDE. $N = 3 \times D$ where D is the number of classes (to be fixed by the user)

Parameter	Value
Population size	$10 \times N$
Scaling factor F	0.25
Crossover rate C_r	0.25
Maximum number of iterations	200

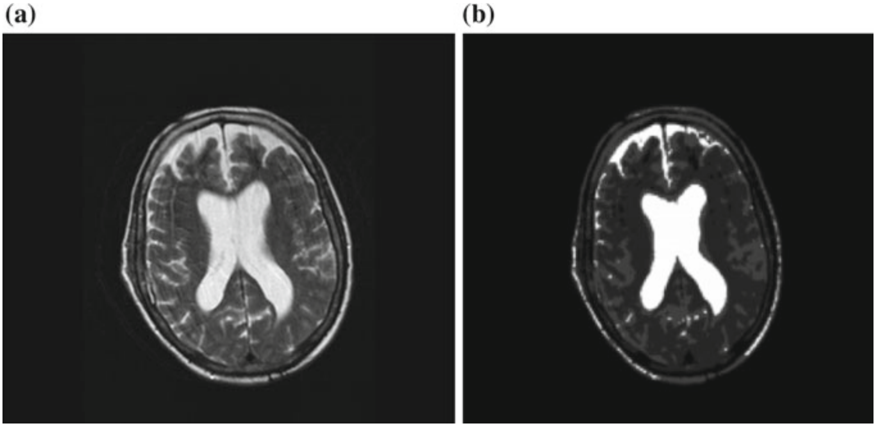


Fig. 1.8 Illustration of the segmentation results using Enhanced DE. **a** Original pathologic MRI 2D image, **b** 5 classes segmented image using Enhanced DE

randomly under some considerations. Moreover, the stopping criterion we used for the algorithm is the maximum number of iterations.

The obtained results through the application of our segmentation algorithm are illustrated through the original brain MRI in Fig. 1.8. Figure 1.8 shows the original images and its multilevel classification (segmented) version when the number of thresholds is 4 (5 classes segmentation). The number of classes is an input parameter of the segmentation algorithm. A non-supervised technique for determining the number of classes was proposed in [16] that can be used at initialization. Our goal is to detect the different spaces and th white matter surrounding the ventricular space quickly.

1.3.4 Metaheuristic for Contours Detection in 2D Ultrasound Images

In Computer Assisted Orthopedic Surgery (CAOS) systems, the intra-operative image modality of choice is often Computed Tomography (CT) or fluoroscopy (X-rays projection). These image modalities are not completely safe for the patient, and for the surgeon who uses them everyday. Within the last decade, ultrasounds (US) became an interesting alternative for orthopedic surgeons. It is well known that US devices are not too expensive, and portable; it also can be used in real-time intra-operatively and it is non-invasive. However, the US images are difficult to analyze for the surgeon, because of the high level of attenuation, shadow, speckle and signal dropouts [10].

In the literature, the extraction of the bone surface in US images was studied in [9]. The authors used a A-mode ultrasound pointer. The probe was tracked mechanically, and it was used to register the distal femur in total hip replacement. The A-mode of ultrasound probes consists in using only one ultrasound beam. Then, the output image is a one dimensional vector. Usually, the used mode is the B-mode, where the resultant image is a matrix, and the number of beams is greater than one. In CAOS systems, ultrasounds can be used to collect some sample points on bone surface [3], or to perform intra-operative registration, extracting the full 3D model [26]. Manual segmentation of the bone surface in US images is highly operator dependent and time consuming [1]. Moreover, the thickness of the response can reach 4 mm in some cases [10], and it can lead to a high error. In [8] developed an automatic segmentation method of bone surface in US images using dynamic programming. This method depends on a threshold value, the obtained average error was between 2.10–2.67 pixels at the comparison between automatic and manual segmentation; the average time of computation per image were 0.55 s.

In this work, our main interests lies in the use of US images in computer assisted intramedullary nailing of tibia shaft fractures. If a surgeon choose to heal a tibia shaft fracture using an intramedullary nail, then he has to lock the nail in the bone. Normand et al. proposed to use some measures on the healthy symmetric tibia to assist the surgeon during the locking of the nail [17]. To do so, the 3D position of some anatomical landmarks is needed (malleolus, trochlea, femoral condyles, …), and the healthy tibia should not be cut. Then, the authors proposed to use the US probe as a subcutaneous pointer. The main goal of this new method is to extract automatically, and in real-time, the bone surface from US images, and particularly, the anterior femoral condyles.

The proposed method consists in two main steps. In the first step, a vertical gradient is applied to extract potential segments of bone from 2D US images. In the second step, *a new method based on shortest path* is used to eliminate all pixels that do not belong to the final contour. Finally, the contour is closed using polynomial interpolation.

1.3.4.1 Proposed Bone Contour Extraction Method Based on Shortest Path

Let $I : \Omega \subset \mathbb{N}^2 \rightarrow \mathscr{I} \subset \mathbb{N}$ be an image (two dimensional (2D) real function). Segmenting bone surface in I consists in extracting $\{P_i | i = 1, \dots, n\}$ a subset of contiguous points in I, where $P_i = (x_i, y_i) \in \Omega, \forall i = 1, \dots, n$. Considering ultrasound properties of bones [10], we admit that $\forall (i, j) \in [1, n]^2$ such that $i \neq j$, $y_i \neq y_j$.

Then, the proposed segmentation method consists in three steps: in the first step, original images are filtered, a vertical gradient is computed, and an extraction of some potential segments of bone contour is performed. Then, the second step consists in characterizing these segments of contour, in order to eliminate those that a priori do not belong to the bone contour. Final step consists in closing the contour using least square polynomial approximation.

It is well known that ultrasound images are highly textured, mainly with speckle. Then, the first step consists in a pre-processing step where a low-pass filter is applied

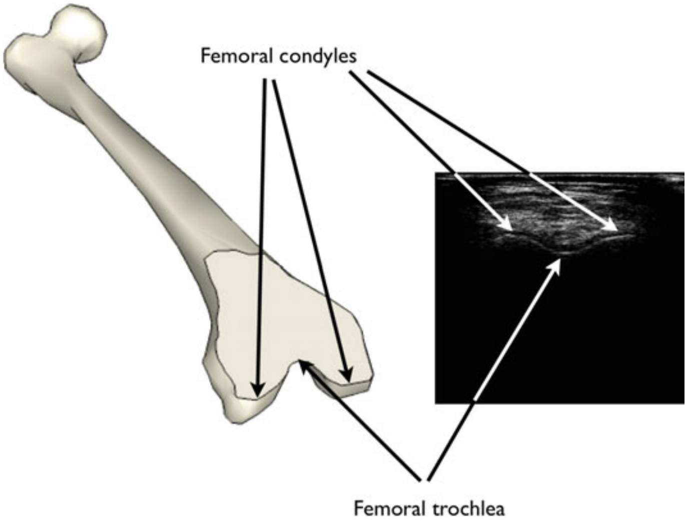


Fig. 1.9 Original US image

to the original image (an example of images at hand is presented in Fig. 1.9) in order to eliminate noise and to strengthen interesting features. Once the filtered image I_s is computed, the vertical gradient is applied to the filtered image, we denote the result image I_g. The choice of the vertical gradient was motivated by ultrasound propagation properties, where the bone contours are mainly horizontal. It was shown in [10] that it is suitable that the bone contour lies on the top of the fiducial surface. Then, we only keep high values of the gradient, we called this image I_{BW}.

Then, using properties of ultrasound imaging on bones [10], we can extract from I_{BW} a first subset of potential contour points $\{Q_i | i = 1, \ldots, c\}$, where c is the number of columns in the original image I. We denote $Q_i = (x_i, i), \forall i = 1, \ldots, c$ in the rest of this section. The subset of Q_i was built by taking the first non-zero point in each column of I_{BW} starting from the bottom of the image. The next step consists in characterizing these points to determine whether or not they belong to the bone contour.

1.3.4.2 False Alarm Elimination Using Shortest Path Formulation

The subset of points $\{Q_i | i = 1, \ldots, c\}$ are potentially part of the bone contour. To select those that belong to the bone contour, we consider them as segments by grouping contiguous points. Two points are considered to be contiguous if they belong to the same neighborhood. In this step, all small segments that are very likely noise, and segments that are too close to the skin are eliminated automatically, and not considered. For each segment k, where $k = 1, \ldots, M$, the first point is designated

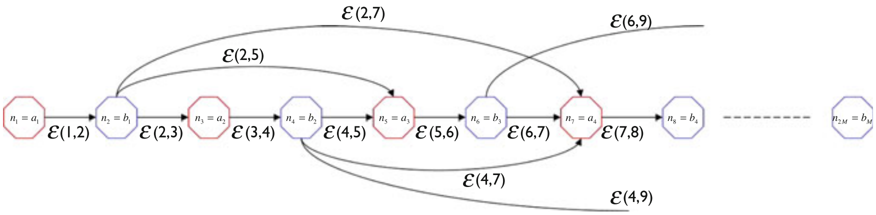


Fig. 1.10 The construction of the graph $\mathscr{G}$. We distinguish nodes called "*start of segments*" which are the a_k nodes and the nodes called "*end of segments*" which are the b_k nodes

by Q_{a_k} and the last point by Q_{b_k}, where a_k and b_k are the column of Q_{a_k} and Q_{b_k}, respectively.

To define segments that belong to the bone contour, we define $\mathscr{G}(\mathscr{N}, \mathscr{E})$ as an oriented graph from the M segments (Fig. 1.10):

$$\begin{aligned}\mathscr{N} &= \{n_i \mid i = 1, \ldots, 2M\}\\ &= \{a_k \mid k = 1, \ldots, M\} \cup \{b_k \mid k = 1, \ldots, M\}\end{aligned} \tag{1.25}$$

is the set of all nodes in the graph, where the node index n_i is defined by:

$$\forall i \in [1, 2M], \quad n_i = \begin{cases} a_{\frac{i+1}{2}} & \text{if } i \text{ is odd} \\ b_{\frac{i}{2}} & \text{if } i \text{ is even} \end{cases} \tag{1.26}$$

We define also the set of edges in the graph as: $\forall (i, j) \in [1, 2M]^2$,

$$\mathscr{E}(i, j) = \begin{cases} \frac{1}{2}(b_{\frac{i}{2}} - a_{\frac{i}{2}}) & \text{if } i \text{ is odd and } j = i + 1 \\ \| Q_{b_{\frac{i+1}{2}}} - Q_{a_{\frac{j+1}{2}}} \| & \text{if } i \text{ and } j \text{ are even, and if } i < j \leq min(2M, i + 6) \\ 0 & \text{otherwise} \end{cases} \tag{1.27}$$

Then, $\mathscr{G}$ is a graph with two types of node: the first nodes of segments, which are $\{a_k \mid k = 1, \ldots, M\}$ with an only one child, b_k and the weight of the edge between them is $\frac{1}{2}(b_{\frac{i}{2}} - a_{\frac{i}{2}})$, and the last nodes of segments which are $\{b_k \mid k = 1, \ldots, M\}$ with at most three children which are $\{b_l \mid l = k + 1, \ldots, min(k + 3, M)\}$, and the weight of edges between them are $\| Q_{b_{\frac{i+1}{2}}} - Q_{a_{\frac{j+1}{2}}} \|$. We penalize the intra-segment distance because the path to be minimized is the inter-segment one. For each segment k, a_k is used to include the segment length in the shortest path computation, and b_k is used to eliminate non-bone segments.

Then, to solve the shortest path problem any metaheuristic can be applied to solve it. In our work, genetic algorithms, ant colony optimization algorithm and Dijkstra's algorithm were tested.

Finally, the closure of the contour is performed by a polynomial approximation using least square method. The contour is computed using points that belong to the remaining set of segments.

1.3.4.3 Results and Discussions

To show the efficiency of the proposed method, different tests were performed on several series of ultrasound images. The probe and the used beamformer were provided by Telemed (Vilnius, Lituania), and a software developed by Aesculap SAS (Echirolles, France) was used for the acquisition. The acquisition protocol consists in putting the probe under the patella, and to perform a scan of the femoral condylar region rotating the probe up and down. Figure 1.11 illustrates the different steps of the proposed framework and especially the contours detection step. One can see that the proposed approach to detect the contours is efficient and low complex because it allows to track the contours in real time and quality of the results was validated on large database. More details about this contribution can be found in [13].

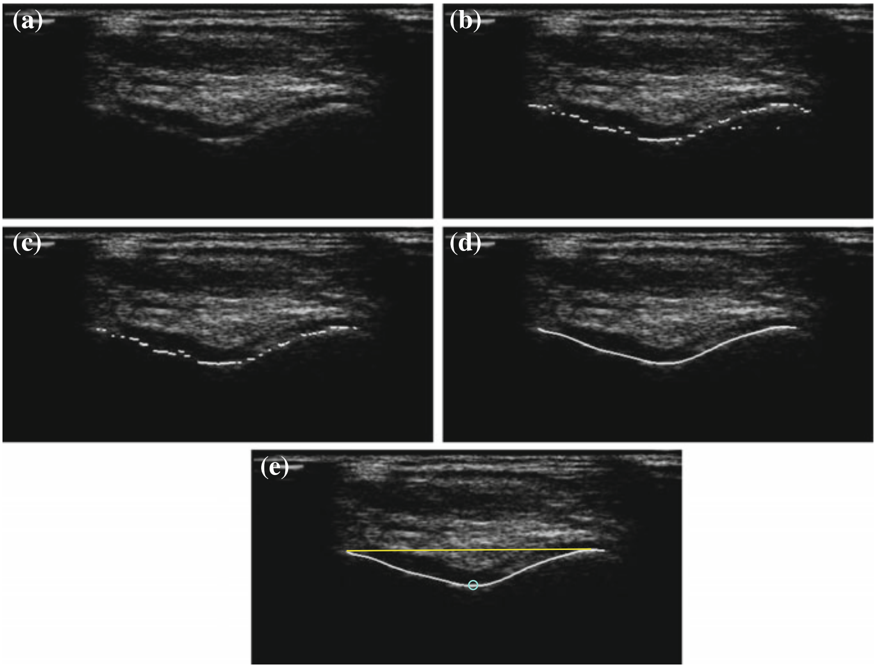


Fig. 1.11 Extraction of femoral condyles and trochlea in an ultrasound image, after the calculation of the bone contour using proposed method. **a** Ultrasound image of femoral condyles, **b** First set of the potential bone contour pixels, **c** Results of the shortest path based step, **d** Result after interpolation step, **e** Extraction of the femoral condyles and trochlea in an US image

1.4 Conclusion

In this chapter, we first formulated the image segmentation problem as an ill posed problem and its formulation as an optimization problem. Then, we outlined our main contributions. Over the last few years, our work has focused on two areas: improving the performance of metaheuristics and developing new segmentation criteria.

Our contributions presented in this chapter can be summarized on the following points:

- New procedure for initializing metaheuristics based on lows discrepancy sequences.
- Adding tabu memory to ACS.
- New formulation of the image thresholding problem as a graph cut problem.
- New formulation for contours detection based on shortest path optimization problem.

Moreover, a new segmentation criteria and formulation based on survival exponential entropy and a new method to extract contours in the case of US images were proposed.

Acknowledgements The author would like to thank, Agnes Masson-Sibut, Salma Hajjem, Hamouche Oulhadj, Amitava Chaterjee and Patrick Siarry for their collaboration.

References

1. D.C. Barratt, G.P. Penney, C.S.K. Chan, M. Slomczykowski, T.J. Carter, P.J. Edwards, D.J. Hawkes, Self-calibrating 3D-ultrasound-based bone registration for minimally invasive orthopedic surgery. IEEE Trans. Med. imaging **25**(3), 312–323 (2006)
2. A. Bastürk, E. Günay, Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm. Expert Syst. Appl. **36**(2), 2645–2650 (2009)
3. M. Beek, P. Abolmaesumi, S. Luenam, R.W. Sellens, D.R. Pichora, Ultrasound-guided percutaneous scaphoid pinning: operator variability and comparison with traditional fluoroscopic procedure, in *MICCAI '06: Proceedings of the 9th International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 9 (2006), pp. 536–43
4. W.-D. Chang, Parameter identification of rosslerś chaotic system by an evolutionary algorithm. Chaos Solitons Fractals **29**(5), 1047–1053 (2006)
5. A. Chatterjee, P. Siarry, A. Nakib, R. Blanc, An improved biogeography based optimization approach for segmentation of human head CT-scan images employing fuzzy entropy. Eng. Appl. Artif. Intell. **25**(8), 1698–1709 (2012)
6. M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evolut. Comput. **6**(4), 317–365 (1997)
7. S.-K.S. Fan, Y. Lin, A multi-level thresholding approach using a hybrid optimal estimation algorithm. Pattern Recognit. Lett. **28**, 662–669 (2007)
8. P. Foroughi, E. Boctor, M.J. Swartz, R.H. Taylor, G. Fichtinger, Ultrasound Bone Segmentation Using Dynamic Programming, in *2007 IEEE Ultrasonics Symposium Proceedings* (IEEE, New York, USA, 2007), pp. 2523–2526
9. S. Heger, F. Portheine, J.A.K. Ohnsorge, E. Schkommodau, K. Radermacher, User-interactive registration of bone with A-mode ultrasound. IEEE Eng. Med. Biol. Mag. **24**(2), 85–95 (2005)

10. A.K. Jain, R.H. Taylor, Understanding bone responses in B-mode ultrasound images and automatic bone surface extraction using a Bayesian probabilistic framework, in *Proceedings of International Conference SPIE Medical Imaging*, vol. 5373 (SPIE, Bellingham, WA, USA, 2004), pp. 131–142
11. B. Liu, L. Wang, Y-H. Jin, D-X. Huang, F. Tang, Control and synchronization of chaotic systems by differential evolution algorithm. Chaos, Solitons Fractals **34**(2), 412–419 (2007)
12. R. MacArthur, E. Wilson, *The Theory of Biogeography* (Princeton University Press, Princeton, NJ, 1967)
13. A. Masson-Sibut, A. Nakib, E. Petit, F. Leitner, A new automatic landmarks extraction framework on ultrasound images of femoral condyles, in *Proceedings of the SPIE 8320, Medical Imaging 2012: Ultrasonic Imaging, Tomography, and Therapy, 83200U* (San Diego, California, USA, 2012). doi:10.1117/12.910604
14. O. Monga, An optimal region growing algorithm for image segmentation. Int. J. Pattern Recognit. Artif. Intell. **1**(3), 351–376 (1987)
15. A. Nakib, H. Oulhadj, P. Siarry, Image histogram thresholding based on multiobjective optimization. Signal Process. **87**, 2516–2534 (2007)
16. A. Nakib, H. Oulhadj, P. Siarry, Non supervised image segmentation based on multiobjective optimization. Pattern Recognit. Lett. **29**, 161–172 (2008)
17. J. Normand, A. Harisboure, F. Leitner, J.B. Pinzuti, E. Dehoux, A. Masson-Sibut, Experimental navigation for bone reconstruction, in *10th Annual Meeting of The International Society for Computer Assisted Orthopaedic Surgery Proceedings* (Versailles, France, 2010) p. 39
18. K. Price, R. Storn, J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization* (Springer, Heidelberg, 2005)
19. D. Simon, Biogeography-based optimization. IEEE Trans. Evolut. Comput. **12**(6), 702–713 (2008)
20. W. Synder, G. Bilbro, A. Logenthiran, S. Rajala, Optimal thresholding a new approach. Pattern Recognit. Lett. **11**, 803–810 (1990)
21. W. Tao, J.-W. Tian, J. Liu, Image segmentatn by three-level thresholding based on maximum fuzzy entropy and genetic algorithm", pattern recognition letters. Pattern Recognit. Lett. **24**, 3069–3078 (2003)
22. W. Tao, H. Jin, L. Liu, Object segmentation using ant colony optimization algorithm and fuzzy entropy. Pattern Recognit. Lett. **28**, 788–796 (2007)
23. J. Vesterstrom, R. Thomsen, A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems, in *Proceedings of IEEE Congress on Evolutionary Computation 2004 (CEC'2004)* (Portland, Oregon, USA, 2004), pp. 1980–1987
24. D.J. Withey, Z.J. Koles, Three generations of medical image segmentation: methods and available software. Int. J. Bioelectromag. **9**(2), 67–68 (2007)
25. E. Zahara, S-K.S. Fan, D-M. Tsai, Optimal multi-thresholding using a hybrid optimization approach. Pattern Recognit. Lett. **26**, 1082–1095 (2005)
26. Y. Zhang, R. Rohling, D.K. Pai, Direct surface extraction from 3D freehand ultrasound images, in *Proceedings of the Conference on Visualization'02* (IEEE Computer Society, Boston, MA, USA, 2002), p. 52
27. M.S. Zhao, A.M.N. Fu, H. Yan, A technique of three-level thresholding based on probability partition and fuzzy 3-partition. IEEE Trans. Fuzzy Syst. **9**(3), 469–479 (2001)

Chapter 2
Multi-level Image Thresholding Based on Hybrid Differential Evolution Algorithm. Application on Medical Images

M. Ali, P. Siarry and M. Pant

2.1 Introduction

Image thresholding is definitely one of the most popular segmentation approaches for extracting objects from the background, or for discriminating objects from objects that have distinct gray-levels. It is typically simple and computationally efficient. It is based on the assumption that the objects can be distinguished by their gray levels. The optimal threshold is the one that can separate different objects from each other or from the background to such an extent that a decision can be made without further processing [8, 13]. The automatic fitting of this threshold is one of the main challenges of image segmentation. Sezgin and Sankur [18] have presented a survey of a variety of thresholding techniques. There are a lot of approaches classifying thresholding methods. Authors in [18] labeled the method according to the information they exploit, such as histogram shape, space measurement clustering, entropy, object attributes, spatial information and local gray-level surface. Another classification approach consists in dividing these techniques into parametric and non-parametric techniques. The parametric thresholding methods exploit the first-order statistical characterization of the image to be segmented. Weszka et al. [16] proposed a parametric method where the gray-level distribution of each class is assumed to be a Gaussian distribution. An attempt to find an estimate of the parameters of the distribution that best fit the given histogram data is made by using the least-squares estimation method. Typically, it leads to a nonlinear optimization problem, its solution is computationally expensive and time consuming. Over the years, many researchers have proposed several algorithms to solve the objective function of Gaussian curve fitting for multi-level

M. Ali · M. Pant
Department of Paper Technology, Indian Institute of Technology Roorkee,
Roorkee 247667, India

P. Siarry (✉)
Laboratory of LISSI, Université Paris Est, Champs-sur-Marne, France
e-mail: siarry@u-pec.fr

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_2

thresholding. For example, Snyder et al. [10] presented an alternative method for fitting curves based on a heuristic method called tree annealing; Nakib et al. [11, 19] proposed a fast scheme for optimal thresholding using a simulated annealing algorithm; Zahara et al. [7] proposed a hybrid Nelder–Mead Particle Swarm Optimization (NM-PSO) method. More recently a hybrid method based on Expectation Maximization (EM) and Particle Swarm Optimization (PSO+EM) is proposed in [14] and the application of basic Differential Evolution (DE) for solving image segmentation problem is shown in [6], and recently in [2]. Moreover, the application of the artificial bees algorithm can be found in [3]. All these metaheuristic based methods are efficient in solving the multi-level thresholding problem and could provide better effectiveness than the other traditional methods (local search and deterministic methods). However, curve fitting is usually time-consuming which indicates that improved methods are needed to enhance the efficiency of existing methods while maintaining quality effectiveness. Further, these methods also have many parameters that must be well fitted. In the present study we have analyzed whether the thresholding techniques can be further improved if we use a modified variant of DE. In the recent years DE [15] [17] has gained much popularity in different kind of applications because of its simplicity and robustness in comparison to other evolutionary algorithms [17]. DE has very few parameters to adjust, making it particularly easy to implement to a diverse set of optimization problems [1, 5, 9]. This paper proposes the development of a new optimal multilevel thresholding algorithm based on image histograms by employing its improved version called Hybrid Differential Evolution (HDE). After fitting the Gaussian curves using HDE, optimal threshold is calculated by minimizing the overall probability error between these Gaussian distributions. The paper is outlined as follows. Section 2.2 introduces the procedure of Gaussian curve fitting. In Sect. 2.3, the overall probability of error for finding optimal thresholds from fitted Gaussian curves is described. Section 2.4, presents enhanced differential evolution version. Section 2.5 provides the experimental results and discussions, while Sect. 2.6 concludes this research.

2.2 Gaussian Curve Fitting

A properly normalized multimodal histogram $h(x)$ of an image I, where $x \in [0, L-1]$ represents the gray levels and, and L is the total number of gray levels, can be fitted with the sum of d probability density functions (pdf's) for finding the optimal thresholds for use in image segmentation [10]. The case where the Gaussian pdf's are used is defined by:

$$p(x) = \sum_{i=1}^{d} P_i \exp\left[-\frac{(x-\mu_i)^2}{\sigma_i^2}\right] \tag{2.1}$$

where P_i is the amplitude of the Gaussian pdf, μ_i is the mean and σ_i^2 is the variance respectively, of mode i and d is number of Gaussians used to approximate the original histogram and corresponds to the number of the segmentation classes. A pdf model must be fitted to the histogram data, typically by using the maximum likelihood or mean-squared error approach, in order to locate the optimal threshold. Given the histogram data $h(j)$ (observed probability of gray level j), it can be defined as follows:

$$h(j) = \frac{g(j)}{\sum\limits_{i=0}^{d} g(i)} \tag{2.2}$$

where $g(j)$ denotes the occurrence of gray-level j over a given image ranges $[0, L-1]$. Our goal is to find a set of parameters,Θ, that minimizes the fitting error J, given by the following expression [11, 19]:

$$\underset{\Theta}{Min}\ J = \frac{\sum_i |h(i) - p(\Theta, x_i)|}{\sum_i h(i)} \tag{2.3}$$

where i ranges over the bins in the measured histogram. Here, J is the objective function to be minimized with respect to, a set of parameters defining the Gaussian pdfs and the probabilities, is given by:

$$\Theta = \{P_i, \mu_i, \sigma_i\} \tag{2.4}$$

The standard process of setting the partial derivatives to zero results in a set of non-linear coupled equations, the system usually being solved through numerical techniques.

2.3 Overall Probability of Error

After fitting the multimodal histogram, the optimal threshold could be determined by minimizing the overall probability of error, for two adjacent Gaussian pdfs, given by

$$e(T_i) = P_i \int_{infty}^{T_i} p_i(x)dx + P_{i+1} \int_{T_i}^{\infty} \tag{2.5}$$

with respect to the threshold T_i, where $p_i(x)$ is the ith pdf [8]. Then the overall probability to minimize is:

$$E(T) = \sum_{i=1}^{d-1} e(T_i) \tag{2.6}$$

where T is the vector of thresholds: $0 < T_1 < T_2 < \ldots < T_{d-1} < L - 1$. In our case L is equal to 256

To find the thresholds values for which this error is minimal requires differentiating $e(T_i)$ with respect to Ti (using Leibniz's rule) and equating the result to zero. It gives:

$$P_i \times p_i(T_i) = P_{i+1} \times p_{i+1}(T_i) \tag{2.7}$$

This equation is solved for Ti to find the optimum threshold. Using Eq. 2.1 in the general solution of Eq. 9.6 results in the following solution for the threshold T_i:

$$AT_i^2 + BT_i + C = 0 \tag{2.8}$$

where:

$$\begin{aligned} &A = \sigma_i^2 - \sigma_{i+1}^2 \\ &B = 2 \times (\mu_i \sigma_{i+1}^2 - \mu_{i+1}\sigma_i^2) \\ &C = \mu_{i+1}^2 \sigma_i^2 - \mu_i^2 \sigma_{i+1}^2 + 4\sigma_i^2 \sigma_{i+1}^2 \log\left(\frac{P_i \sigma_{i+1}^2}{P_{i+1}\sigma_i^2}\right) \end{aligned}$$

Since a quadratic equation has two possible solutions, only one of them is a feasible solution [6].

2.4 Hybrid Differential Evolution (HDE)

In this section we briefly describe HDE, an enhanced version of basic DE. HDE uses the concepts of opposition based learning, random localization and has a one population set structure. The working of HDE is as follows. *Population initialization:* HDE starts with a population $S = \{X_1, X_2, \ldots, X_{NP}\}$ of NP solutions: $X_i = (x_{1,i}, \ldots, x_{NP,i})$, where the index i denotes the i^{th} solution of the population. For this we randomly construct a population P_1 of NP solutions, using the following rule:

$$x_{i,j} = x_{\min,j} + \mathrm{Sob}(0, 1) \times (x_{\max,j} - x_{\min,j}) \tag{2.9}$$

where $x_{\min,j}$ and $x_{\max,j}$ are lower and upper bounds respectively, for j^{th} component, respectively. $\mathrm{Sob}(0, 1)$ is a number between 0 and 1 from a low discrepancy sequence generated using Sobol's method [4].

We construct another population P_2 of NP opposite solutions to the solutions in population P_1 using the following rule:

$$y_{i,j} = x_{\min,j} + x_{\max,j} - x_{i,j} \tag{2.10}$$

where $x_{i,j}$ is the component of solution X_i of the population P_1.

Now the initial population S is constructed by taking the NP best solutions from union of P_1 and P_2.

Mutation: The mutation operation of HDE applies the vector difference between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation.

The mutation process at each generation begins by randomly selecting three solutions X_{r1}, X_{r2}, X_{r3} from the population corresponding to target solution X_i.

Unlike DE, HDE holds a tournament between the three solutions and the region around the best point is explored. That is to say if X_{r1} is the point having the best fitness function value then the region around it is searched with the hope of getting a better solution. Assuming that $X_{tb} = X_{r1}$, the mutation equation is given as:

$$V_i = X_{tb} + F \times (X_{r2} - X_{r3}) \tag{2.11}$$

where $r1, r2, r3 \in 1, \ldots, NP$ are randomly selected such that $r1 \neq r2 \neq r3 \neq i$, and F is the control parameter such that $F \in [0, 1]$.

This variation gradually transforms itself into a search intensification feature for rapid convergence once the points in S form a cluster around the global minima.
The Crossover: crossover operator of HDE is same as of DE. According to it, once the perturbed individual $V_i = (v_{i,j}, \ldots v_{n,i})$ is generated, it is subjected to a crossover operation with the target individual $X_i = (x_{1,i}, \ldots, x_{n,i})$, that finally generates the trial solution, $U_i = (u1, i, \ldots, un, i)$, as follows:

$$u_{ij} = \begin{cases} v_{i,j} \text{ if } rand_j \leq C_r \text{ or } j = k \\ x_{i,j} \text{ Otherwise} \end{cases} \tag{2.12}$$

where, $j = 1, \ldots, n$ and $k1, ..., n$ is a random parameters index, chosen once for each i. The crossover rate, $C_r \in [0, \ 1]$, is set by the user.

Selection: The selection operator used in HDE is same as that of the classical DE, but the method of updating the solutions differs from that of it. After generation of new solution a selection operation is performed between it and its corresponding target solution by the following equation:

$$X_i' = \begin{cases} U_i \text{ if } f(U_i) \leq f(X_i) \\ X_i \text{ Otherwise} \end{cases} \tag{2.13}$$

If new solution is better than target solution then it replaces target solution in current population. This is in contrast to basic DE, where, the better one of the two is added to an auxiliary population. In DE, two populations (current and auxiliary) are considered simultaneously in all the iterations that result in the consumption of extra memory and CPU time. On the other hand in HDE, only one population is maintained and the individuals are updated when a better solution is found. Also, the newly found better solution that enters the population instantly becomes a variable to take part in the creation of new solution.

2.5 Experimental Results

In this section, we evaluate the performance of the algorithm while implementing Gaussian curve fitting for multi-level thresholding. The test images Sailboat is of size 512 × 512, Cameraman and Lena, are of size pixels with 8 bit gray-levels, taken under natural lighting without the support of any special light source. Test images and their respective normalized histograms are given in Fig. 2.1. The algorithm is implemented on a 2.4 GHz Intel Core i5 Macbook pro with 4GB RAM using Matlab R2013a. The stopping criterion we used for the algorithm is the maximum number of iteration. HDE has only 4 parameters that must be well fitted. We have done preliminary testing for the purpose of getting suitable values of these parameters and the fine tuned results are listed in Table 2.1. The parameters P_i, μ_i and Σ_i are randomly initialized along with some restrictions to each parameter (for example P_i must be between 0 and 1, μ_i must be between 0 and 255).

The experimental results are listed in Table 2.2. This shows the number of classes, parameters of Gaussian curves, the threshold values and the CPU time achieved by the proposed method. The CPU times recorded do not include computation times of the threshold values. Figures 2.2, 2.3, 2.4, 2.5, 2.6 and 2.7 show the results of individual Gaussian curves, fitting to a sum of Gaussian curves to the histograms of the images of Fig. 2.1, and their corresponding segmented images, respectively.

We have conducted two experiments with every image. Experiments on images Cameraman and Sailboat are performed taking three and four classes while in the case of Lena it is three and five. The layout in Figs. 2.2b, 2.3b, 2.4b, 2.5b, 2.6b and 2.7b suggests an easy combination of the Gaussian functions which approaches to shape of the histogram of the original image. Figures 2.2c, 2.3c, 2.4c, 2.5c, 2.6c and 2.7c show the segmented image, in these cases thresholds values are calculated according to (2.7). It is evident that the resulting function approaches the original histogram in all the cases.

In the above experiment, the number of iterations, which is used as stopping criterion, was fixed and corresponding results are noted. However, in order to compare the convergence time of HDE algorithm with basic DE, we have computed the number of function evaluations (NFEs) and the corresponding CPU time for both the algorithms. The run of each algorithm was stopped when the fitting error J of the best solution reached ε. i.e. min $f \leq \varepsilon = 10^-1$, where ε is a threshold value which fixes the accuracy of the measurement. Therefore, the stopping criterion is modified; it is based on the value of the fitting and not on the number of iterations. Table 2.3 gives the NFEs and the CPU time taken by each algorithm to meet the stopping criteria. From this Table we can clearly see the competitive performance of HDE.

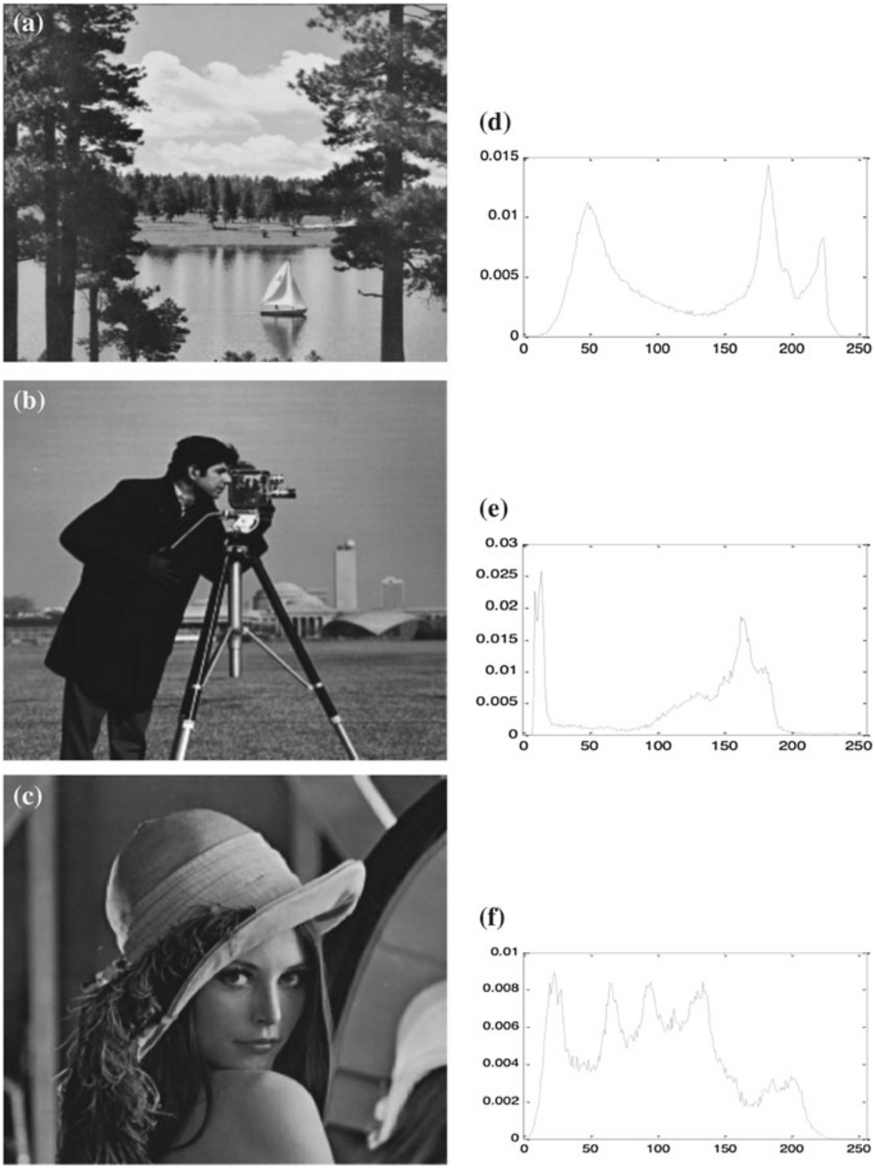


Fig. 2.1 Test images and their normalized histograms. **a** Sailboat, **b** Cameraman, **c** Lena, **d** histogram of Sailboat image, **e** histogram of Cameraman image and **f** histogram of Lena image

Table 2.1 Parameters of HDE. $nc = 3 - D$, where D is the number of segmentation classes (to be fixed by the user)

Parameter	Value
Population size NP	$10 \times N$
Scaling factor F	0.25
Crossover rate C_r	0.20
Maximum iteration	200

Table 2.2 Comparison of HDE with basic DE in terms of CPU time and NFE

Images	No. of Classes	DE		HDE	
		Time	NFE	Time	NFE
Silboat	3	5.9143	17280	5.8968	10110
	4	9.6739	23280	9.2041	1630
Camera man	3	5.8034	16740	5.7876	10320
	4	9.0024	22920	8.8609	16090
	3	5.9675	17010	5.8968	11270
Lena	5	12.934	27750	12.8077	18300
		5			

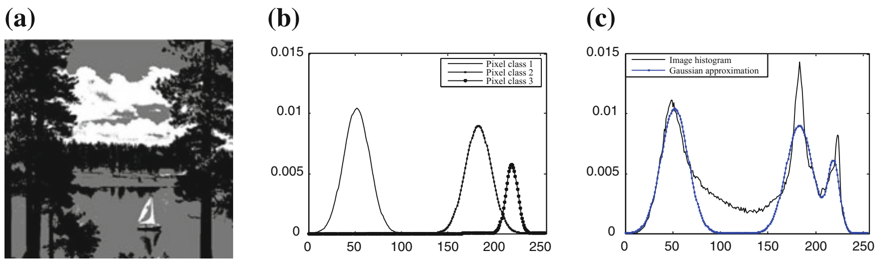


Fig. 2.2 Results of Sailboat image with three classes: **a** segmented image, **b** Gaussian function of each class and **c** original histogram and corresponding Gaussian approximation

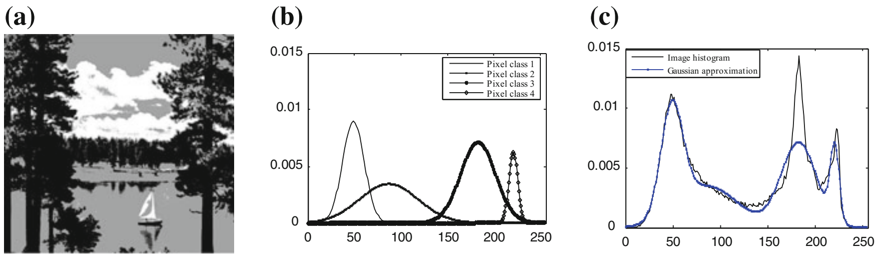


Fig. 2.3 Results of Sailboat image with four classes: **a** segmented image, **b** Gaussian function of each class and **c** original histogram and corresponding Gaussian approximation

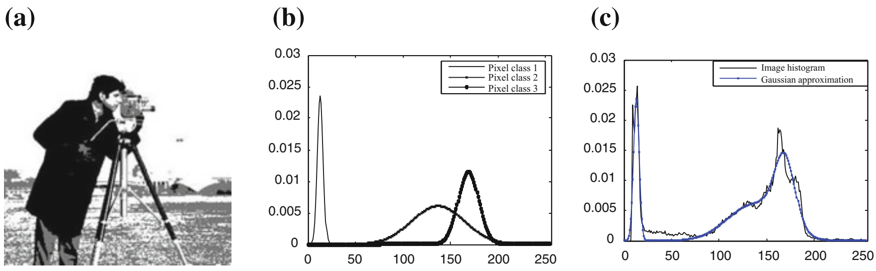


Fig. 2.4 Results of Cameraman image with three classes: **a** segmented image, **b** Gaussian function of each class and **c** original histogram and corresponding Gaussian approximation

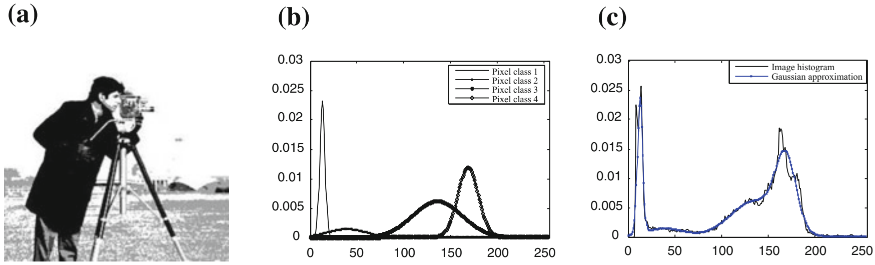


Fig. 2.5 Results of Cameraman image with four classes: **a** segmented image, **b** Gaussian function of each class and **c** original histogram and corresponding Gaussian approximation

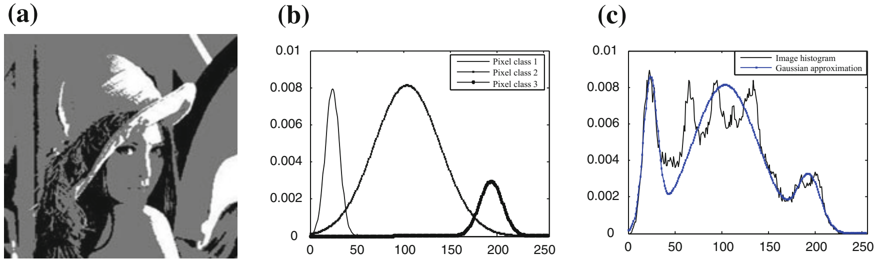


Fig. 2.6 Results of Lena image with three classes: **a** segmented image, **b** Gaussian function of each class and **c** original histogram and corresponding Gaussian approximation

To further quantitatively judge the quality of the algorithm with several other thresholding-based segmentation algorithms [12], the uniformity measure is utilized which has also been extensively used in the literature. This uniformity measure is given by:

$$U = 1 - 2 \times (nc - 1) \times \frac{\sum_{j=0}^{nc-1} \sum_{j \in R_j} (f_i - m_j)^2}{N \times (f_{max} - f_{min})} \tag{2.14}$$

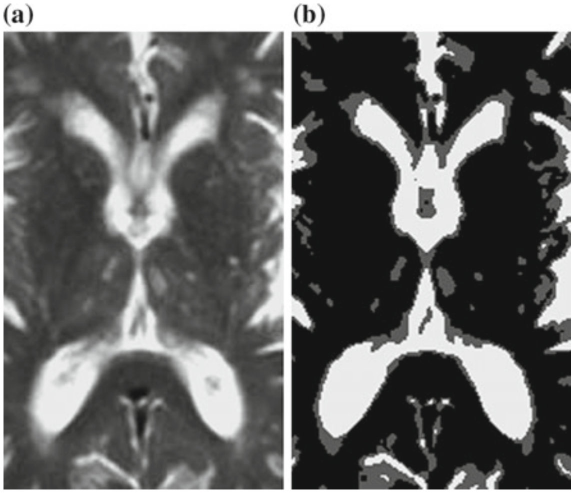


Fig. 2.7 Segmentation of brain MRI of the ventricles. **a** Original slice, **b** Segmented slide into 3 classes, t = (134; 187)

where, nc denotes number of classes, R_j denotes the j^{th} segmented region, f_i indicates the gray level of the pixel i, m_j mean gray level of pixels in j^{th} region, N denotes the total number of thresholds in the given image, f_{max} gives the maximum gray level of pixels in the given image and f_{min} gives minimum gray level of pixels in the given image.

The value of the uniformity measure, U, should be a positive fraction i.e. it should lie between 0 and 1. A higher value of U indicates that there is better uniformity in the thresholded image, depicting better quality of thresholding and vice versa. It can be also seen from Table 2.4 that the proposed HDE algorithm could achieve significantly better segmentation results as demonstrated by its higher values of U in each case, compared to other methods

In order to analyze the obtained results from statistical point of view, we do a Wilcoxon test. Then, the p-value obtained from the results of Table 2.4 is equal to 0.0152 that indicates a significant different between the original DE and the enhanced DE.

2.6 MRI Slices Segmentation

Magnetic resonance imaging (MRI) is a medical imaging technique used in radiology to image the anatomy and the physiological processes of the body in both health and disease. MRI devices or scanners are based on strong magnetic fields, radio waves, and field gradients to form images of the body.

Table 2.3 Results obtained by HDE for images given in Fig. 2.1

Image	Size(in number of pixels)	Number of classes	Parameters of Gaussian approximations	Time (s)	Threshold
Sailboat	512 × 512	3	P(0.0102, 0.0078, 0.0065)	5.8968	117,205
			μ(53, 182, 221)		
			σ(21.8457, 24.9995, 7.0544)		
		4	P(0.0090, 0.0035, 0.0071, 0.0062)	9.2041	95,121,205
			μ(49,87,183,221)		
			σ(16.4485, 44.4340, 26.5073, 6.6560)		
Camerman	256 × 256	3	P(0.0235, 0.0061, 0.0114)	5.7876	33,130
			μ(13, 137, 169)		
			σ(4.3845, 39.2224, 15.4275)		
		4	P(0.0231, 0.0014, 0.0061, 0.0118)	8.8609	30,52,131
			μ(13, 39, 136, 169)		
			σ(3.9712, 29.6954, 37.0420, 15.5644)		
Lena	256 × 256	3	P(0.0079, 0.0081, 0.0029)	5.8968	53,170
			μ(24, 104, 194)		
			σ(10.7663, 49.9796, 17.4342)		
		5	P(0.0084, 0.0070, 0.0066, 0.0074, 0.0032)	12.8077	46, 80, 114,176
			μ(25, 63, 94, 131, 191)		
			σ(12.1269, 15.6615, 16.0053, 23.8994, 22.1594)		

Since its early development in the 1970s and 1980s, MRI has proven to be a highly versatile imaging technique. While MRI is most prominently used in diagnostic medicine and biomedical research, it can also be used to form images of non-living objects. MRIs are able to produce a variety of chemical and physical data, in addition to detailed spatial images.

MRI is widely used in hospitals and clinics for medical diagnosis, staging of disease and follow-up without exposing the body to ionizing radiation. For our work, the data were from the CHU Henri Mondor, Créteil (France).

Table 2.4 Comparison of HDE with basic PSO and GA

Image	No. of classes	Threshold			Uniformity measure		
		PSO	GA	EDE	PSO	GA	EDE
Sailboat	3	96,201	89,210	117,205	0.9632	0.9535	0.9697
	4	90, 115, 208	88, 115, 205	95, 121, 205	0.9664	0.9681	0.9694
Camera man	3	30, 135	30, 142	33, 130	0.9752	0.9744	0.9764
	4	28, 48, 145	28, 50, 145	30, 52, 131	0.9735	0.9732	0.9736
Lena	3	61, 166	53, 178	53, 170	0.9597	0.9490	0.9533
	5	46,84, 119, 186	46, 77, 115, 186	46, 80, 114, 176	0.9774	0.9758	0.9807

MRI has a wide range of applications in medical diagnosis and over 25000 scanners are estimated to be in use worldwide. MRI affects diagnosis and treatment in many specialties although the effect on improved health outcomes is uncertain. Since MRI does not use any ionizing radiation, its use is generally favored in preference to CT when either modality could yield the same information. For all these reasons developing tools for analysis these data is very important.

To illustrate the performance of our segmentation algorithm for the analysis of CT-Scan images. Two examples are presented in Figs. 2.7 and 2.8. In the first example, the region of interest was the ventricles and the goal was to extract the segment the

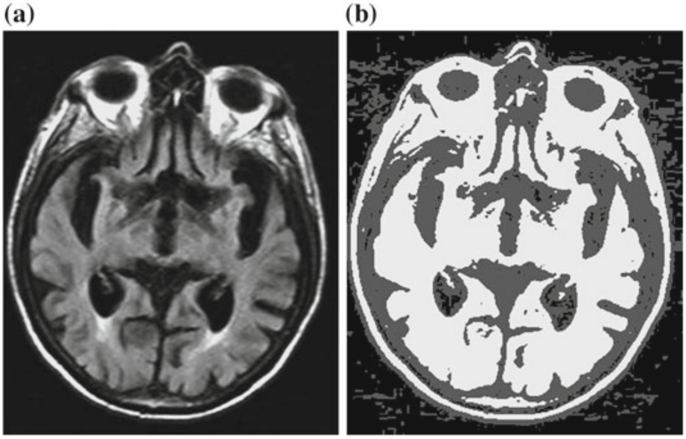


Fig. 2.8 Illustration of the Segmentation of a retinal angiography image where the goal is to extract drusens. **a** Original pathologic Image, **b** Original histogram and its approximation, **c** Segmentation on 2 classes, t = 150

ventricular system represented by high intensity voxels. In this case, the presented results consists in a segmentation on 3 classes to have more accuracy.

The second example presented here

2.7 Conclusions

In this paper, a modified DE algorithm namely HDE, is used for image segmentation. The objects and background components within the image are assumed to fit into Gaussian distributions exhibiting non-equal means and standard deviations. The histogram can thus be approximated by a mix of Gaussian probability functions. The algorithm HDE is used to estimate the parameters for the mixing density function as it seeks to get a minimum error between the density function and the original histogram. Experimental results show that HDE produces satisfactory results, indicating that it can be used for image segmentation in multi-thresholding due to its computational efficiency. Additionally, HDE appears to be effective due to its quality performance. The proposed work can easily be extended in several directions. In the future we intend to perform a formal comparison with other state-of-the-art image segmentation techniques and also we will take a wider range of test images.

References

1. A. Baxsturk, E. Gnay, Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm. Expert Syst. with Appl. **36**(2), 26452650 (2009)
2. A.K. Bhandari, A. Kumar, G.K. Singh, Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms. Expert Syst. with Appl. **42**(22), 8707–8730 (2015)
3. A. Bouaziz, A. Draa, S. Chikhi, Artificial bees for multilevel thresholding of iris images. Swarm Evolut. Comput. **21**, 32–40 (2015)
4. P. Bratley, B.L. Fox, ALGORITHM 659 implementing Sobol's Quasi random sequence generator. ACM Trans. Math.Softw. **14**(1), 88–100 (1988)
5. W.-D. Chang, Parameter identification of Rosslers chaotic system by an evolutionary algorithm. Chaos, Solitons & Fractals **29**(5), 1047–1053 (2006)
6. E. Cuevas, D. Zaldivar, M. Prez-Cisneros, A novel multi-threshold segmentation approach based on differential evolution optimization. Expert Syst. Appl. **37**(7), 265–5271 (2010)
7. S.-K.S. Fan, Y. Lin, A multi-level thresholding approach using a hybrid optimal estimation algorithms. Pattern Recognit. Lett. **28**, 662–669 (2007)
8. R.C. Gonzalez, R.E. Woods, Digital Image Processing (Prentice Hall, Upper Saddle River, 2002). N. Otsu, A threshold selection method for gray-level histogram. IEEE Trans. Syst. Man Cybernet **9**, 62–66 (1979)
9. B. Liu, L. Wang, Y.-H. Jin, D.-X. Huang, F. Tang, Control and synchronization of chaotic systems by differential evolution algorithm. Chaos, Solitons & Fractals **34**(2), 412419 (2007)
10. A. Nakib, H. Oulhadj, P. Siarry, Image histogram thresholding based on multiobjective optimization. Signal Process. **87**, 2516–2534 (2007)
11. A. Nakib, H. Oulhadj, P. Siarry, Non supervised image segmentation based on multiobjective optimization. Pattern Recognit. Lett. **29**, 161–172 (2008)

12. P.D. Sathya, R. Kayalvizhi, Development of a new optimal multilevel thresholding using improved particle swarm optimization algorithm for image segmentation. Int. J. Electron. Eng. **2**(1), 63–67 (2010)
13. M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation. J. Electron. Imaging **13**(1), 146165 (2004)
14. R. Storn, K. Price, Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, Berkeley, CA (1995)
15. R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
16. W. Synder, G. Bilbro, A. Logenthiran, S. Rajala, Optimal thresholding-a new approach. Pattern Recognit. Lett. **11**, 803–810 (1990)
17. J. Vesterstrom, R. Thomsen, A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems, in *Proceedings of the Congress on Evolutionary Computation 2004 (CEC2004)*, vol. 2, Portland, Oregon, 20–23 June 2004, pp. 1980–1987
18. J.S. Weszka, R. Azriel, Histogram modifications for threshold selection. IEEE Trans. Syst. Man Cybern. **9**(1), 38–52 (1979)
19. E. Zahara, S.-K.S. Fan, D.-M. Tsai, Optimal multi-thresholding using a hybrid optimization approach. Pattern Recognit. Lett. **26**, 1082–1095 (2005)

Chapter 3
Fuzzy Edge Detection in Computed Tomography Through Genetic Algorithm Optimization

A.M.T. Gouicem, M. Yahi and A. Taleb-Ahmed

Abstract The ill posedness of the image reconstruction problem requires approached solution as a regularization of a specific criterion, in general, a penalty is imposed on the solution. The challenge is to avoid the smoothing of edges which are very important attributes of the image when it is regularized. The x-ray Tomography is classified as sensing problems for which we do not know the equipment measurement transfer function so it is considered as an ill posed inverse problem. Many studies have been developed to solve this problem, among them the Bayesian inference which aims at smoothing artifact in image. The problem for Bayesian methods is the edge penalization. In this work, we first present a fuzzy inference model for the edge preservation. Under this condition, we show that it is possible to find the best global solution to the problem by introducing genetic algorithm optimization (GA).

3.1 Introduction

In many applications such as Non Destructive Testing (NDT) we seeks a defect. In reality, we can't directly measure a quantity as in our case the shape of the unknown object. The image reconstruction approaches are very ill posed because the projections are in limited numbers. It is therefore essential to get a priori information to regularize the problem. Many studies have been done to solve this problem, such as the Bayesian inference which includes a penalty function to control the propagation of noise and produce a satisfactory reconstruction. This method aims at smoothing artifact in image. The problem for Bayesian methods is the edge penalization, that is to say the choice of the distribution that models the prior information. Many nonlinear distributions were proposed to avoid the edge penalization. We can cite those

A.M.T. Gouicem · M. Yahi
Laboratory of Image and Signal Processing (LISP), Welding and NDT
Research Center CSC, Chéraga, Algeria
e-mail: medthrali@yahoo.fr

A. Taleb-Ahmed (✉)
LAMIH UMR CNRS UVHC 8201, Valenciennes, France
e-mail: taleb@univ-valenciennes.fr

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_3

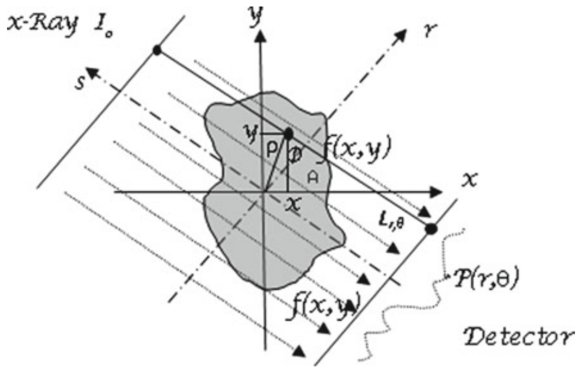


Fig. 3.1 Projection measures in x-ray tomography

from A.M. Djafari [2, 3] where the MAP solution may be found by combining the likelihood function by Gauss or Poisson estimation with the image priori. The MAP algorithm has a disability in the local neighborhood to eliminate the noise and to smooth the reconstructed image. There exist many others works that use the techniques of edge preserving like the mean root prior (MRP) [6] and total variation prior (TV) [10]. The penalty of TV reduces the contrast of the resulting image [6]. Recently, J. Fessler et al. [1] used the weighted least squares (WLS) method,where he developed the WLS algorithm and proved that it converges rapidly and produce images with best quality. For the fuzzy inference in the beginning (1996) F. Russo [10] use the fuzzy model to filter noise in the sensor data. Rajan [7] in 2004 and Gouicem A.M.T et al. [4] used the fuzzy inference system to model a priori information. In this work we assume that the noise smoothing and, the edge detection and preservation are well done with the fuzzy inference. Here we propose to solve the optimization problem using an evolutionary algorithm, so this paper is organized as follows: first, an introduction of inverse problems in tomography-x is presented, then a statistical iterative method and Bayesian inference MAP EM are also presented. After that, our contributions, fuzzy inference and implementation of fuzzy penalty FP EM, also the fuzzy penalty genetic algorithm method FP GA, are described. Finally, a conclusion and work under progress end this paper.

3.1.1 Problem Statement

X-ray computed tomography determines an object function $f(x, y)$ from measures known as projections p (Fig. 3.1).

$$f(x, y) = \int\int p(r, \theta)e(r - xcos\theta - ysin\theta)drd\theta \tag{3.1}$$

where $(r;\theta)$: are the relative coordinates of the object. For the ill posed problem $f^* = \text{ArgMin}{-}\log(J(f/p))$ is the quasi solution with minimization of the criterion $J(f)$ chosen to be likelihood estimation.

$$J(f) = L(f) + P(f) \tag{3.2}$$

where $L(f)$ is a likelihood function $l(f)$.

$$l(f) = \sum_{i=1}^{n}\left(-\sum_{j=1}^{m} a_{ij} f_j + p_i ln\left(\sum_{j=1}^{m}(a_{ij} f_j) - ln(p_i!)\right)\right) \tag{3.3}$$

$P(f)$ is the priori information, the most used prior is the GIBBS distribution.

$$P(f) = C^{-\beta U(f)} \tag{3.4}$$

$U(f)$ is the potential function defined in Markov field, and C chosen to modulate the importance of the prior value.

$$U(f) = |f - f_0| \Rightarrow \frac{\partial}{\partial f_j^k} U(f_j^k) = \sum_{b\in N} w_{jb}(f_j^k - f_b^k) \tag{3.5}$$

k iteration number, w_N weight of pixel j_N the nearest neighbor set of the pixel, f_0 the information image. So the final form of the criteria $J(f)$ in the penalized-likelihood estimation called maximum a posteriori expectation maximization (MAPEM) method is:

$$J(f) = \sum_{i=1}^{n}\left(-\sum_{j=1}^{m} a_{ij} f_j + p_i ln\left(\sum_{j=1}^{m} a_{ij} f_j\right) - ln(p_i!)\right) - \beta U(f) + k \tag{3.6}$$

$J(f)$ must be optimized to reach the best solution of the inverse problem.

3.2 Applied Methods

Edge detection is used to identify different objects that make up the image scene. There are many ways to find the edge of objects; most are based on the first and second derivatives of the image (Fig. 3.2).

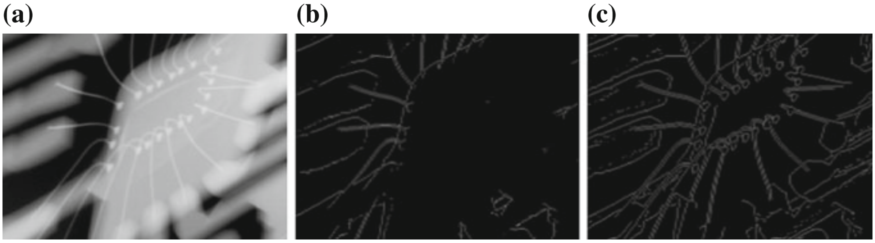


Fig. 3.2 Edge Detection in Image, **a** original image, **b** edge with the first derivative, **c** edge with the second derivative

3.2.1 Implementation of Fuzzy Inference

First we express the Value of the prior term in MAP criterion with fuzzy penalized model.

$$U(f) = G_N = |f_{i,j-1} - f_{i,j}| \tag{3.7}$$

U(f) is the prior law of penalty, which is chosen like a fuzzy distribution defined as the first derivative of the pixel value (i, j). Where $f_{i,j-1}$ represents the neighborhood pixel in the direction N(North). To compute an edge E along a direction, three derivatives G (which are perpendicular to this direction in the 3*3 windows) must be calculated and summed.

$$E_W^k(i, j) = G_W^k(i, j) + G_W^k(i-1, j) + G_W^k(i+1, j) \tag{3.8}$$

where k designs the iteration number and W designs the west direction. To express the degree of smallest of the fuzzy derivative E_N in one direction, we use the fuzzy sets small instead of the hard computations. Now we must find the fuzzy model of the Gibbs prior function, in two steps edge detection and penalization.

3.2.1.1 Detection Steps

If the difference G_N between the pixel gray levels is large, then there is an edge, else the region is smooth and the edge value E_N is small. After application of conditions to edge detection in Fuzzification step, we obtain the following rules base [4]:

If $G_W^k(i, j)$ is small and $G_W^k(i-1, j)$ is small then $E_W^k(i, j)$ is small
If $G_W^k(i, j)$ is small and $G_W^k(i+1, j)$ is small then $E_W^k(i, j)$ is small
If $G_W^k(i-1, j)$ is small and $G_W^k(i+1, j)$ is small then $E_W^k(i, j)$ is small
If $G_W^k(i, j)$ is large and $G_W^k(i+1, j)$ is large and $G_W^k(i-1, j)$ is large then $E_W^k(i, j)$ is large

3.2.1.2 Penalization Steps

The second step is the penalization of pixels for which we havent detected an edge (smoothing region). Eight fuzzy rules of the FIS penalty-w are used to indicate the correction in penalization of the eight directions.

If $E_W^k(i, j)$ is small then $C^k(i, j) = E_W^k(i, j)$ else $C^k(i, j) = 0$

$C^k(i, j)\overrightarrow{N}$ is the correction at site (i, j) due to the adjacent pixel in the direction N. The total fuzzy correction C_T is:

$$C_T^k(i, j) = \frac{1}{8}\sum_N C^k(i, j)\overrightarrow{N} \tag{3.9}$$

Replacing the error term (3.9) in (3.6) we obtain the fuzzy criterion to optimize by GA to solve the inverse problem.

3.2.2 Genetic Algorithm (GA)

The GA is inspired from the natural selection mechanism. Where the potential solution to a problem is an individual who can be coded by a set of genes. Then, it can be structured with a string of values in binary or real format. Positive values are used to reflect the degree of fitness of the chromosome to solve the optimization problem, and this value is really related to its objective value. In genetic evolution, we seeks to produce a chromosome with a good quality, which means a good solution to the problem. In application of the GA, a population of chromosomes must be initialized. The population size changes from a problem to another. The parent chromosomes are selected through a specific selection routine. The genes of the parents are mixed and recombined to produce new chromosomes for the next generation. Good chromosomes produce a generation with a high chance of survive, simulating the natural survival mechanism. One of the best techniques used in selection is the wheel roulette of selection [5]. To facilitate the evolutionary cycle of the GA, two basic operators are created: crossover and mutation, also the wheel roulette of selection can be taken as another kind of operator.

3.2.2.1 Encoding

The real encoding is widely used, especially in optimization of real variables problems [8]. Each variable represents a gene and all the values that can take this feature represent the possible alleles for this gene, and concatenating all these genes to obtain a chromosome representing a solution in its entirety. There are three main types of usable encoding, and you can switch from one to another with relative ease.

The Binary Encoding

It is the most used, each gene has the same binary alphabet {0, 1}, A gene is then represented by a long integer (32 bits). Chromosomes which are sequences of genes are represented by an array of genes and the individuals of our search space are represented by chromosomes arrays.

The Real Encoding

This can be useful especially when you search for the maximum of a real function.

Gray Encoding

Gray encoding is a coding which has the property: between an element n and an element $n + 1$, so neighbor in the search space, a single bit changes.

Algorithm 3.1: Genetic Algorithm

```
Initialize a population of size $P$
Evaluate the $P$ individuals
while Stopping Conditions are not satisfied do
    $P'$ = Parents Selection in $P$
    $P'$ = Crossover Operator Applicated on $P'$
    $P'$ = Mutation Operator Applicated on $P'$
    $P$ = Replace the older of $P$ by the Descendants in $P'$
    Evaluate $P$
end
```

3.2.2.2 Selection Operator

The selection operator define which individuals in P will be duplicated in the new population P' and will serve as parents [9]. This operator is the most important since it allows individuals in a population to survive, to reproduce or die. The probability of survival of an individual will be directly related to its efficacy relative to the population. Wheel Roulette is the most popular method. With this method every individual has a chance of being selected proportional to its performance, so more people have adapted to the problem, the more likely they are to be selected. In this method, each individual is assigned a sector whose angle is proportional to his adaptation (fitness). We turn the wheel and when it stop rotating it select the individual.

3.2.2.3 The Crossover Operator

The crossover is a computer transposition of the natural mechanism, this one is the production of chromosomes which partially inherit the characteristics of the parents.Its fundamental role is to allow the recombination of information in the gene pool of the population. This operator is applied after applying the selection operator P on the population; we end up with a population P'. It may be noted that the number of crossover points and crossover probability pc allow the introduction of more or less diversity. To simplify the process we choose a crossover mechanism with a random point.

3.2.2.4 Mutation Operator

This operator is to change the value of an allele with a very low probability pm. A mutation is simply the inversion of a bit. The mutation operator randomly changes the characteristics of a solution, which allows introducing the diversity of our population of solutions. This operator introduces "noise" in the population.

3.2.2.5 Selection (Fitness) Function

The selection function is used to create an evolutionary pressure. Chromosomes with high performance have a great chance of surviving. The method of selection of the lottery wheel is used to select the chromosomes from operators. A chromosome can be selected well than another. Everything depends on luck in the roulette wheel, which is adaptive.

3.2.3 Optimization by Genetic Algorithm

Our contribution takes the prior information as a fuzzy penalty and optimizes the new criteria by the genetic algorithm, which is based on three steps reproduction, evaluation and selection [4].

Initialization

An initial population is generated randomly with N chromosome. The best initial population is the FBP (filtered back projection).

Coding

Each chromosome is decoded (coding elements of the population) to give the phenotype. Actually the real codes are now widely used, especially in the application areas for optimizing real variables problems.

Objective Function

This phenotype is used to evaluate the corresponding genotype by calculating the objective function (cost) in R.

Selection

With these values a list of people likely to reproduce is established (new generation) by a selection operator. The selection scheme used is that of the wheel of the lottery (RWS).

Reproduction

Requires an inherited memory as genes. The genetic algorithm then diversify the population i.e. improve the objective function of the population by reproduction.

Crossover

The Conventional crossover involves the exchange of genes between each parent together (arithmetic crossover). We choose a crossover mechanism with random point. It is applied with a high probability Pc.

Mutation

In our application a single chromosome is selected from each operation (uniform mutation). It is applied with a probability Pm.

Fitness evaluation

The fitness function J chosen for maximization is defined in Eq. 3.1 from which we obtain:

$$J(f) = \sum_{i=1}^{n} \left(-\sum_{j=1}^{m} a_{ij} f_j + p_i ln \left(\sum_{j=1}^{m} (a_{ij} f_j) - ln(p_i!) \right) \right) - \beta U(f) + k \tag{3.10}$$

with:

$$frac\partial\partial f_j^k U(f_j^k) = C_T(i, j) \Rightarrow \beta U(f_j^k) = f_j C_T(i, j) \tag{3.11}$$

In (3.10) we express $U(f)$ in function of CT Value of prior term in MAP [2] criteria. We will express it with fuzzy model in the FP GA method by taking $J(f)$ as a fitness function to optimize. In order to avoids the local minimum and converged to the global solution.

3.3 Results and Discussion

Data resulting from our Tomography Fein-Focus at the research center in welding and NDT, are simple and degraded first by a Gaussian noise level with 5 added value Fig. 3.3, then a Speckle product noise with 4 value. Images have been reconstructed

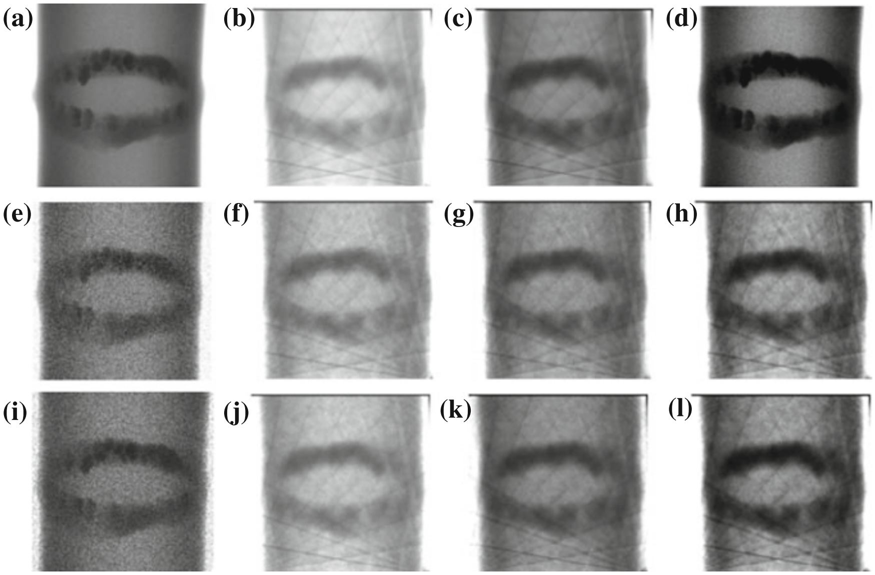


Fig. 3.3 Welding joint Image reconstructed with different algorithms, **a**, **e**, **i** Original image respectively without noise and with Gaussian, Speckle noise **b**, **f**, **j** MAPEM Method **c**, **g**, **k** FPEM Method and **d**, **h**, **l** FPGA Method

with different algorithms: the maximum a posteriori-expectation maximization algorithm MAP EM [2], the Maximum a Posterior reconstruction algorithm with fuzzy Prior FP EM and the FP GA [3]. We measured the convergence rate by computing the peak signal to noise ratio PSNR between the simulated noiseless activity distribution and the image estimated as a function of the iteration number k. This item expresses the dispersion between the reconstructed image and the original image. The PSNR resulting for the MAP EM, FP EM and FP GA are in Table 3.1 and, are plotted in Fig. 3.4. To show the efficiency to use the GA in image reconstruction you can see the increases of the resulting PSNR in case of the FP GA method for the two kinds of images with and without noise. The image size is 256 × 256; we can see the optimization given by the combined algorithm. We note that the CPU characteristics are an I7 with 3.4 GHz and 16 Go of RAM. The number of iterations choice is comprised between 10 and 100 to remove the accumulation of noise. The parameter N number of projections is equal to 18, the size of image number of pixel $N = 256 \times 256$. The parameters for dimensioning the genetic problem are population size (20), generation number (100), crossover probability (90) and mutation probability (10).

Table 3.1 PSNR Simulation quantitative results after N iterations for Welding joint image

Iteration		10	20	30	40	50	60	70	80	90	100
MAPEM	None	5.05	7.86	7.49	9.35	10.25	11.57	12.01	12.1	13.16	14.09
	Gauss	6.35	7.58	8.09	9.49	10.23	10.96	11.06	11.92	12.12	13.07
	Speckle	6.05	6.74	7.09	7.74	8.23	8.876	9.063	9.920	10.125	11.09
FPEM	None	6.70	7.92	8.52	9.72	10.96	11.83	12.58	12.66	13.73	14.25
	Gauss	6.56	7.52	8.96	9.62	10.26	11.96	12.42	12.96	13.13	14.05
	Speckle	6.21	7.35	8.41	9.70	10.74	11.32	12.08	12.52	13.85	14.05
FPGA	None	7.96	8.97	9.12	10.25	11.7	12.25	13.85	14.12	15.38	16.75
	Gauss	6.90	7.84	8.85	10.95	11.9	12.32	13.63	14.85	15.74	16.85
	Speckle	6.86	7.7	8.92	9.4	10.9	11.6	13.5	14.7	15.3	16.2

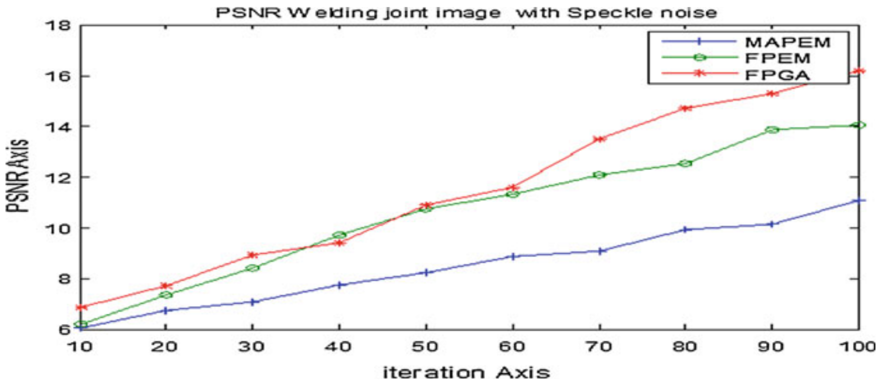


Fig. 3.4 PSNR for MAPEM, FP EM and FP GA for Welding joint image with Speckle noise

3.4 Conclusion

The new iterative algorithm exploits the fuzzy reasoning to regularize the penalty in terms of noise in the image, which accumulates for regularized iterative methods (ML-EM, OS-EM ...) when the number of iterations increases. Introducing genetic optimization gives a global solution and avoids blocking around local solutions to the inverse problem. Our combined method minimizes the computational cost and reduces the required memory capacity. The algorithm eliminates noise and preserves the image edges. The algorithm minimizes the number of projections needed for tomographic reconstruction. This reduces the number of practices by X-ray bombardment. Results from the fusion of these two intelligent disciplines are encouraging especially when the number and angles of the projections are limited.

References

1. S.Y. Chun, Y.K. Dewaraja, J.A. Fessler, Alternating direction method of multiplier for tomography with non-local regularizers. IEEE Trans. Med. Imag **33**(10), 1960–8 (2014)
2. A.M. Djafari, Joint Estimation of parameters and hyperparamaters in a Bayesian approach of solving inverse problems, in *Proceedings of the International Conference on Image Processing*, vol. II (Lausane, Suisse, 1996), pp. 473–477
3. A.M. Djafari, *A Full Bayesian Approach for Inverse Problems* (Kluwer Acadimic Publishers, New Mexico, 1996), pp. 135–143
4. A.M.T. Gouicem, K. Benmahammed, R. Drai, M. Yahi, A. Taleb-Ahmed, Multi-objective G-a optimization of fuzzy penalty for image reconstruction from projections in x-ray tomography. Digit. Signal Process. Elsevier **22**(3), 486–596 (2012)
5. J.H. Holland, Outline for logical theory of adaptive systems. J. ACM **03**, 297–314 (1962)
6. J. Ma, Q. Feng, Y. Feng, J. Huang, W. Chen, Generalized Gibbs priors based positron emission tomography reconstruction. Comput. Biol. Med. **40**, 565–571 (2010). Elsevier
7. P.P. Mondal, K. Rajan, Iterative image reconstruction for emission tomography using fuzzy potential. IEEE Trans. Image Signal Process. (2005)
8. S.A. Qureshi, S.M. Mirza, M. Arif, Determination of optimal number of projections for parallel-ray transmission tomography using hybrid continuous genetic algorithm, in textit International Journal of Imaging Systems and Technology, in Process (2006)
9. S.N. Sivanandam, S.N. Deepa, *Introduction to Genetic Algorithms* (Springer, Berlin, 2008). ISBN 978-3-540-73189-4
10. D.V. Ville, M. Nachtegael, D.V. Weken, E.E. Kerre, W. Philips, I. Lemahieu, Noise reduction by fuzzy image filtering. IEEE Trans. Fuzzy Syst. **11**, 429–435 (2003)

Chapter 4
Particle Swarm Optimization Based Fast Chan-Vese Algorithm for Medical Image Segmentation

Devraj Mandal, Amitava Chatterjee and Madhubanti Maitra

4.1 Introduction

Image segmentation is a very important part of image pre-processing and its application towards computer vision. The basic objective of image segmentation is to extract the constituent objects of interest from a given image, which can be used for further processing on them. Several common methods such as edge-detection, thresholding, histogram-based methods, region-based methods, region-growing or split-merge methods have already been developed for this purpose. A novel idea to detect objects in an image is by the use of active contours [3, 4, 35].

In this methodology, an initial active contour [3, 4, 35] is drawn on the image and a fitness functional energy [6] is associated with it. The initial contour is evolved by expanding or contracting it around the desired object of interest. The driving energy that moves the initial contour is either edge-dependent [3, 37, 38, 49, 50] or region-dependent [4, 51, 59, 61, 63, 64]. For edge-based models, the contour is attracted towards the strongly-defined edges present in the image. This model however fails to detect objects demarcated by weakly-defined or smoothened edges [3, 46, 47]. Region-based models, on the other hand, use specific region descriptors to drive the curve towards the objects and are even quite successful in detecting objects defined by weak edges.

Mathematically, the representation and evolution of active contours [35] can be carried out in different ways. The classical approach [35] favours the parametric representation of the curves and the evolution of the points in the curve in order to minimize the energy associated with it. However, this formulation [35] fails to detect interior contours within an image and cannot deal with cusps, shocks, triple junctions or any such topological changes.

D. Mandal · A. Chatterjee (✉) · M. Maitra
Electrical Engineering Department, Jadavpur University, Kolkata 700032, India
e-mail: cha_ami@yahoo.co.in

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology,*
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_4

A new formulation developed independently by Tsai [62] and Chan and Vese [6, 60], instead utilizes the mathematical tool of level sets [50] to represent and evolve the curve. In this methodology, the evolving curve is embedded into a higher dimension, known as level sets [50]. The driving energy moves this level set and the zero-level of this level set is known as the evolving contour. The numerous advantages of this method include the ability to detect interior contours within the image, handle topological changes such as breaks or merging, and the ease with which it can be extended to higher dimensions.

The work carried out by Chan and Vese (popularly called the C-V model) [6, 60] uses level sets [50] to reformulate the Mumford-Shah energy [49] into a new region-based energy fitting functional to segment images for both the piecewise-constant [6] (for homogeneous images) and the piecewise-smooth [60] (for images suffering from intensity inhomogeneity) cases. The energy functional associated with the C-V model detects the desired objects within an image by minimizing the dissimilarity within each region. When the energy becomes minimum, the evolved curve i.e., the zero-level curve of the level set, sits exactly on the boundary of the desired objects.

The C-V model suffers from its own set of limitations in spite of the abundant advantages it provides in the field of image segmentation. The energy fitting functional is non-convex and non-unique in nature and may have several local minimum points [2, 6]. The level set framework is implemented by solving a partial differential equation (PDE), using the gradient descent search method [6, 60]. Although the gradient descent based adaptation procedure is very easy to implement, as it only involves the first order partial derivatives of the fitness energy functional, it suffers from the problem of getting trapped at the first local minimum it encounters. Hence it is of paramount importance in C-V model that suitable initialization of an active contour [2] is a requisite criterion to achieve the desired, accurate segmentation result. In addition, the C-V model comprises various curve evolution parameters that need to be properly tweaked to achieve satisfactory results.

In this work, we propose a new formulation of the C-V model, where we search for particular intensity values on the basis of which the image can be divided into different classes based on the C-V energy fitting functional. Although the method reported in [40] using discrete level sets employed a somewhat similar philosophy, however our method has solved this problem in a novel manner, by employing different swarm intelligence based optimization procedures, without utilizing level sets at all.

The swarm intelligence and evolutionary optimization based algorithms model the problem at hand by randomly generating some candidate solutions and letting them interact with each other locally as well as with the environment, in search for better solutions to the problem. The interaction and movement of these candidate solutions is often inspired by naturally occurring biological phenomenon such as cross over and mutation of genes, as in Genetic Algorithm [33], flocking of birds in search of food, as in Particle Swarm Optimization (PSO) [36, 49], or swimming and tumbling of bacteria through a nutrient medium, as in Bacterial Foraging Algorithm [41], etc. These kinds of evolutionary techniques are often useful in solving non-gradient type optimization algorithms and have found major use in various problems in image segmentation, image compression, image retrieval, image classification, etc. [10, 15, 16, 45, 52]. The present work has shown how such an algorithm can be used to solve the C-V model with reduced computational time and we have used PSO as a candidate swarm intelligence technique for solving such an algorithm.

The outline of this work is as follows: In Sect. 4.2 we describe the C-V model for the piecewise-constant case in detail, for both the scalar and the vector-valued cases, including its multi-phase level set extension. A detailed account of the advantages and disadvantages of this model is also included. Section 4.3 presents the evolutionary formulation of our proposed fast C-V model. In Sect. 4.4 we describe the basic PSO algorithm and a few of its variants. Our proposed PSO based Fast C-V model for both scalar and vector-valued images is also presented here. In Sect. 4.5, we implement our model for various grayscale and colour medical images and perform a detailed performance analysis. Section 4.6 presents the conclusions.

4.2 The C-V Model (Piecewise-Constant Model) for Image Segmentation

The piecewise-constant model for image segmentation developed by Chan and Vese [43] implements the evolving active contour in a level set framework.

The evolving curve C in Ω, is defined as the boundary of an open subset ω of Ω (i.e. $\omega \subset \Omega$, and C = $\partial\omega$), where Ω is the total image domain. Then the regions ω and Ω/ϖ are defined by the area inside and outside the curve C [6], respectively. The image (u_0)to be segmented, is assumed to consist of two regions of piecewise-constant intensities u_0^i (object) and u_0^0 (background) with the object having a boundary or a bounding contour C_0.
Thus the C-V energy fitting term can be defined as [6]:

$$\begin{aligned} F(c_1, c_2, C) &= v.Length(C) + \mu.Area(inside(C)) \\ + \quad & \lambda_1 \int_{inside(c)} \mid u_0(x, y) - c_1 \mid^2 \mathrm{d}x\mathrm{d}y \\ + \quad & \lambda_2 \int_{inside(c)} \mid u_0(x, y) - c_2 \mid^2 \mathrm{d}x\mathrm{d}y. \end{aligned} \tag{4.1}$$

where, C is any curve that is being iteratively evolved, and the constants c_1 and c_2 [6] denote the average intensities of the image inside and outside the evolving curve, C respectively. The free parameters in Eq. (4.1) must all be positive. The penalizing terms in Eq. (4.1) such as $Length(C)$ $Area(C)$ and force the evolving active contour C to be as smooth as possible. The minimization of energy in Eq. (4.1) occurs when, $C \approx C_0$ i.e., the evolving contour sits exactly on the object boundary.

The above functional in Eq. (4.1) is quite similar to the Mumford-Shah Functional [65]. The Mumford-Shah Functional (F^{MS}) tries to segment an image into its various sub-regions by using region-based information. The C-V model tries to represent each region or connected component R_i of Ω/C using a constant intensity c_i. This c_i represents the average intensity of the region i.e. c_i = average(u_0) on each connected component R_i [6, 49]. This reduced case is called the minimal partition problem. Here the values of the constants c_1 and c_2 [6] are:

$$\begin{aligned} c_1 &= mean(inside(C)) \\ c_2 &= mean(outside(C)). \end{aligned} \tag{4.2}$$

4.2.1 Level Set Formulation of the Model

Level set methods [6, 50], a contour $C \subset \Omega$ is represented by the zero level set of a Lipschitz function $\Phi\ : \Omega \rightarrow R$. This is also called a level set function and it is defined in such a way that

$$\begin{cases} C \quad = \partial\omega \quad \{(x, y) \in \Omega : \Phi(x, y) = 0\} \\ inside(C) \quad = \omega \quad = \{(x, y) \in \Omega : \Phi(x, y) > 0\} \\ outside(C) = \Omega/\varpi \quad = \{(x, y) \in \Omega : (x, y) < 0\} \end{cases} \tag{4.3}$$

Using the level set function Φ and also the Heaviside function H and the one-dimensional Dirac measure δ_0 so as to use one-dimensional calculations, the energy fitting terms $F(c_1, c_2, C)$ in Eq. (4.1) can be reformulated as [6]:

$$\begin{aligned} F(c_1, c_2, \Phi) = \ & v.\int_\Omega \partial(\Phi(x, y))|\nabla\Phi(x, y)|\mathrm{d}x\mathrm{d}y + \mu.\int_\Omega H(\Phi(x, y))\mathrm{d}x\mathrm{d}y \\ & + \lambda_1 \int_\Omega |u_0(x, y) - c_1|^2 H(\Phi(x, y))\mathrm{d}x\mathrm{d}y \\ & + \lambda_2 \int_\Omega |u_0(x, y) - c_2|^2 (1 - H(\Phi(x, y)))\mathrm{d}x\mathrm{d}y \end{aligned} \tag{4.4}$$

Keeping the level set function Φ fixed, and minimizing the energy functional $F(c_1, c_2, \Phi)$ in Eq. (4.4) with respect to c_1 and c_2, the values of the constants c_1 and c_2 in terms of Φ is derived as, [6]:

$$c_1(\Phi) = \frac{\int_\Omega u_0(x, y)H(\Phi(x, y))\mathrm{d}x\mathrm{d}y}{\int_\Omega H(\Phi(x, y))\mathrm{d}x\mathrm{d}y} \tag{4.5}$$

$$c_2(\Phi) = \frac{\int_\Omega u_0(x, y)(1 - H(\Phi(x, y)))\mathrm{d}x\mathrm{d}y}{\int_\Omega (1 - H(\Phi(x, y)))\mathrm{d}x\mathrm{d}y} \tag{4.6}$$

Keeping the constants c_1, c_2 fixed, and minimizing F with respect to Φ, the associated Euler–Lagrange equation for is deduced. To do so, regularized (smooth) versions of the Heaviside function H and the one-dimensional Dirac function δ_0, called H_ε and δ_ε [43] respectively, as given in Eq. (4.7), are required to be used. This finally gives rise to the partial differential equations (PDE) in Eqs. (4.8)–(4.10), that need to be solved iteratively for the progressive evolution of the active contour C.

$$H_\varepsilon(x) = \frac{1}{2}[1 + \frac{2}{\Pi}arctan(\frac{x}{\varepsilon})]\ \delta_\varepsilon(x) = H_\varepsilon^{'}(x) = \frac{1}{\Pi}\frac{\varepsilon}{\varepsilon^2 + x^2} \tag{4.7}$$

$$\frac{\partial \Phi}{\partial t} = \delta_{\varepsilon}(\Phi)[v.div(\frac{\nabla \Phi}{|\nabla \Phi|}) - \mu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2] = 0 \tag{4.8}$$

$$\Phi(0, x, y) = \Phi_0(x, y) \; in \; \Omega \tag{4.9}$$

$$\frac{\delta_{\varepsilon(\Phi)}}{|\nabla \Phi|} . \frac{\partial \Phi}{\partial \overrightarrow{n}} \; on \; \partial \Omega \tag{4.10}$$

One of the fundamental features of level sets is that, they can develop shocks and progressively degrade with time. To rectify this, periodic re-initialization of the zero level curve of Φ to the signed distance function is suggested in [3, 12, 18]. This step involves computing the following Eq. (4.11) periodically so as to re-initialize the level set Φ, where $sign(\Phi)$ is the sign function.

$$\frac{\partial \Phi}{\partial t} = sign(\Phi)(1 - |\nabla \Phi|) \tag{4.11}$$

Studies in [3] have however shown that this step of re-initialization introduces further complications like moving the zero level set away from its original location or increasing the computation time. In addition, this step is often applied in an ad-hoc manner without any proper rationale for its implementation [3].

4.2.2 The C-V Algorithm

The steps of the basic C-V algorithm [43] can be summarized as given in Fig. 4.1.

4.2.3 The Strengths and Drawbacks of the C-V Algorithm

The level set implementation of the C-V model enables it to detect interior contours within the image [6]. As already pointed out in [2, 55], due to the non-unique and non-convex nature of the energy fitting functional, suitable choice of the initial contour is a mandatory requirement. Poor choices often lead to the functional getting stuck at a local minimum or being unable to converge towards the global minimum within the desired number of iterations. In fact, suitable initial position of the contour is a necessity for the gradient descent based adaptation procedure to achieve the final segmented result quickly, within a small number of iterations, and, for fast convergence towards the global minimum.

The free parameters λ_1, λ_2, v, μ, involved in the energy fitting functional of the C-V model in Eq. (4.1) are also needed to be properly tuned to drive the energy to its global minimum. Manual tuning, often a brute-force method, is an ill-advised technique in practice due to its large time-complexity and high computational power requirements.

```
C-V Level set
Data: Initialize the iteration number n=0.
Data: Construct the initial level set function $\Phi_0$
if n=0 then
    Compute $c_1(\Phi)^0$ and $c_2(\Phi)^0$ using eqns. (5) and (6).
    Set n=n+1
    Compute the new level set $\Phi^n$ by solving the PDE using eqns. (8), (9) and (10).
    Set regularized level set function $\Phi_r^n$ as $\Phi_r^n = \Phi^n$ .
end
while $((\Phi_r^n - \Phi^{n-1}) \neq 0)$ do
    Compute $c_1(\Phi)^n$ and $c_2(\Phi)^n$ using eqns. (5) and (6).
    Set n=n+1 .
    Compute the new level set by solving the PDE using eqns. (8), (9) and (10).
    if $(|\nabla(\Phi^n)|) \neq 0$ then
        Reinitialize level set function $\Phi^n$ to $\Phi_r^n$ as:
        $\frac{\partial \Phi_r^n}{\partial t} = sign(\Phi^n)(1 - |\nabla \Phi^n|)$
    end
    else
        Set $\Phi_r^n = \Phi^n$
    end
end
Stop contour evolution and report the segmentation result.
```

Fig. 4.1 The C-V algorithm for image segmentation using level set approach

The re-initialization step of the C-V model also introduces several extra complications that may give rise to erroneous segmentation results. A better approach is discussed in [39], where the authors have added an extra regularization term to force the level set to be as close to the signed distance function as possible. The metric in Eq. (4.12) can be added to the energy fitting term in Eq. (4.1), to achieve such a desired effect [39]. By adding this term, now Eq. (4.14) can be solved iteratively without the need for any re-initialization of level set function. The mathematical formulation can then be described as [39]:

$$p(\Phi) = \int \frac{1}{2}(|\nabla \Phi| - 1)^2 \mathrm{d}x\mathrm{d}y \tag{4.12}$$

$$E(\Phi, c_1, c_2) = F(c_1, c_2, \Phi) + P(\Phi) \tag{4.13}$$

$$\frac{\partial \Phi}{\partial t} = \left(\nabla^2 \Phi - div\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right)\right) + \delta_\varepsilon(\Phi)\left[v.div\left(\frac{\nabla \Phi}{\nabla \Phi}\right) - \mu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\right]. \tag{4.14}$$

4.2.4 Extension of C-V Model for Vector-Valued Images

For segmentation of vector-valued images (such as RGB or multi-spectral images), Chan and Vese simply extended their scalar model [5] by defining a fitness functional energy that minimizes the fitting energy over each channels of the vector-valued images. This model also has very strong de-noising capabilities.

Let $u_{0,i}$ be the i^{th} channel of an image on Ω, with i = 1,2,3,...,N channels and C be the evolving curve. The constant vectors are defined as $c^+ = (c_1^+, c_2^+, \ldots, c_N^+)$ and $c^- = (c_1^-, c_2^-, \ldots, c_N^-)$ [5]. Then, similar to Eq. (4.1), for the vector-valued case, the energy functional can be given as [17]:

$$\begin{aligned} F(c^+, c^-, C) &= v.Length(C) + \mu.Area(inside(C)) \\ &+ \int_{inside(c)} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^+ |u_{0,i}(x, y) - c_i^+|^2 \mathrm{d}x\mathrm{d}y \\ &+ \int_{outside(c)} \frac{1}{N} \sum_{i=1}^{N} \lambda_i^- |u_{0,i}(x, y) - c_i^-|^2 \mathrm{d}x\mathrm{d}y \end{aligned} \tag{4.15}$$

where λ_i^+ and λ_i^- are the parameters for the i^{th} channel and must be positive.

The constant c^+ and c^- vectors represent the average image intensities, inside and outside the evolving contour C, for each channel of the vector-valued image. Similar to the scalar case, it is prudent to use level set formulation for minimizing the above functional for the vector valued case too. Also as given before, regularized (smooth) version of the Heaviside function H_ε and the one-dimensional Dirac function δ_ε are used in the formulation.

Minimizing the energy in Eq. (4.15) with respect to the constants c^+ and c^-, for i = 1,2,3,…,N, we obtain [5]:

$$\begin{aligned} c_i^+(\Phi) &= \frac{\int_\Omega u_{0,i}(x, y) H(\Phi(x, y)) \mathrm{d}x\mathrm{d}y}{\int_\Omega H(\Phi(x, y)) \mathrm{d}x\mathrm{d}y} \\ c_i^-(\Phi) &= \frac{\int_\Omega u_{0,i}(x, y)(1 - H(\Phi(x, y))) \mathrm{d}x\mathrm{d}y}{\int_\Omega (1 - H(\Phi(x, y))) \mathrm{d}x\mathrm{d}y} \end{aligned} \tag{4.16}$$

The extra regularizing term in Eq. (4.12) is added as in the scalar case to Eq. (4.15) to avoid the necessity of the additional re-initialization phase. Similarly, the energy functional in Eq. (4.15) can be minimized by the gradient descent search, with respect to Φ by keeping the vectors c^+ and c^- constant. The following Euler–Lagrange equation has been derived that needs to be solved iteratively so as to evolve the curve [5]:

$$\begin{aligned} \frac{\partial \Phi}{\partial t} &= \left(\nabla^2 \Phi - div\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right)\right) \\ &+ \delta_\varepsilon(\Phi)\left[v div\left(\frac{\nabla \Phi}{\nabla \Phi}\right) - \mu - \frac{1}{N} \sum_{i=1}^{N} \lambda_i^+ |u_{0,i}(x, y) - c_i^+|^2 + \frac{1}{N} \sum_{i=1}^{N} \lambda_i^- |u_{0,i}(x, y) - c_i^-|^2\right]. \end{aligned} \tag{4.17}$$

4.2.5 *Extension of C-V Model for Multi-phase Level Set Implementation*

One of the advantages of level sets is the ease with which it can be extended to higher dimensions. The C-V model likewise in its level set implementation can be extended to cater to multi-phase image segmentation and is able to detect more than two regions within an image and also to deal with triple junctions [68]. For the multi-phase level set implementation, the number of level sets required is directly proportional to the number of regions into which the image has to be segmented. For segmenting an image into a maximum of 2^n number of regions, number of level sets is sufficient. The formulations for the four-phase and three-phase level set implementations are presented below.

The four-phase implementation can be stated as: Let us assume that the level sets $\varphi = (\Phi_1, \Phi_2)$ are being used to segment the given image. Then the fitting energy functional [68] can be described as in Eq. (4.18):

$$F(c,\varphi) = v.\int_\Omega |H(\Phi_1)|\mathrm{d}x\mathrm{d}y + v.\int_\Omega |H(\Phi_2)|\mathrm{d}x\mathrm{d}y + \int \frac{1}{2}(|\nabla\Phi_1 - 1)^2\mathrm{d}x\mathrm{d}y + \int \frac{1}{2}(|\nabla\Phi_2 - 1)^2\mathrm{d}x\mathrm{d}y \\ +\lambda_1 M_1(\varphi) + \lambda_2 M_2(\varphi) + \lambda_3 M_3(\varphi) + \lambda_4 M_4(\varphi) \tag{4.18}$$

where,
$M_1(\varphi) = \int |u_0 - c_{11}|^2 H(\Phi_1)H(\Phi_2)\mathrm{d}x\mathrm{d}y$
$M_2(\varphi) = \int |u_0 - c_{10}|^2 H(\Phi_1)(1 - H(\Phi_2))\mathrm{d}x\mathrm{d}y$
$M_3(\varphi) = \int |u_0 - c_{01}|^2 (1 - H(\Phi_1))H(\Phi_2)\mathrm{d}x\mathrm{d}y$
$M_4(\varphi) = \int |u_0 - c_{11}|^2 (H(\Phi_1))(H(\Phi_2))\mathrm{d}x\mathrm{d}y$
and $\mathbf{c} = (c_{11}, c_{10}, c_{01}, c_{00})$ is a constant vector. The first two penalizing terms force the evolving contour to be as smooth as possible. The next two regularizing terms are added so as to force the evolving curve to stay as close to a signed distance function as possible. The constant vector **c** comprises the average intensities within each particular region of the level sets.

Similar to the two-phase level set implementation, the associated Euler–Lagrange equations as in Eqs. (4.19), (4.20) for φ can be deduced which are needed to be solved iteratively for evolution of the curve [68].

$$\begin{aligned} &\frac{\partial \Phi_1}{\partial t} = \left(\nabla^2 \Phi_1 - div\left(\frac{\nabla \Phi_1}{|\nabla \Phi_1|}\right)\right) \\ &+\delta_\epsilon(\Phi_1)\left[v div\left(\frac{\nabla \Phi 1}{\nabla \Phi_1}\right) - \mu\right] - \delta_\epsilon(\Phi_1)[(\lambda_1(u_0 - c_{11})^2 - \\ &\lambda_3(u_0 - c_{01})^2)H(\Phi_2)] \\ &-\delta(\Phi_1)[(\lambda_2(u_0 - c_{10})^2 - \lambda_4(u_0 - c_{00})^2(1 - H(\Phi_2))] \end{aligned} \tag{4.19}$$

$$\begin{aligned}
&\frac{\partial\Phi_2}{\partial t}=\Big(\nabla^2\Phi_2-div\Big(\frac{\nabla\Phi_2}{|\nabla\Phi_2|}\Big)\Big)+\\
&\delta_\varepsilon(\Phi_2)\Big[vdiv\Big(\frac{\nabla\Phi 2}{\nabla\Phi_2}\Big)-\mu\Big]-\\
&\delta_\varepsilon(\Phi_2)[(\lambda_1(u_0-c_{11})^2-\lambda_2(u_0-c_{10})^2)H(\Phi_1)]\\
&-\delta_\varepsilon(\Phi_2)[(\lambda_3(u_0-c_{01})^2-\lambda_4(u_0-c_{00})^2(1-H(\Phi_1))]
\end{aligned}\tag{4.20}$$

In a similar manner, the three-phase level set implementation [68] of the C-V algorithm also utilizes two level sets $\varphi=(\Phi_1,\Phi_2)$ to segment the image into three distinct regions. The energy functional associated with this formulation is given in Eq. (4.21) as:

$$\begin{aligned}
&F(c,\varphi)=v.\int_\Omega|H(\Phi_1)|\mathrm{d}x\mathrm{d}y+v.\int_\Omega|H(\Phi_2)|\mathrm{d}x\mathrm{d}y+\int\frac{1}{2}(|\nabla\Phi_1|-1)^2\mathrm{d}x\mathrm{d}y+\\
&\int\frac{1}{2}(|\nabla\Phi_2|-1)^2\mathrm{d}x\mathrm{d}y\\
&+\lambda_1M_1(\varphi)+\lambda_2M_2(\varphi)+\lambda_3M_3(\varphi)
\end{aligned}\tag{4.21}$$

where,
$M_1(\varphi)=\int|u_0-c_1|^2H(\Phi_1)H(\Phi_2)\mathrm{d}x\mathrm{d}y$
$M_2(\varphi)=\int|u_0-c_2|^2H(\Phi_2)(1-H(\Phi_2))\mathrm{d}x\mathrm{d}y$
$M_3(\varphi)=\int|u_0-c_3|^2(1-H(\Phi_1))\mathrm{d}x\mathrm{d}y$
and $c=(c_1,c_2,c_3)$ is a constant vector.

The PDEs that are needed to be solved to evolve the curve is given in Eqs. (4.22) and (4.23).

$$\begin{aligned}
&\frac{\partial\Phi_1}{\partial t}=\Big(\nabla^2\Phi_1-div\Big(\frac{\nabla\Phi_1}{|\nabla\Phi_1|}\Big)\Big)+\\
&\delta_\varepsilon(\Phi_1)\Big[v.div\Big(\frac{\nabla\Phi_1}{|\nabla\Phi_1}\Big)-\mu\Big]-\delta_\varepsilon(\Phi_1)[\lambda_1(u_0-c_1)^2H(\Phi_2)]\\
&-\delta_\varepsilon(\Phi_1)[(\lambda_2(u_0-c_2)^2(1-H(\Phi_2))-\lambda_3(u_0-c_3)^2)]
\end{aligned}\tag{4.22}$$

$$\begin{aligned}
&\frac{\partial\Phi_2}{\partial t}=\Big(\nabla^2\Phi_2-div\Big(\frac{\nabla\Phi_2}{|\nabla\Phi_2|}\Big)\Big)+\\
&\delta_\varepsilon(\Phi_2)\Big[v.div\Big(\frac{\nabla\Phi_2}{|\nabla\Phi_2}\Big)-\mu\Big]-\delta_\varepsilon(\Phi_2)[(\lambda_1(u_0-c_1)^2-\lambda_2(u_0-c_2)^2H(\Phi_1)]
\end{aligned}\tag{4.23}$$

However, it should be emphasized here that both these multi-phase level set formulations (and, for that matter, any other multi-phase level set formulation) suffers from the same problems as the two-phase ones as discussed above.

4.3 Evolutionary Formulation of Our Fast C-V Model

The C-V model essentially attempts to divide the image into different segments during the evolution of the contour. The evolving contour tries to minimize the fitting energy functional as defined in Eqs. (4.1), (4.15), (4.18) or (4.21) by dividing the image into the required number of class divisions, with each division having a distinct mean intensity value denoted by c_i. This division into different classes can be done by the use of discrete level sets as was shown in [40].

For any particular intensity value L, the image can be divided into two regions I_1 and I_2 where $I_1^L = \{u_0 < L\}$ and $I_2^L = \{u_0 \geq L\}$. The constants c_1 and c_2 are the mean intensity values in the regions I_1^L and I_2^L, respectively.

The fitting energy functional for L can then be calculated by Eq. (4.1). For all such values of L within the allowable intensity range $L_{(min,max)}$, the fitting energy can be calculated as $(Fit_{Lmin}, ..., Fit_{Li}, ..., Fit_{Lmax})$. Then the segmentation of the image by the C-V model is to simply establish a search procedure to obtain that value of for which the fitting energy functional in Eq. (4.1) becomes minimum. A similar approach can also be adopted for the vector-valued images, where the only difference arises in the calculation of the mean intensity values over all the channels of the given image, so as to minimize the fitness functional energy in Eq. (4.15).

For the extension of this algorithm towards multiphase level set implementation, the search procedure has to be simply extended to search for multiple levels or classes. To segment the given image into K number of divisions, the search procedure is executed to search for (K−1) number of levels. Using this technique, the fitting energy in Eq. (4.18) or (4.21) has to be minimized for image segmentation into four-class or three-class divisions.

The search procedure, in this case, can be implemented in various ways e.g. by using the basic linear search method [40]. However, it is well known that linear search is too slow to converge to its minimum value, if the search space is too large. This occurs primarily for the cases in which more than two classes are required to be detected. To overcome these limitations, in this work we propose a novel method using PSO based swarm intelligence algorithm [30, 49] and some of its variants [18, 34, 42, 57, 58] which can obtain the required solution quickly and with sufficient accuracy.

4.4 Particle Swarm Optimization (PSO) Based Fast C-V Model

PSO [36, 54] is a very popular swarm intelligence based algorithm developed by Kennedy and Eberhart. This model was inspired from the study of bird flocks or fish schools searching for food and their interactions within the group. PSO has been found to be efficient in finding optimal or near-optimal solutions for problems involving large search spaces. The algorithm has carved out a niche for itself in solving high-dimensional, nonlinear, non-convex, optimization problems and has been applied in a wide range of applications encompassing a variety of domains like image processing [7, 44], robot navigation and control [8, 9, 11, 12, 14, 19], fuzzy control [20, 53], instrumentation design problems [13] etc.

The simulation of the swarm movement is implemented by assigning randomized position and randomized velocity to each particle of the swarm and their respective position and velocities are updated in each iteration by following certain heuristic rules. The candidate solutions are allowed to fly through the search space, in search of better positions, with attraction to positions which yield better results. Let a candidate solutions position and velocity be denoted as $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots.., x_{iD})$ and $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots.., v_{iD})$, where i denotes the i^{th} particle in the D-dimensional search space. Each particle remembers its best performance i.e., the best position obtained by it as P_i, and also the global best position of the swarm as a whole, as P_b.

The present velocity v_i of the i^{th} particle at iteration (t+1) is affected by a combined effect of its old velocity, the particles distance from the global best position p_b and its distance from its own previous best position p_i, as given in Eq. (4.24). Using this new velocity, the position of the th particle can be updated using Eq. (4.25):

$$v_{id}(t+1) = wv_{id}(t) + m_1 r_1 (P_{id}(t) - x_{id}(t)) + m_2 r_2 (P_{bd}(t) - x_{id}(t)) \tag{4.24}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{4.25}$$

where, d is the d^{th} dimension in the dimensional search space, x_{id} is the position and v_{id} is the velocity of the i^{th} particle in the d^{th} dimension, m_1 and m_2 are the called the cognitive and social acceleration learning rates, r_1 and r_2 are two random numbers between 0 and 1 with uniform distribution and is the inertia weight, which is used to draw a balance between the local and the global searches [54].

The pseudo-code of the PSO algorithm is given in Fig. 4.2.

Till date, many different variants of the basic PSO algorithm have been proposed. To name a few variants, PSO by Clerc [18], Quantum behaved PSO (QPSO) [42, 57, 58] and QPSO with Differential Mutation operator (QPSO-DM) [34] have been devised and these variants can also be used in our proposed formulation of the C-V model.

Now, Fig. 4.3 describes our proposed implementation of PSO based Fast C-V algorithm.

Algorithm 5: PSO

```
Initialize the number of swarm particles $N$ and the dimension $D$ of the search space.;
Initialize $x$ and v in every D for all $N$.;
Initialize the lower and upper bounds on $x$ and $v$ as $(x_{min}, x_{max})$ and $(v_{min}, v_{max})$;
Initialize the constants $m_1$ , $m_2$ and $w$. ;
Initialize the total number of iterations T.;
Set itr=1.;
Set minimum fitness value of the entire swarm to a very high value, $f_b = 10^{200}$.;
while $itr < T$ do
    Evaluate the objective function $f$ for all $N$;
    for $i = 1$ to $N$ do
        Find minimum objective value $f_i^{min}$ attained by each $i^{th}$ particle till now.;
        Find $x_i^{min}$ at which $f_i^{min}$ occurs.;
        Store $x_i^{min}$ for every $i^{th}$ particle as $P_i$;
        if $f_i^{min} < f_b$ then
            Set $f_b = f_i^{min}$ ;
            Set $P_b = p_i$;
        end
    end
    for $i = 1$ to $N$ do
        for $j = 1$ to $D$ do
            Update $v_{ij}$ using eqn. (24).;
            Limit $v_{ij}$ within allowable limits as:;
            if $v_{ij} < v_{min}$ then
                set $v_{ij} = v_{min}$
            end
            else
                if $v_{ij} > v_{min}$ then
                    set $v_{ij} = v_{max}$
                end
            end
            Update $x_{ij}$ using eqn. (25). ;
            Limit $x_{ij}$ within allowable limits by ;
            if $x_{ij} < x_{min}$ then
                set $x_{ij} = x_{min}$;
            end
            else
                if $x_{ij} > x_{min}$ then
                    set $x_{ij} = x_{max}$
                end
            end
        end
    end
    Set itr=itr+1.;
end
Report $P_b$ as the best solution obtained by the swarm.;
```

Fig. 4.2 The classical PSO algorithm

```
Initialize the number of swarm particles $N$.;
Initialize the number of classes to be detected as $(K-1)$.;
Initialize x and v in every D for all $N$.;
Initialize the lower and upper bounds on v as $()v_{min}, v_{max})$.;
Initialize the allowable ranges of x for all N by setting $x_{min} = 1$ and $x_{max} = 2500$;
Initialize the constants $m_1$ ,$m_2$ and $w$.;
Initialize the total number of iterations $T$.;
Set itr =1.;
Set minimum fitness value of the entire swarm to a very high value, $f_b = 10^{200}$. ;
Evaluate $L_{min}$ and $L_{max}$ as ;
$L_{min}$ minimum intensity in image $I$;
$L_{max}$ maximum intensity in image $I$;
while $itr < T$ do
    Divide I into K segments for all N by the following approach:;
    Let the class values be denoted as $S_i^1, S_i^2, ....., S_i^{k-1}$ with the assumption
    $S_i^1 < S_i^2 < ..... < S_i^{k-1}$ and also following the relation $S_i^1 \geq L_{min}$ and $S_i^{k-1} \leq L_{max}$ ;
    Each segmented region is generated by utilizing the different class values as shown
    below $I_{i^1} = \{L_{min} \leq I < S_i^1\}$ $I_{i^k} = \{S_i^{k-1} \leq I < S_i^2\}$ and so on … till
    $I_{i^2} = \{S_i^1 \leq I < L_{max}\}$;
    The above steps 1 and 2 are repeated for all $N$. ;
    Evaluate the objective function f for all $N$ by using the required equation for different
    image segmentations;
    eqn. (1) for two-class gray-scale images;
    eqn. (15) for two-class vector-valued images;
    eqn. (18) for three-class gray-scale images;
    eqn. (21) for four-class gray-scale images;
    for $i = 1$ to $N$ do
        Find minimum objective value $f_i^{min}$ attained by each th particle till now.;
        Find $x_i^{min}$ at which $f_i^{min}$ occurs.;
        Store $x_i^{min}$ for every $i^{th}$ particle as $P_i$.;
        if $f_i^{min} < f_b$ then
            Set $f_b = f_i^{min}$;
            Set $P_b = p_i$;
        end
    end
    for $i = 1$ to $N$ do
        for $j = 1$ to $D$ do
            Update $v_{ij}$ using eqn. (24). Limit $v_{ij}$ within allowable limits by
            if $v_{ij} < v_{min}$ then
                set $v_{ij} = v_{min}$;
            end
            else if $v_{ij} > v_{max}$ then
                set $v_{ij} = v_{max}$ ;
            end
            Update $x_{ij}$ using eqn. (25).;
            Limit $x_{ij}$ within allowable limits by ;
            if $x_{ij} < x_{min}$ then
                set $x_{ij} = x_{min}$ ;
            end
            else if $x_{ij} > x_{max}$ then
                set $x_{ij} = x_{max}$ ;
            end
        end
    end
    set $itr = itr + 1$. ;
end
Segment the image using the classes in and report the segmentation result.;
```

Fig. 4.3 Implementation of PSO based Fast C-V algorithm

4.5 Implementation and Results

Our proposed model has been implemented for several medical images and extensive experimental investigations and analyses of the results obtained are carried out, as discussed next.

4.5.1 Two-Class Implementation for Scalar Images

Figures 4.5, 4.6, 4.7 and 4.8 show the implementation of our proposed model on some sample medical images and their segmented results. The search procedure has been implemented by using PSO and some of its variants.

The segmentation performance can be validated both by visual perception and quantitative means. Among the common quantitative indices employed, Dice Coefficient (DC) [21, 56] is a popular choice. To calculate the DC, the manually segmented image (MSI) and ground truth (ASI) are compared for similarity by using Eq. (4.26)

$$DC = [2 * (ASI \bigcap MSI)]/[|ASI| + |MSI|] \tag{4.26}$$

where,
$|ASI|$ = cardinality of automatically segmented image (segmentation output from our proposed fast implementation of C-V algorithm)
$|MSI|$ = cardinality of manually segmented image (can also be considered as the ground truth)
$(ASI \bigcap MSI)$ = degree of correlation between the two image matrices.

Table 4.1 shows the performance of our proposed model for image segmentation results of Figs. 4.5 and 4.7. The segmentation performance is highlighted in the DC value obtained with our proposed model. For an accurate segmentation, the maximum

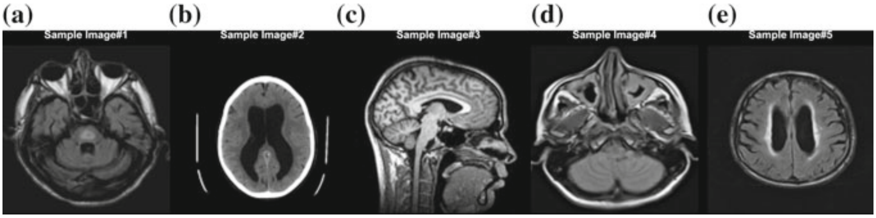


Fig. 4.4 Implementation of our proposed algorithm on some sample gray-scale images: **a** An image showing Central pontine myelinolysis [30], **b** A CTscan of the brain with hydrocephalus [48], **c** A brain MRI image [65], **d** MRI image showing sinusitis with edema and mucosal thickening appearing in both maxillary sinuses [28], **e** An image showing a patient with normal pressure hydrocephalus [22]

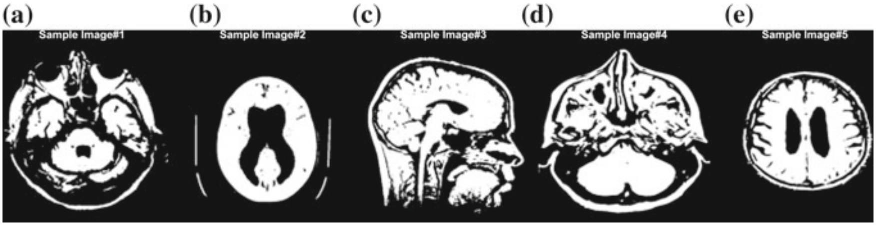


Fig. 4.5 Segmentation result obtained with our proposed model on images of Fig. 4.4. All the images are of 128 × 128 dimension

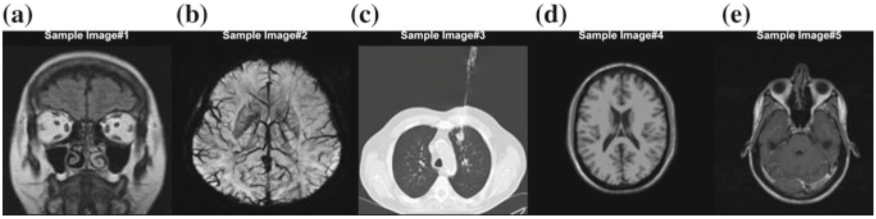


Fig. 4.6 Segmentation of some more sample medical images: **a** An image showing Coronal MRI Flair [27] **b** MRI image showing veins [1] **c** A sample CT scan image [31] **d** A sample brain MRI image **e** MRI image of brain [66]

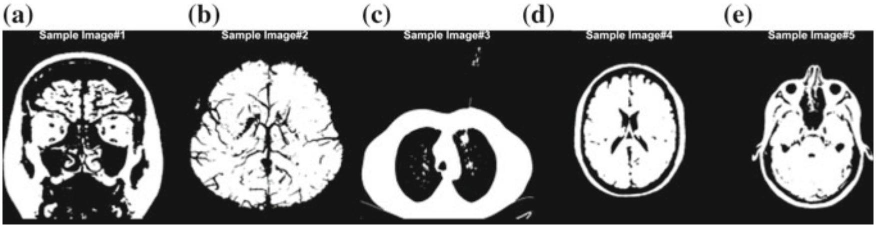


Fig. 4.7 Segmentation result of images of Fig. 4.6. All the images are of 128 × 128 dimension

DC value that can be obtained is 1. For our algorithm, the minimum DC value obtained is 0.91 which highlights the high performance achieved by our proposed model.

The segmentation of all the images in Figs. 4.5 and 4.7 has also been carried out using the basic C-V algorithm. The algorithm has been applied on each image for 2000 iterations. Table 4.2 highlights the fact that our PSO based C-V model is much faster than the basic C-V model while achieving a comparable segmentation performance.

Various variants of the PSO algorithm such as PSO-Clerc [18], QPSO [42, 57, 58] and QPSO-DM [34] have also been utilized to segment the images in Figs. 4.4 and 4.6. The swarm size for all the variants has been fixed at 10. To perform a suitable

Table 4.1 The segmentation performance of our proposed algorithm for the images in Figs. 4.4 and 4.6

Figure 4.2	DC	Figure 4.4	DC
(a)	0.9102	(a)	0.9577
(b)	0.9816	(b)	0.9845
(c)	0.9731	(c)	0.9935
(d)	0.9690	(d)	0.9905
(e)	0.9770	(e)	0.9833

Table 4.2 The computation time comparison between our proposed algorithm and the basic C-V model for the images of Figs. 4.4 and 4.6

Figure 4.2	C-V (s)	PSO based C-V (s)	Figure 4.4	C-V (s)	PSO based C-V (s)
(a)	5.658	0.070	(a)	4.152	0.062
(b)	3.260	0.062	(b)	4.777	0.054
(c)	4.599	0.056	(c)	0.802	0.054
(d)	5.206	0.060	(d)	4.510	0.051
(e)	4.229	0.051	(e)	3.793	0.053

comparison, the search procedure has also been carried out using the linear search method. Table 4.3 gives a computation time-complexity comparison between these competing methods as applied on the C-V algorithm. As it is evident from Table 4.3, it can be easily observed that our PSO based method and its variants are much faster than the linear search based one.

The computation time-complexity comparison in Table 4.3 shows that the PSO based method and its different variants are successful in segmenting the given images within a very short time and are comparatively much faster than both the linear search method and the basic C-V model implemented with level sets. For proper assessment of the different algorithms, accuracy is also a very important criterion which is highlighted in Table 4.4. Table 4.4 compares the accuracy of the result obtained between PSO and its different variants. The error is calculated by computing the difference of the value of "class" obtained by the linear method and those obtained by the PSO based method and its different variants, where the result obtained by the linear method is considered as the reference. It can be seen clearly that the error of the QPSO-DM based method is negligible for almost all the cases and thus, QPSO-DM emerges as the superior algorithm of choice, among different competing methods considered.

Table 4.3 The computation time comparison between the linear search method and the PSO based methods (and its different variants) for our proposed algorithm for the images of Figs. 4.4 and 4.6

Image	Linear search (s)	PSO (s)	PSO-Clerc (s)	QPSO (s)	QPSO-DM (s)
2(a)	0.128	0.070	0.074	0.073	0.076
2(b)	0.129	0.062	0.061	0.062	0.065
2(c)	0.134	0.056	0.055	0.056	0.057
2(d)	0.128	0.06	0.059	0.064	0.063
2(e)	0.102	0.051	0.053	0.053	0.052
4(a)	0.128	0.062	0.061	0.061	0.064
4(b)	0.118	0.054	0.053	0.053	0.053
4(c)	0.0987	0.054	0.056	0.056	0.056
4(d)	0.086	0.051	0.050	0.051	0.051
4(e)	0.098	0.053	0.054	0.053	0.058

Table 4.4 The accuracy of segmentation result obtained between the linear search method and the PSO based methods for the images of Figs. 4.4 and 4.6

Image	Class by linear search	Class by PSO	Error by PSO	Class by PSO-Clerc	Error by PSO-Clerc	Classes by QPSO	Error by QPSO	Class by QPSO-DM	Error by QPSO-DM
2(a)	64	61	3	64	0	64	0	64	0
2(b)	67	71	4	69	2	67	0	67	0
2(c)	79	83	4	83	4	79	0	79	0
2(d)	71	65	6	70	1	71	0	71	0
2(e)	61	58	3	62	1	61	0	61	0
4(a)	83	83	0	83	0	83	0	83	0
4(b)	72	76	4	74	2	72	0	72	0
4(c)	144	146	2	150	6	145	1	144	0
4(d)	62	61	1	62	0	62	0	62	0
4(e)	44	43	1	44	0	44	0	44	0

4.5.2 Two-Class Implementation for Vector Valued Images

Figures 4.8, 4.9, 4.10, 4.11 and 4.12 (Images courtesy [29], and Kieran Maher [43]) show the performances of our proposed C-V algorithm, when applied for segmentation of some vector-valued images. The results show that the segmentation achieved is quite successful in each case.

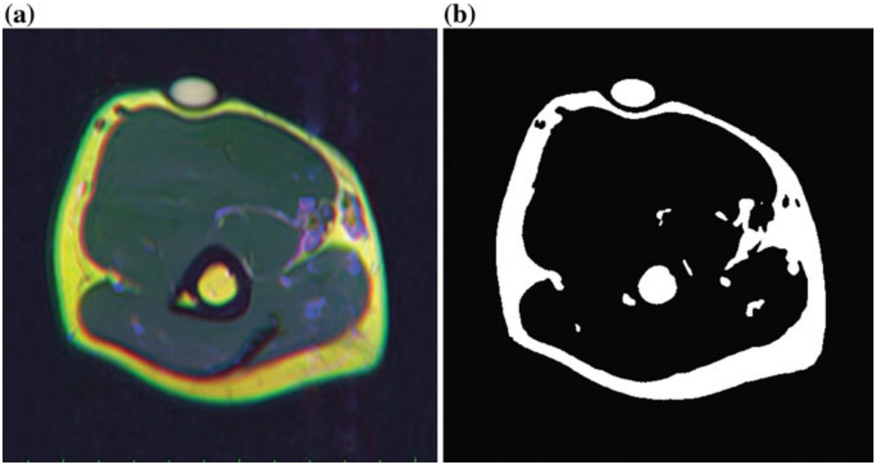


Fig. 4.8 An image showing subcutaneous lipoma in an arm MR image [23] (**a**), and its segmented result (**b**). Class value detected is 93. Time taken for segmentation = 0.203 s

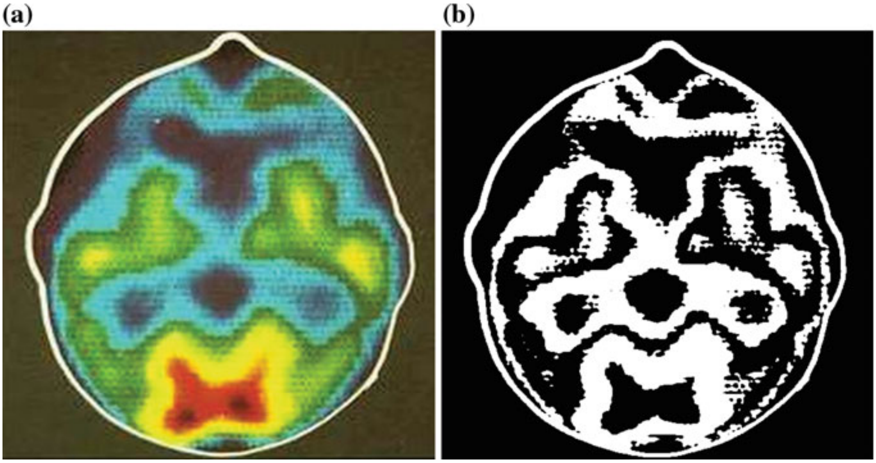


Fig. 4.9 A PET slice of a patient's brain, with a region of interest drawn to indicate skin surface [43] (**a**) and its segmented result (**b**). Class value detected is 114. Time taken for segmentation = 0.179 s

4.5.3 Multi-class Implementation of Our Proposed Model

Our proposed model can be very easily extended to the multi-class implementation for segmentation of images. For the three-phase level set implementation, the required number of classes i.e., $K = 3$ whereas for the four-phase level set implementation, K = 4. Figures 4.13, 4.14, 4.15, 4.16 and 4.17 show the segmentation result obtained on some sample gray-scale images (each having dimension 512 × 512) for the

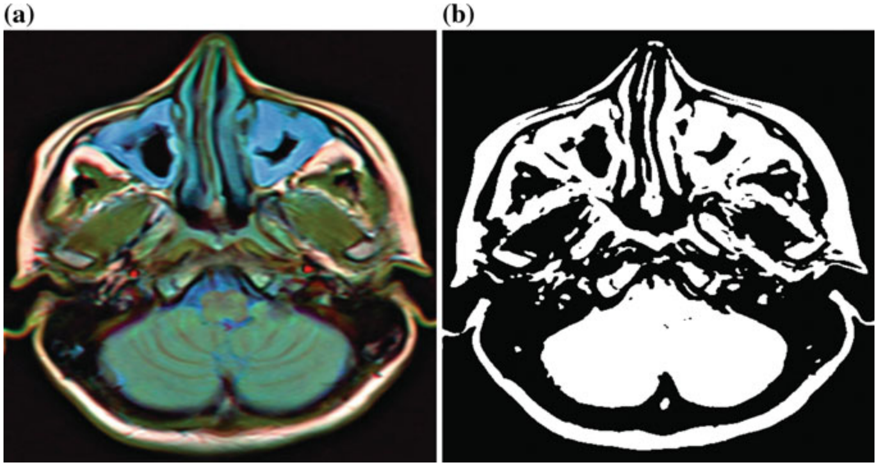


Fig. 4.10 A color brain MRI showing mucosal thickening in both maxillary sinuses [24] (**a**) and its segmentation (**b**). Class value detected is 77. Time taken for segmentation = 0.21 s

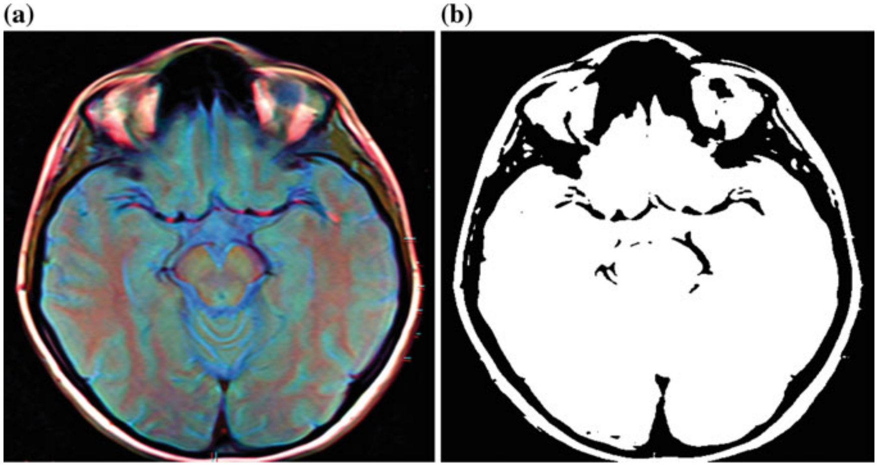


Fig. 4.11 A brain MR image [25] (**a**) and its segmentation result (**b**) by our proposed algorithm. Class value detected is 83. Time taken for segmentation = 0.194 s

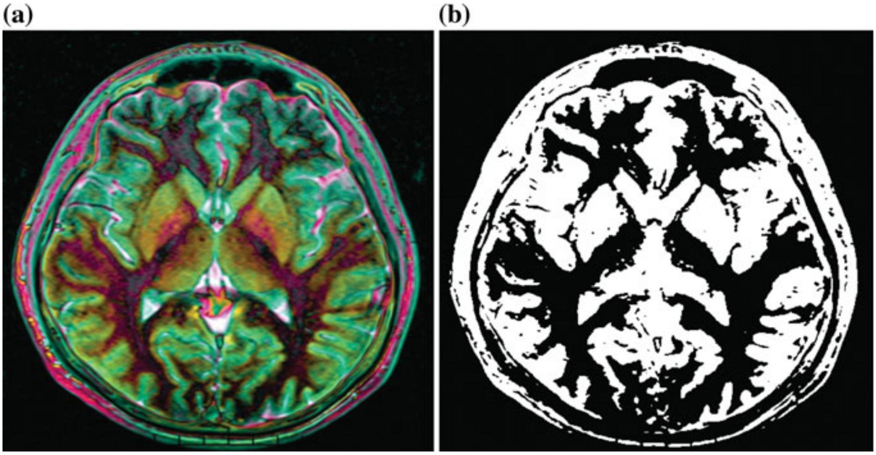


Fig. 4.12 A brain Magnetic Resonance image [26] (**a**) on which our algorithm has been applied (**b**). Class value detected is 74. Time taken for segmentation = 0.228 s

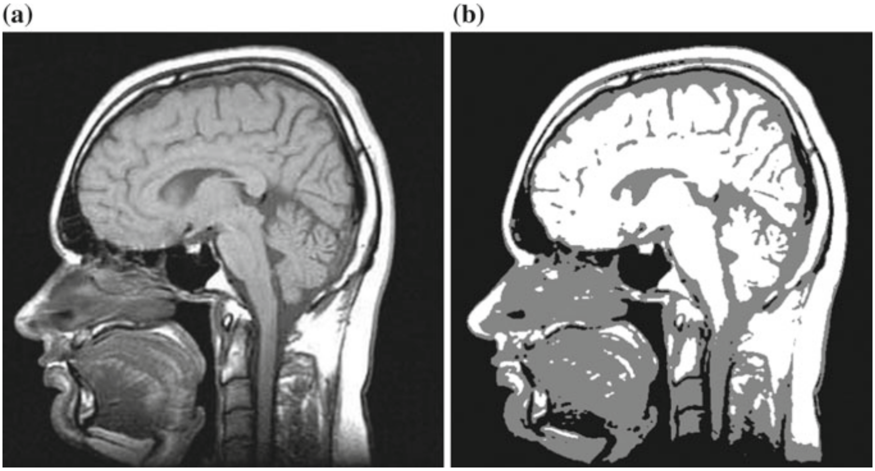


Fig. 4.13 A Magnetic Resonance Imaging scan of a head [68] (**a**), and its segmented result (**b**)

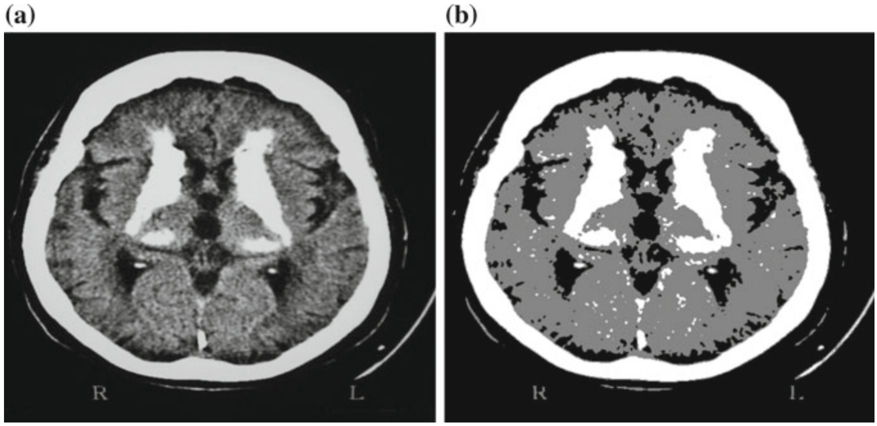


Fig. 4.14 A sample MRI image [67] (**a**) and the three-phase segmented result (**b**) with different colours showing the three different segmented regions

Table 4.5 The segmentation performance of our proposed algorithm for the images of Figs. 4.13, 4.14, 4.15, 4.16 and 4.17

Figure	Class values detected	
4.13	56	144
4.14	57	156
4.15	62	181
4.16	69	170
4.17	53	129

three-class implementation. Table 4.5 shows the detected class values for all the images of Figs. 4.13, 4.14, 4.15, 4.16 and 4.17. The performance enhancements achieved with our proposed algorithm for higher number of classes is similar to the case for the two-class implementation. It is quite evident from Table 4.6 that the speedup achieved by our PSO based algorithm is higher for larger number of classes, compared to the linear search based method or the basic C-V model, because of the larger search space, when more number of classes are required to be detected (Figs. 4.15, 4.16 and 4.17).

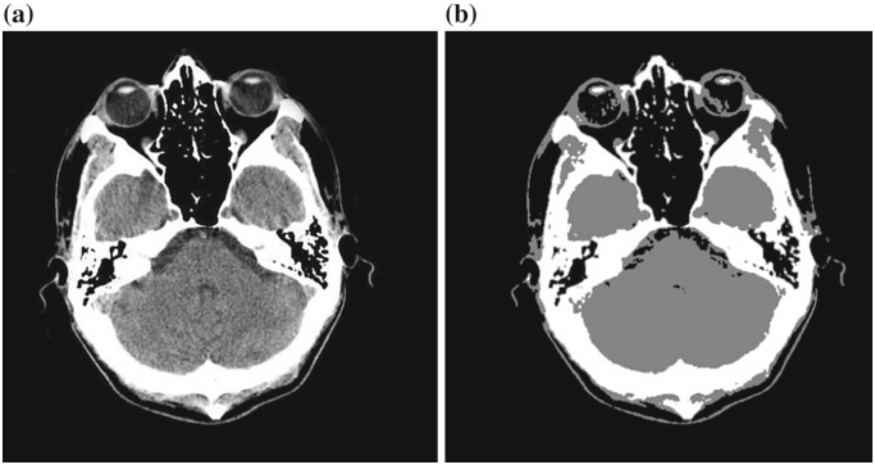


Fig. 4.15 A Normal CT scan of the head (**a**); this slice shows the cerebellum, a small portion of each temporal lobe, the orbits, and the sinuses [17]. The segmentation result is on (**b**)

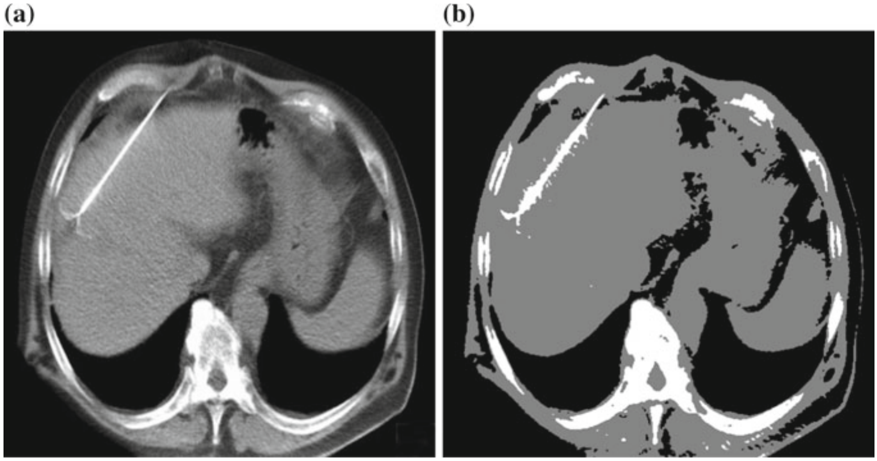


Fig. 4.16 A sample CT scan image [32] (**a**) and its segmentation result (**b**)

Table 4.6 The speedup achieved by our method compared to the linear search method for Figs. 4.13, 4.14, 4.15, 4.16 and 4.17

Figure	PSO based method (in seconds)	Linear search method (in seconds)	Speedup achieved
4.13	4.971	44.077	8.964
4.14	5.556	31.802	5.724
4.15	5.404	40.695	7.530
4.16	4.432	40.163	9.062
4.17	5.322	33.697	6.331

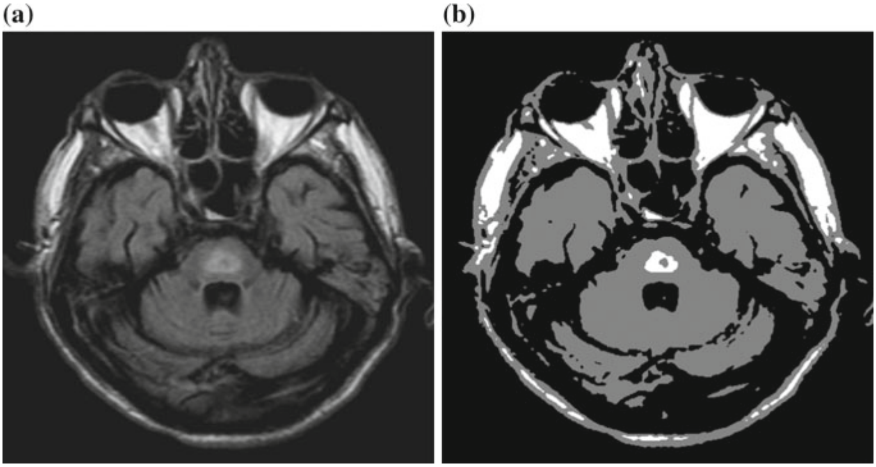


Fig. 4.17 An image showing Central pontine myelinolysis [30] (**a**) on which our algorithm has been applied (**b**)

4.6 Conclusions

In this work, a new, fast implementation of the C-V algorithm for image segmentation has been proposed. The problem has been reformulated into a new approach so that it can be solved by swarm intelligence based optimization algorithms such as PSO. The method has been employed for both scalar-valued and vector-valued images and has also been extended for multi-class level image segmentation. The proposed algorithm performs much faster than the original C-V model while also achieving accurate segmentation results. Future works will also attempt to use this proposed model to include local region information along-with the global information of the C-V model so that, for images suffering from intensity in-homogeneity too, better segmentation results can be achieved with reduced computation time.

References

1. S. Barnes, (2009), http://en.wikipedia.org/wiki/File:SWI_4Tesla.png
2. E.S. Brown, T. Chan, X. Bresson, Completely convex formulation of the Chan-Vese image segmentation model. Int. J. Comput. Vis. **98**, 103–121 (2012)
3. V. Caselles, F. Catt, T. Coll, F. Dibos, A geometric model for active contours in image processing. Numer. Math. **66**, 1–31 (1993)
4. V. Caselles, R. Kimmel, G. Sapiro, On geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
5. T. Chan, B.Y. Sandberg, L. Vese, Active contours without edges for Vector-Valued images. J. Vis. Commun. Image Represent. **11**, 130–141 (1999)
6. T. Chan, L. Vese, Active contours without edges. IEEE Trans. Image Process. **10**(2), 266–277 (2001)
7. A. Chander, A. Chatterjee, P. Siarry, A new social and momentum component adaptive PSO algorithm for image segmentation. Expert Syst. Appl. **38**(5), 4998–5004 (2011)
8. A. Chatterjee, F. Matsuno, A neuro-fuzzy assisted extended Kalman filter-based approach for Simultaneous Localization and Mapping (SLAM) problems. IEEE Trans. Fuzzy Syst. **15**(5), 984–997 (2007)
9. A. Chatterjee, F. Matsuno, A geese PSO tuned fuzzy supervisor for EKF based solutions of simultaneous localization and mapping (SLAM) problems in mobile robots. Expert Syst. Appl. **37**(8), 5542–5548 (2010)
10. A. Chatterjee, P. Siarry (eds.), *Computational Intelligence in Image Processing* (Springer, Heidelberg, 2013)
11. A. Chatterjee, K. Pulasinghe, K. Watanabe, K. Izumi, A particle swarm optimized fuzzy-neural network for voice-controlled robot systems. IEEE Trans. Ind. Electron. **52**(6), 1478–1489 (2005)
12. A. Chatterjee, R. Chatterjee, F. Matsuno, T. Endo, Neuro-fuzzy state modeling of flexible robotic arm employing dynamically varying cognitive and social component based PSO . Measurement **40**(6), 628–643 (2007)
13. A. Chatterjee, M. Dutta, A. Rakshit, An intelligent method of impedance measurement employing PSO-aided neuro fuzzy system with LMS algorithm, in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007)*, London, UK, 23–26 July 2007
14. A. Chatterjee, R. Chatterjee, F. Matsuno, T. Endo, Augmented stable fuzzy control for flexible robotic arm using LMI approach and neuro-fuzzy state space modeling. IEEE Trans. Ind. Electron. **55**(3), 1256–1270 (2008)
15. A. Chatterjee, P. Siarry, A. Nakib, R. Blanc, An improved biogeography based optimization approach for segmentation of human head ct-scan images employing fuzzy entropy. Eng. Appl. Artif. Intell. **25**, 1698–1709 (2012)
16. A. Chatterjee, H. Nobahari, P. Siarry (eds.), *Advances in Heuristic Signal Processing and Applications* (Springer, Heidelberg, 2013)
17. A. Ciscel, (2005), http://commons.wikimedia.org/wiki/File:Head_CT_scan.jpg
18. M. Clerc, J. Kennedy, The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space, in *IEEE Transactions on Evolutionary Computation*, Piscataway, NJ, (2002), p. 5873
19. K. Das Sharma, A. Chatterjee, A. Rakshit, A PSO-Lyapunov hybrid stable adaptive fuzzy tracking control approach for vision based robot navigation. IEEE Trans. Instrum. Measurement **61**(7), 908–1914 (2012)
20. K. Das Sharma, A. Chatterjee, A. Rakshit, A random spatial lbest PSO-based hybrid strategy for designing adaptive fuzzy controllers for a class of nonlinear systems. IEEE Trans. Instrum. Meas. **61**(6), 1605–1612 (2012)
21. R.L. Dice, Measures of the amount of ecologic association between species. Ecology **26**, 297–302 (1945)
22. N. Dilmen, (2005), http://commons.wikimedia.org/wiki/File:NPH_MRI_106.png

23. N. Dilmen, (2012), http://commons.wikimedia.org/wiki/File:Arm_mri.jpg
24. N. Dilmen, (2012), http://commons.wikimedia.org/wiki/File:Brain_MRI_112010_rgbca.png
25. N. Dilmen, (2012), http://commons.wikimedia.org/wiki/File:Brain_MRI_112445_rgbca.png
26. N. Dilmen, (2012), http://commons.wikimedia.org/wiki/File:Max_contrast_Brain_MRI_131058_rgbcb.png
27. N. Dilmen, (2012), http://commons.wikimedia.org/wiki/File:RetroOrbita_124440_MRI_FLAIR.png
28. N. Dilmen, (2012), http://en.wikipedia.org/wiki/File:Brain_MRI_112010_rgbca.png
29. N. Dilmen, http://commons.wikimedia.org/wiki/User:Nevit
30. Dr Frank Gaillard, (2008), http://en.wikipedia.org/wiki/File:CPM3.jpg
31. Hellerhoff, (2011), http://commons.wikimedia.org/wiki/File:CT-Biopsie-Lunge-BC.jpg
32. Hellerhoff, (2012), http://commons.wikimedia.org/wiki/File:RFA_CT_Leber_001.jpg
33. J.H. Holland, *Adaptation in Natural and Artificial Systems* (MIT Press, Cambridge, 1992)
34. M. Jamalipour et al., Quantum behaved particle swarm optimization with differential mutation operator applied to WWER-1000 in-core fuel management optimization. Ann. Nucl. Energy **54**, 134–140 (2013)
35. M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models. Int. J. Comput. Vis. **1**, 321–331 (1988)
36. J. Kennedy, R. Eberhart, Particle swarm optimization. Proc. IEEE Int. Jt. Conf. Neural Netw. Perth Aust. **4**, 1942–1948 (1995)
37. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A. Yezzy, Gradient flows and geometric active contour models, in *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, (1995), p. 810815
38. M.-S. Lee, G. Medioni, Inferred descriptions in terms of curves, regions and junctions from sparse, noisy binary data, in *Proceedings of the IEEE International Symposium on Computer Vision*, Coral Gables, FL, (1995), p. 7378
39. C. Li, C. Xu, C. Gui, M.D. Fox, Level set evolution without re-initialization: a new variational formulation, in *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, vol. 1, (2005), pp. 430–436
40. S. Li, Q. Zhang, Fast Image Segmentation Based on Efficient Implementation of the Chan-Vese Model with Discrete Gray Level Sets, on Mathematical Problems in Engineering (2013)
41. Y. Liu, K.M. Passino, Biomimicry of social foraging Bacteria for distributed optimization: models, principles, and emergent behaviors. J. Optim. Theory Appl. **115**, 603–628 (2002)
42. F. Liu, H. Duan, Y. Deng, A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching. Optik - Int. J. Light Electron Optics **123**, 1955–1960 (2012)
43. K. Maher, http://commons.wikimedia.org/wiki/File:PET_Slice_of_Brain.jpg
44. M. Maitra, A. Chatterjee, A hybrid cooperative comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. Expert Syst. Appl. **34**(2), 1341–1350 (2008)
45. M. Maitra, A. Chatterjee, A novel technique for multilevel optimal magnetic resonance brain image thresholding using bacterial foraging. Measurement **41**(10), 1124–1134 (2008)
46. R. Malladi, J.A. Sethian, B.C. Vemuri, A topology independent shape modelling scheme, in *Proceedings of the Trento Conference Geometric Methods Computer Vision II*, San Diego, CA, vol. 2031, (1993), pp. 246–258
47. R. Malladi, J.A. Sethian, B.C. Vemuri, Shape modeling with front propagation: a level set approach. IEEE Trans. Pattern Anal. Mach. Intell. **17**, 158–175 (1995)
48. L. Monfils, (2008), http://en.wikipedia.org/wiki/File:Hydrocephalus.jpg
49. D. Mumford, J. Shah, Optimal approximation by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42**, 577–685 (1989)
50. S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulation. J. Comput. Phys. **79**, 12–49 (1988)
51. L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms. Phys. D **60**, 259268 (1992)

52. N. Sanyal, A. Chatterjee, S. Munshi, An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation. Expert Syst. Appl. **38**(12), 15489–15498 (2011)
53. K. Sharma, A. Chatterjee, A. Rakshit, A hybrid approach for design of stable adaptive fuzzy controllers employing Lyapunov theory and particle swarm optimization. IEEE Trans. Fuzzy Syst. **17**(2), 329–342 (2009)
54. Y. Shi, R. Eberhart, A modified particle swarm optimizer, in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73
55. J.E. Solem, N.C. Overgaard, A. Heyden, Initialization techniques for segmentation with the Chan Vese model. Proc. Int. Conf. Pattern Recognit. **2**, 171–174 (2006)
56. T. Sorensen, A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons, Kongelige Danske Videnskabernes Selskab, vol. 5, (1957), pp. 1–34
57. J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior. IEEE Proc. Congr. Evol. Comput. **1**, 325331 (2004)
58. J. Sun, B. Feng, W. Xu, Adaptive parameter control for quantum behaved particle swarm optimization on individual level, in Proceedings of the 2005 IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, (2005), pp. 30493054
59. M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow. J. Comput. Phys. **119**, 146159 (1994)
60. L. Vese, T. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model. Int. J. Comput. Vis. **50**, 271–293 (2002)
61. C. Xu, J.L. Prince, Snakes, shapes and gradient vector flow. IEEE Trans. Image Proces. **7**, 359369 (1998)
62. T.A. Yezzi, A.S. Willsky, Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. IEEE Trans. Image Process. **10**(8), 1169–1186 (2001)
63. H.K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion. J. Comput. Phys. **127**, 179195 (1996)
64. S.C. Zhu, T.S. Lee, A.L. Yuille, Region competition: unifying snakes, region growing, energy/bayes/MDL for multi-band image segmentation, in *Proceedings of the IEEE 5th International Conference on Computer Vision*, Cambridge, MA, (1995), p. 416423
65. (2006), http://en.wikipedia.org/wiki/File:Sagittal_brain_MRI.JPG
66. (2007), http://en.wikipedia.org/wiki/File:Deviated_septum_MRI.jpg
67. http://commons.wikimedia.org/wiki/File:MD_de_Fahr1.jpg
68. http://commons.wikimedia.org/wiki/File:MRI_head_side.jpg

Chapter 5
Evidential Deformable Model for Contour Tracking. Application on Brain Cine MR Sequences

Sarra Naffakhi, Amir Nakib and Atef Hamouda

Abstract The goal of this paper is to introduce an efficient evidential particle filter for complex shapes tracking. The particularity of that particle filter is not only the fair use of the observation at the current time in the update step of it by performing a curve evolution but also it represents a bridge between Probability function and Evidence theory. This bridge can be illustrated by incorporating a data fusion step in the update phase. This method builds a track by selecting the best particles between the particle candidates. This re-sampling phase is based on choosing the particles possessing the higher value of the basic belief assignment function. The values of these basic belief assignment functions are resulting from the fusion process of two distinctive sources of information. The first source is the energy functional and the second one is the local sensitive histogram. The evaluation of our approach, which is made on a realistic Brain cine RM sequences, aims at tracking the motion of the walls of the third ventricle. Therefore, the latter shows its obvious and clear efficiency. In order to validate our proposal, we present a comparative study between our proposal and the state of the art methods. The obtained results are encouraging.

Keywords Complex shape tracking · Bayesian filtering · Particle filter · Evidence theory · Transferable belief model

5.1 Introduction

Researchers in distinct fields such as video compression, visual servoing, human/computer interaction and robotics have been interested in the topic of tracking object. The purpose of tracking, moving and deforming objects is to generate their state and positions at each frame of the video. For the non-rigid objects (or objects with complex shapes), one of the most descriptive representations is the contour representation

S. Naffakhi · A. Nakib (✉)
Laboratoire LISSI, Université Paris Est, 122 Rue P. Armangot, 94400 Vitry-sur-Seine, France
e-mail: nakib@u-pec.fr

S. Naffakhi · A. Hamouda
LIPAH, FST, University Tunis El Manar, Tunis, Tunisia

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_5

with either parametric or non-parametric models. Firstly, it is necessary to mention that one of the most frequent approaches (used among the diverse approaches suggested) for the state estimation is the Bayesian filtering. The oldest filter used is the Kalman Filter. It was used because of its optimal estimation for normally distributed state. Particularly, it requires a linear system and measurement models with Gaussian noises. Although this method allows obtaining good results in the case of linear or Gaussian context. This limitation was addressed by the use of the Extended Kalman Filter (EKF). In the first step the EKF linearizes the state and measurements equations, then, it applies the Kalman Filter to obtain the estimation of the state. However, the local linearization produced by the EKF does not fit estimations, particularly, when the state model is extremely non-linear. Similarly to the previous filters, the particle filter (PF) is widely used and it has been developed in order to represent the posterior density in terms of random samples and associated weights. The PF was proposed for solving both linear and non-linear system without any assumptions on the probability distribution.

In this paper, we are interested in the tracking of the movements of the walls of the third ventricle during the cardiac cycle independently of the cerebrospinal fluid (CSF). In the state of the art, there are other works related to the motion analysis using CSF [16, 17, 21, 22]. This tracking problem can be seen as a tool for motion estimation of the third ventricle and thus a tool for facilitating diagnostic and therapeutic decision. In this work, we propose an algorithm called Evidential Particle Filter For Complex Shapes Tracking (EPF-CSCT). This algorithm aims to exploit the Evidence Theory in facing up to the drawbacks of the probability theory. Moreover, the purpose of this paper is to compare the use of the enhanced Particle Filter, the Extended Kalman Filter and the Kalman Filter in a medical image context. Analysing accuracy and robustness against the noise presented in the cine RM images, we show that the EPF-CSCT performs better than the KF and the EKF.

The rest of the paper is organized as follows: in Sect. 5.2, we present the related work before introducing the Background. Then, in Sect. 5.4, the proposed approach is detailed. In Sect. 5.5, the validation of our approach on synthetic and real world data is presented. Finally, conclusion and perspectives are presented in the last section.

5.2 Related Works

Numerous works on objects tracking were proposed in the literature and many classifications also were proposed as that in [11, 41] according to the shape or appearance models of the tracked targets: (i) point tracking, (ii) kernel based tracking and (iii) silhouette based tracking.

(i) *The Point Tracking Methods*: this is the most used approach due to its simplicity and its easy manipulation with complex problems. The point tracking methods are mainly used for point tracking in radar imagery [26], distributed point tracking [13] or for Monte Carlo techniques where the number of samples prohibits heavy calculations [1, 6, 28]. The target is represented by set of pixels having some particular characteristics of interest in the object. It requires an efficient algorithm to detect objects in each consecutive framed.

(ii) *The Kernel Tracking*: [32], the kernel or object can be a rectangular template or an elliptical with a corresponding histogram. Then, the motion of the kernel in consecutive frames is computed to track the object. This motion is usually in the form of rigid transformation. The convergence of these algorithms depends on the representation, the number of object tracked and the method used for approximating the object motion. We can cite the most used techniques that use this approach Simple Template Matching [30], The Mean Shift Method [3], the Support Vector Machine [25], and the Layering based tracking [37].

(iii) *Silhouette Tracking Methods*: in the silhouette tracking methods, the tracking is performed by using the information encoded within the segmented object region. The encoded information can be: appearance density or shape models. Then, the models are tracked by the contour evolution or shape matching.

Contour tracking methods deform iteratively an initial contour at $t-1$ to its new position at t. This contour evolution requires that some parts of the object in the current frame overlap with the object region in the previous frame. In literature two classes of contour representation can be distinguished: the explicit and implicit representation. The explicit representation defines the boundary of the silhouette by a set of control points, while, the implicit one defines the silhouette by means of a function defined on a grid.

In [9] the authors use an explicit method where the B-spline representation for contours of objects is used. They defined a propagation rule of shapes for inferring a posterior state density from the data to provide a prior to learn. Then, the state is updated using a particle filter. The image edges' calculated in the normal direction at the control points on the contour are used to define the current state (the measurements) to update the state. The initialization of the particle filter algorithm is obtained by computing the state variables from the contours extracted in consecutive frames during a training phase. This work was extended in [19] to a particle filter-based multiple object tracker by including the exclusion principle for handling occlusion.

In [14, 39] a particle filter framework is proposed besides the use of the prior dynamical model on the deformation and on the similarity group parameters for tracking the deformation and the global motion overtime. In [38], the authors proposed an algorithm where the Kalman filter and active contours are combined to track non-rigid objects, where the prominent B-spline method is again used for representation.

In [24], the authors put dynamics into the geodesic active contour framework for contour. In [27], the authors describe a unified approach for the detection and tracking of moving objects by the propagation of curves. A scheme is proposed that can be viewed as a geodesic active motion detection and tracking model which basically attracts the given curves to the bottom of a potential well corresponding to the boundaries of the moving objects. Yilmaz et al. model the object shape and its changes by means of a level set based shape model in [40], where the grid points of the level set hold the means and the standard deviations of the distances of points from the object boundary and the object occlusions are resolved during the course of tracking. In [10], the authors propose a nonlinear model for tracking a slowly deforming and moving contour despite significant occlusions. In [29], the authors add Mumford-Shah model into the particle filter framework. Both of the contour tracking methods and shape matching have the advantage of dealing with variety of object shapes, handling occlusions and object split and merge.

5.3 The Evidence Theory

Evidence theory (ET) is known in the literature also under the terminology of Dempster–Shafer theory or theory of belief functions. The ET was introduced first by Shafer in the 1970s [31] as a reinterpretation of Dempster's statistical inference, which led to define a theory particularly suitable for handling imprecise and uncertain information. Since that different interpretations of its foundations have been suggested. We have adopted, here, the interpretation proposed by Smets: called the transferable belief Model (TBM) which is characterized by a non-probabilistic appearance [33, 35, 36]. It is based on two levels of information perception. The first consists in the credal level allows formatting and editing information. While the second, it is the pignistic level which is completely dedicated to decision making and it is clearly separated from the data modeling.

Let Ω be the frame of discernment which contains N hypothesis or solutions for the problem to be solved: $\Omega = \{H_1, .., H_N\}$. The set of propositions associated to Ω is defined as 2^{Ω}=$\{\emptyset$, $\{H_1\}$,..,$\{H_N\}$,$\{H_1 \bigcup H_2\}$,..,$\Omega\}$. Under the TBM, a statement of belief from a source S is characterised by a degree of belief, then, it is defined as a basic belief assignment (bba) function m: $2^{\Omega} \rightarrow [0, 1]$ that satisfies the following conditions: $\sum_{A \subseteq \Omega} m(A) = 1$. For each proposition $A \in 2^{\Omega}$, $m(A)$ represents the belief strictly placed in A at a given time t. Besides to the bba function m, there are other representations for the same information such as the credibility, the plausibility and it is possible to switch from one to the other by the Mobius transformations [15]. For the dynamic credal level and the pignistic level we will present only the items that are used in this work. They are defined in the context of the application, and thus are detailed in the next section.

5.4 Proposed Method

In this section, we present our approach and focus on the way of taking profit from the evidence theory to enhance the quality of the prediction of the particle filter. Especially, in the case of tracking an object with complex shape.

In order to enhance the performance, we formulated the problem of adding information as a fusion information problem. Then, the evidence theory is exploited for this procedure of fusion.

In the classical form of the particle filter, the step after the prediction phase consists in the update step. We propose to use not only one source of information but also combine two sources to assign weights for particles. This proposal is motivated by:

1. The nature of data to treat: the cine-MRI sequences are noised due to physical phenomena and to the acquisition system. Consequently, imprecision and uncertainty are present, especially, when the resolution is poor (lack of information).
2. Regarding the particle filter: if a particle cannot be observed what is its likelihood?

To address these different problems, a method that is based on information fusion using the evidence theory is proposed.

5.4.1 Description

In our context of contour tracking, the belief that the particle belongs to the contour of the object of interest must be evaluated.

- Target and its state: the target of interest is the contour of the region of interest ROI (such as walls of the third ventricle: ROI1, ROI2 see Fig. 5.1), so the state is defined at each frame t as $X_t = (x, y)$ where x, y are the coordinates in the frame t.
- The observations Y_t: the frame at t will be used to extract the set of features based on the sources of information. Through the use of the final information drawn from the combination of two sources, the posterior probability distribution $p(X_t|Y_{1..t})$ can be estimated.

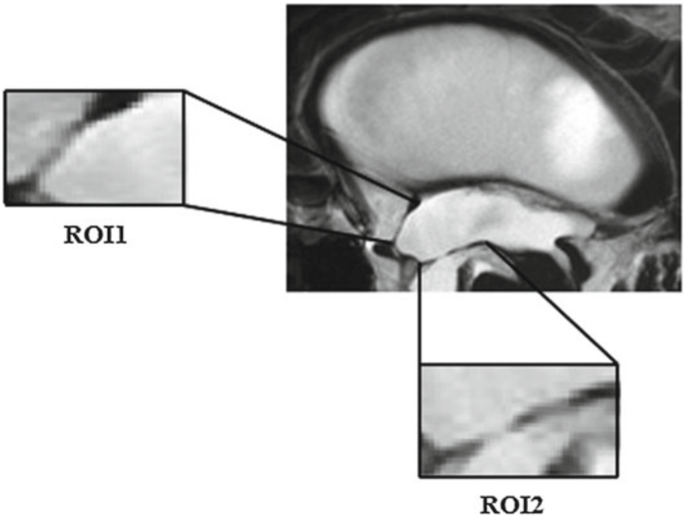


Fig. 5.1 Example of the two ROI

- The state transition density $p(X_t|X_{t-1})$: In our case, this density is chosen to be a linear motion.
- Particles: In our case, we have chosen the particles as the scatter plot around the contour detected at $t = 0$ with the model proposed by [18].
- The frame of discernment will be composed of two exclusive hypotheses $\Omega = \{H1, H2\}$ such as:
  H1: the particle is a pixel of the contour.
  H2: the particle is a pixel of the background.
- Two sources of information are used $\{S_d\}_{1 \leq d \leq 2}$:
  - The energy functional: we considered the contour as a snake and used the energy expression according to Kass et al. [14] model. We recall that a snake is an elastic curve, that from an initial state tries to adjust to the most significant features of the scene. It is deformed due to external forces that attract it to suitable feature of the image, and internal forces that try to preserve the condition of smoothness in the shape of the curve. A final solution (the contour) is given by the minimum total energy of the snake.
  - The local sensitive histogram: the local sensitive histogram for an image I at a pixel p is defined by:

$$H_p^E(b) = \sum_{q=1}^{W} \alpha^{|p-q|} Q(I_q, b), b = 1, .., B \tag{5.1}$$

    where $\alpha \in [0, 1]$ is a parameter controlling the decreasing weight as a pixel moves away from the target center. B is the total number of gray levels, W is the number of pixels and $Q(I_q, b)$ is zero except when intensity value I_q (at pixel location q) belongs to a gray level b[7].

- Initialization: the initialization step of the algorithm differs from $t = 1$ to $t > 1$. If $t = 1$, the position of the scatter point will be generated from the contour point detected at $t = 1$. If $t > 1$, the position of the scatter point will be generated from the estimated contour at $t - 1$.

5.4.2 Evidential Modeling

We propose the use of the distance-based model introduced by Denoeux [4, 42]. This model requires the knowledge of a training set. Let L be a training set, then:

$$L = \{(x^i, H_q), x^i \in \phi_k^s\} \tag{5.2}$$

with ϕ_k^s is the set of the K nearest neighbors to x^s in the training set X, and $X = \{x^s\}$ where x^s is the pattern to be classified.

As hypothesis $H_q \in \Omega$ is associated for each x^i, and using the training set, we can build a bba function for each pattern x^s to be classified via the knowledge of belonging of its k nearest neighbors.

The information provided by each neighbor is, then, modeled by a belief function m^i which is defined by:

$$\begin{cases} m^i(H_q) = & \alpha_q \exp\{-\gamma_q^2 d(x^s, x^i)^2\} \\ m^i(\Omega) = & 1 - m^i(H_q) \\ m^i(A) = 0 \; \forall A \in 2^{\Omega} \backslash \{\Omega, H_q\} \end{cases} \tag{5.3}$$

where $\alpha_q \in [0, 1]$ is a constant, γ_q is a parameter associated to the class H_q, and $d(x^s, x^i)$ is the distance between $x^s$and x^i.

The first source, $d(x^s, x^i)$ is defined as the simple distance between the two energy's values. While the second source, is the distance between two local sensitive histograms:

$$d(H_1, H_2) = \sum_{b=1..B} \mid H_1(b) - H_2(b) \mid \tag{5.4}$$

where H_1 and H_2 are the local sensitive histograms of the real particle and the candidate, in the current frame. B is the number of gray levels.

The Eq. (5.3) reflect that the couple (x^i, H_q) provides information about the class membership of x^s to the hypothesis H_q, because x^i has only the information that it belongs to the class H_q. So, the bba function calculated is only distributed on the hypothesis H_q or on the frame of discernment Ω. These belief functions are, then, combined to take a decision on the membership of the vector x^s.

5.4.3 *Combination Rule from Each Source*

Smets [35] defined the conjunctive rule of combination, noted $m_\cap = m_1 \cap m_2$, by:

$$\begin{cases} m_\cap : \quad 2^{\Omega} \times 2^{\Omega} \longrightarrow [0, 1] \\ m_\cap(A) = \sum_{B \cap C = A} m_1(B).m_2(C) \forall A \subseteq \Omega \end{cases} \tag{5.5}$$

However, two conditions must be satisfied to apply this combination rule:

- The combined bba should be independent.
- The bba function should be in $[0, 1]$ and $\sum m(A) = 1$.

5.4.4 Decision Strategy

To take a decision, Smets and Kennes [34, 35] propose to use the pignistic transformation that transforms the beliefs into optimum decisions. The Smets pignistic probability transformation ($BetP$) uses the bbs to assign a pignistic probability to subsets. This transformation allows to build a uniform probability distribution for each subset. It is defined by:

$$BetP(H_k) = \frac{1}{1 - m(\emptyset)} \sum_{A \in H_k} \frac{m(A)}{|A|} \tag{5.6}$$

where $|A|$ cardinality of $A \subseteq \Omega$.

Moreover, the pignistic probability can be obtained from an infinite number of bba functions. Indeed according to a bba function we can associate only one pignistic probability. In our case, the important particles will be those with highest values of the pignistic probability.

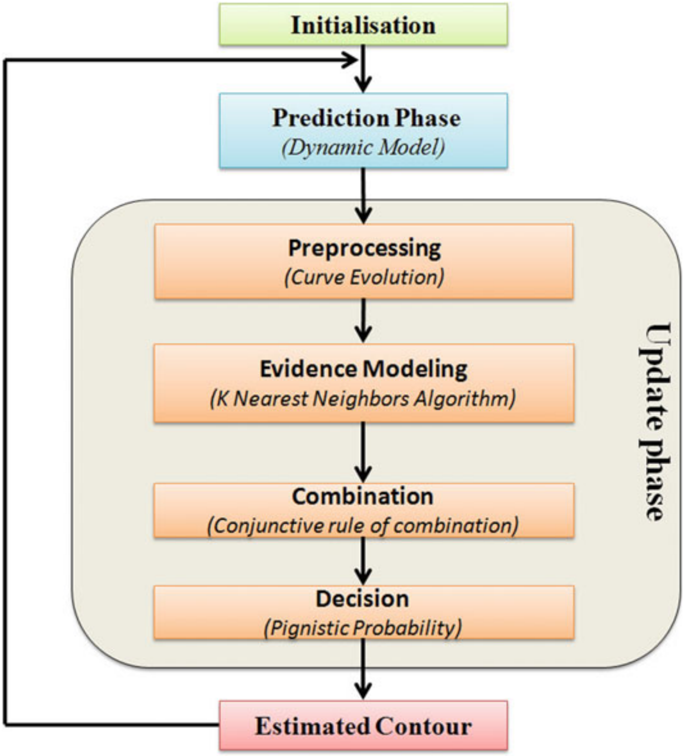


Fig. 5.2 The Evidential Particle Filter for Complex Shape's Contour Tracking

5.4.5 Proposed Algorithm

Our proposed method is explained in the next flowchart shown in Fig. 5.2.

Algorithm 6: The Evidentiel Particle Filter for Complex Shape Tracking

```
0. Initialisation:
if $t = 1$ then
  Generate the position of the scatter point from the contour points detected at t=1.
else
  Generate the position of the scatter point from the contour points estimated at t-1.
end if
for each frame do
  1.Prediction Step:
  Generate the set of particles $\{x_t^i\}_{i=1}^N$ in the current frame, at t, following the dynamic
  model $p(x_t|x_{t-1})$.
  2.Update Step:
  for each particle $x_t^i$ do
    2.1.$Preprocessing$ : Perform M iterations of curve evolution depend on the
    observation $y_t$
    2.2.$EvidentialModeling$ :
    Calculate its value from each source of information
    Apply the K nearest Neighbor Algorithm
    Compute the bba function for each nearest neighbor $x_t^i$ $1 \le i \le K$
    Apply the conjunctive rule of combination to get the bba function for each particle $x_t^i$
    2.3.$Combination$ :
    Input:Bba Function derived from each source $\{x_t^i, S_1\}$ and $\{x_t^i, S_2\}$
    Apply the conjunctive rule of combination again.
    Output: Final bba function m($x_t^i$).
    2.4.$Decision$ :
    Input:m($x_t^i$)
    311267 Calculate the pignistic probability
    output: Pignistic probabilty of each $x_t^i$
    2.5.$Resampling$ :
    Resample from the above distribution to generate particles, then we have: the final
    contour points'position to display.
  end for
end for
```

5.5 Results

In this section, we are going now test our proposed approach on several synthetic and real videos with dynamic objects and demonstrate the tracking performance of our algorithm against the state-of-the-art. We are also going to compare our proposed algorithm: the Evidential Particle Filter for Objects'Contour Tracking with Complex Shape against traditional tracking methods such as the Kalman Filter (KF) and the

Extended Kalman Filter (EKF). The different tracked states result from the proposed Evidential Particle Filter, the Kalman Filter and the Extended Kalman Filter have been validated by comparing them with the state of each region detected manually in t. We would like to point out that despite taking a lot of care while testing active contour segmentation methods, and getting in touch with some authors, we have not been able to get accurate results. However, we are confident that our manual detection is correct and accurate.

5.5.1 Tracking Synthetic Ball

We have set up experiments first using synthetic images to qualitatively evaluate the proposed algorithm towards the Kalman Filter and the Extended Kalman Filter. For our purpose, we have created a sequence of synthetic images. It has one dynamic target object (white ball) surrounded by a black background. Because of most of the complex dynamic trajectories cannot be modeled by lineal systems, which results in that we have to use for the modeling nonlinear equations, we distinguish two cases: the first one is a linear moving ball which is shown in Fig. 5.3 and the second case is a non-linear motion. The latter is illustrated also in Fig. 5.4. To make it even more complicated we have added to all the images a speckle noise and gaussian noise so that all the images in general and the edges in particular become fuzzy. In order to validate and so evaluate our proposal, we describe first the experiments models, parameters for each filter, and second the preliminary results that we realized to this aim.

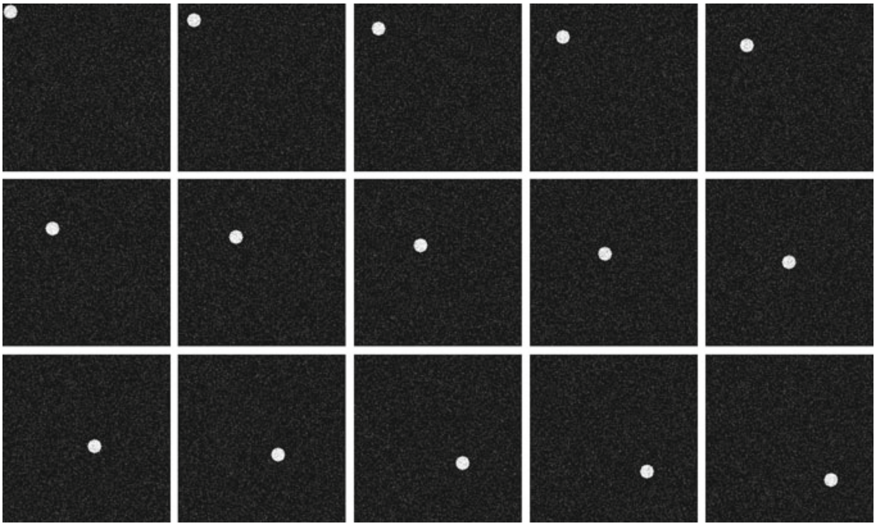


Fig. 5.3 Ball in a linear motion

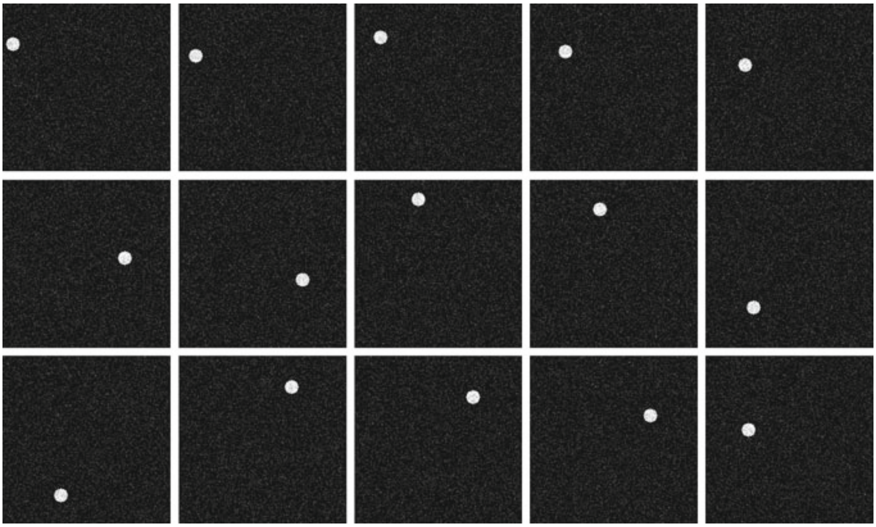


Fig. 5.4 Ball in a non-linear motion

1. Kalman Filter:
   In our case for the estimation of the contour motion tracking, we choose the dynamic model as follow:

$$\begin{cases} x(t) = x(t-1) + v * t + 1/2 * u *^2 t \\ v(t) = v(t-1) + u * t \end{cases} \tag{5.7}$$

   where the state includes position in x and y direction, velocity $v(t)$ and u is the acceleration. For the ball's contour tracking, u is fixed to the value 10^{-1}. The variance of the process noise is fixed to the value 10^{-2} and the measurement noise is fixed to the value 10^{-3}. The matrix F, G, H in equations defined in (5.15) and (5.19) are derived from this supposed model, dt is fixed to 1, as follow:

$$F = \begin{pmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} G = \begin{pmatrix} dt^2/2 \\ dt^2/2 \\ dt \\ dt \end{pmatrix} \text{ and } H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

2. Extended Kalman Filter: We have used the following modeling criterion to capture the non-linear dynamics motion:

$$\begin{cases} x(t) = x^2(t-1) + 1/2x(t-1) \\ y(t) = y^2(t-1) + 1/2y(t-1) \\ v_x(t) = v_x^2(t-1) \\ v_y(t) = v_y^2(t-1) \end{cases} \tag{5.8}$$

where the state include position in x and y direction, velocity $v(t)$. For the ball's contour tracking, the variance of the process noise is fixed to the value 10^{-2} and the measurement noise is fixed to the value 10^{-3}. The matrix F for state and the matrix H for measurement, in which we just measure the x and y position and not the velocity, which are defined in Eqs. 5.29 and 5.30 and are derived from this supposed model, as follow:

$$F = \begin{pmatrix} 2*x+1/2 & 0 & 0 & 0 \\ 0 & 2*x+1/2 & 0 & 0 \\ 0 & 0 & 2*x & 0 \\ 0 & 0 & 0 & 2*x \end{pmatrix} \text{ and } H = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix}$$

3. Evidential Particle Filter For Contour Tracking:

First, we note that the parameters used are defined in Table 5.1.

To carry out the comparative studies between the Kalman Filter and the Extended Kalman Filter for the ball tracking. Figure 5.5 represents the results obtaining with the Kalman Filter and in which we consider the linear uniform movement defined in Eq. (5.7). Next, Fig. 5.6 illustrates those results in which we consider the same linear

Table 5.1 Parameters of the Evidential particle filter for Complex Shape Tracking

	♯ Number of particles		♯ iterations	♯ bin
	Region 1	Region 2		
Dataset1	288	264	50	32
Dataset2	408	432	50	32
Dataset3	432	408	50	32
Synthetic images	352		40	32

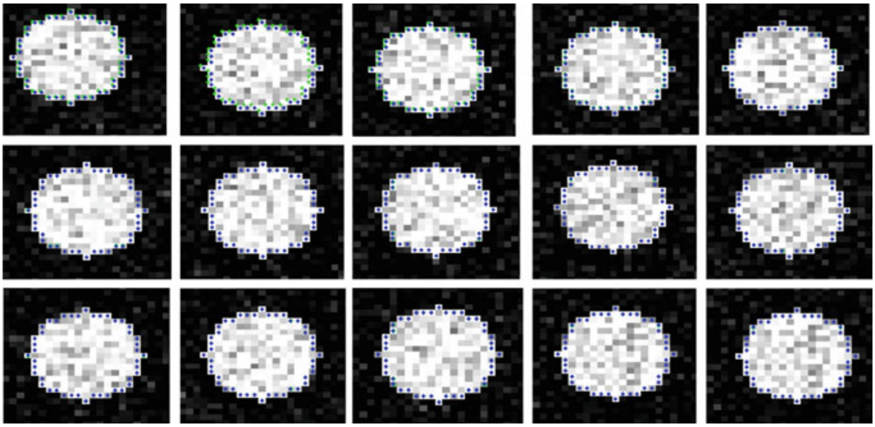


Fig. 5.5 Result of the Kalman Filter for the contour tracking of the ball in linear motion

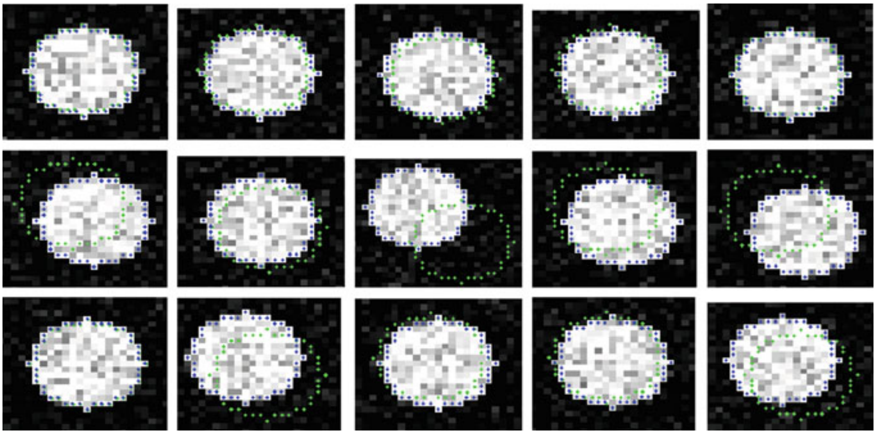


Fig. 5.6 Screenshots result of the Kalman Filter for the contour tracking of the ball in non-linear motion

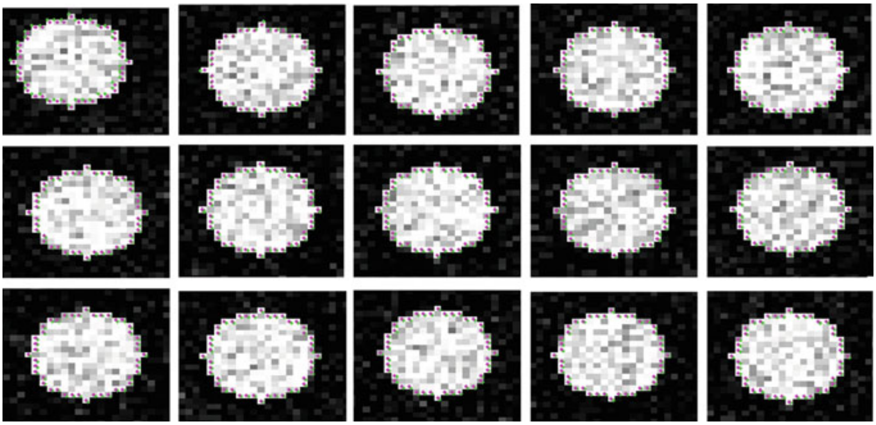


Fig. 5.7 Screenshots result of the Extended Kalman Filter for the contour tracking of the ball in linear motion

uniform movement for tracking the non-linear motion of the ball. The real positions are represented with blue point and the estimated one are represented with green color. Then, Fig. 5.7 shows the results obtained with the Extended Kalman Filter and in which we use the non-linear uniform movement described by the system equations in Eq. (5.8) for tracking the linear motion of the ball. After, Fig. 5.8 shows the results obtained with the Extended Kalman Filter using the same non-linear uniform movement described in Eq. (5.8) for tracking the non-linear motion of the ball. The estimated positions are represented with green color and the right ones with magenta color.

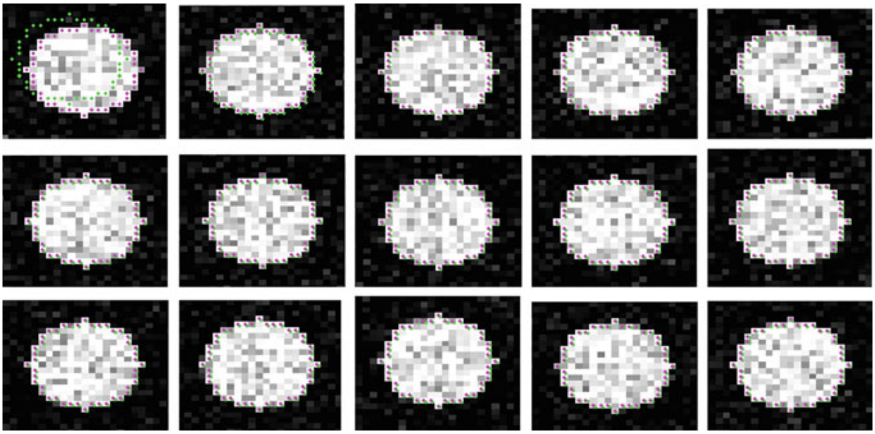


Fig. 5.8 Screenshots result of the Kalman Filter for the contour tracking of the ball in non-linear motion

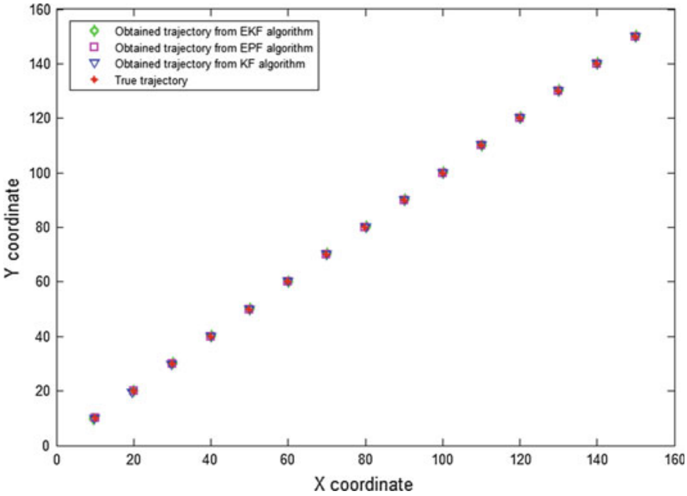


Fig. 5.9 Comparisons between the real state position and the obtained trajectories of the Center ball along x-y axis from the different algorithms (the linear motion)

Finally, Figs. 5.9 and 5.10 illustrate the center of the target position is shown for each instant as well as the real the real trajectory for the linear and non-linear motion respectively. Therefore, both figures show the acting of the Extended Kalman Filter and its ability to be adapted to any type of motion versus the poor performance of the normal Kalman filter in the non-linear motion. In deed, the later loses good state's estimation at each time the object changes its direction of motion. From these observations, we can also conclude that the comparison of estimated trajectories from the Extended Kalman Filter with the estimated trajectories from the Evidential Parti-

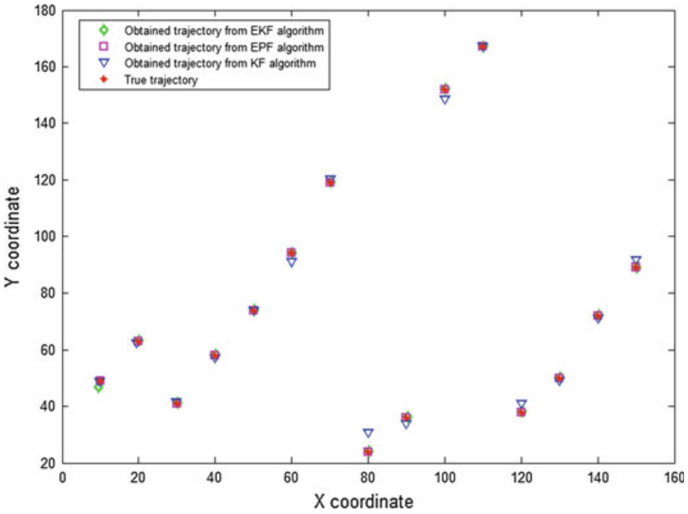


Fig. 5.10 Comparisons between the real state position and the obtained trajectories of the Center ball along x-y axis from the different algorithms (the non-linear motion)

cle Filter is reasonably good. However, in some cases, as illustrated, the Evidential Particle Filter exhibits better performance than the EKF and it is more accurate.

5.5.2 Datasets and Data Acquisition

Here, after we describe the data sets used:

1. We used three brains Cine-MRI for testing and validating the proposed algorithm and the other algorithms described earlier.
2. The image dataset is of type cine-MRI image. This technique of acquiring brain Cine-MRI images was carried out by Professor P. Decq (neurosurgeon) and Dr. J. Hodel (neuroradiologist) at Henri Mondor Hospital as part of the antenna Analysis and Restoration of Movement (CNRS-ENSAM Paristech 8005). The principle of this technique is to synchronize the MRI signal with the ECG signal (electrocardiogram). The MRI signal provides three-dimensional images and anatomically precise. The ECG signal is the result of cardiac activity. A cine-MRI brain image is constructed by averaging the MRI signal acquired during a period of RR ECG signal. This technique allows good visualization of the movements of the walls of the third ventricle and the turbulent flow of cerebro-spinal fluid (LCS) in the cardiac cycle. The details of this technique are presented in [8]
3. The parameters of the acquisition are given in Table 5.2.
4. Each sequence is composed of 20 frames. These frames are obtained using the Cerebmov tool. It allows the user to load a sequence of Cine-MRI images, select a

Table 5.2 Parameters of the MRI acquisition

Parameters	Values
TR/TE	78 ms/3 ms
Number of excitations	7
Flip angle	82
Field of view	160 mm
Matrix size	256 × 256
Slice thickness	2.5 mm
Acquisition time	From 2 to 4 min
Phases during the R to R interval	20

region of interest and then quantify its movements and see the curves of displacement of each point of the contour. It contains six steps that are: setup, resolution, reference, preprocessing, segmentation and movement. In conclusion, it allows the assessment and quantification of the movement of the walls of the third cerebral ventricles. Indeed, it may facilitate diagnosis of brain diseases. For more details about the tool, you can see [23].

5.5.3 Tracking Real Dynamic Membrane of the Third Ventricle

For consistency, tracks for Kalman filter, Extended Kalman Filter and Evidential Particle Filter are shown in the above figures with black stars and the true state is represented with white circles. To better illustrate the tracks, we have zoomed in to the portion of the frame which represented the ROI. The images in the Figs. 5.11 and 5.14 are sequentially presented from left to right the tracking results obtained thanks to the Kalman Filter Algorithm and using the parameters defined before. From these figures, we see that the Kalman Filter Algorithm stays close to the right positions of the dynamic membrane's contour points in most frames.

Figure 5.12 shows example frames from the experiments' sequences with the tracking results overlaid for the first region of interest. Additionally, Fig. 5.15 illustrates example results of the Extended Kalman Filter for the second region of interest. In these cases, the Extended Kalman Filter performs worse than the Kalman filter and it loses track of the third ventricle's dynamic membrane. More accurate results have been obtained in some other experiments with the two other data sets, this is can be illustrated in Figs. 5.21 and 5.22.

Figures 5.13 and 5.16 provide the tracking results for the two ROI by the Evidential Particle Filter. As expected and as it can be seen, the performance of our algorithm is better than the one of the Kalman filter and the Extended Kalman Filter. Our proposal

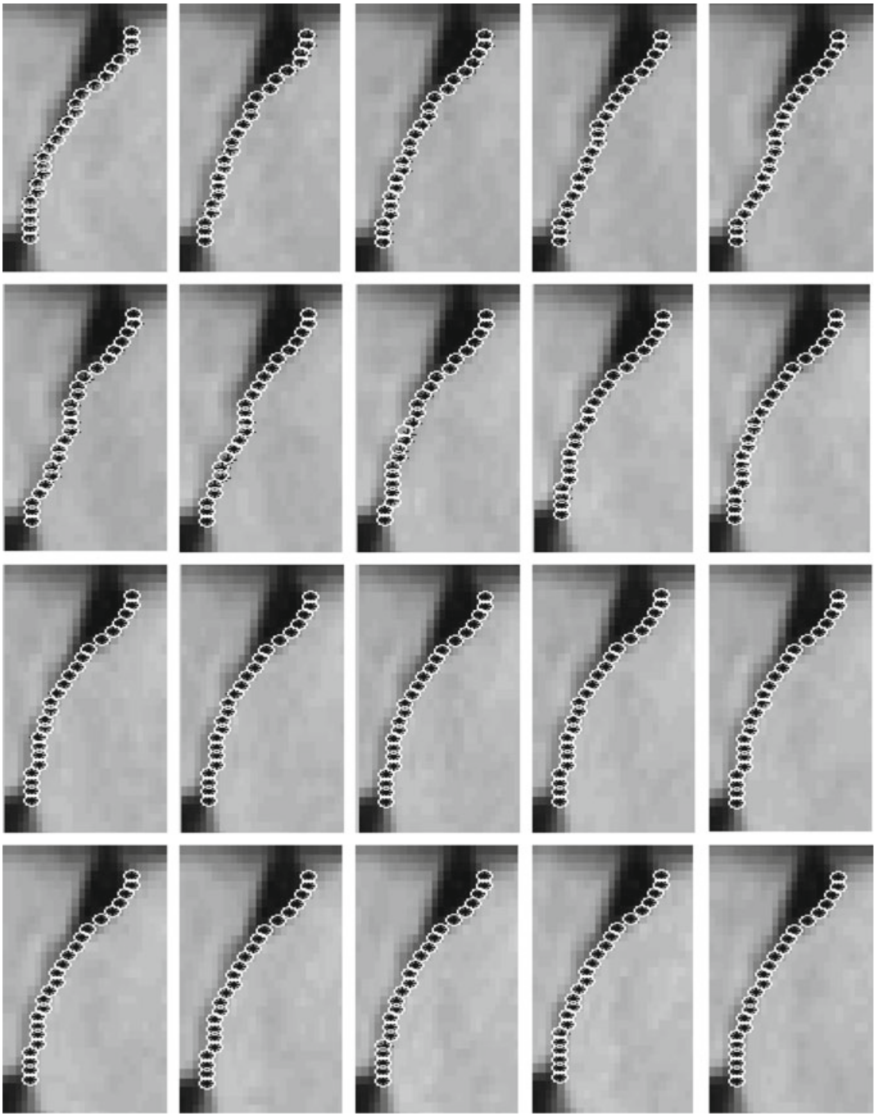


Fig. 5.11 Illustration of the first image of the original sequence with its first ROI and tracking results of the Kalman Filter algorithm for the first region of interest ROI1

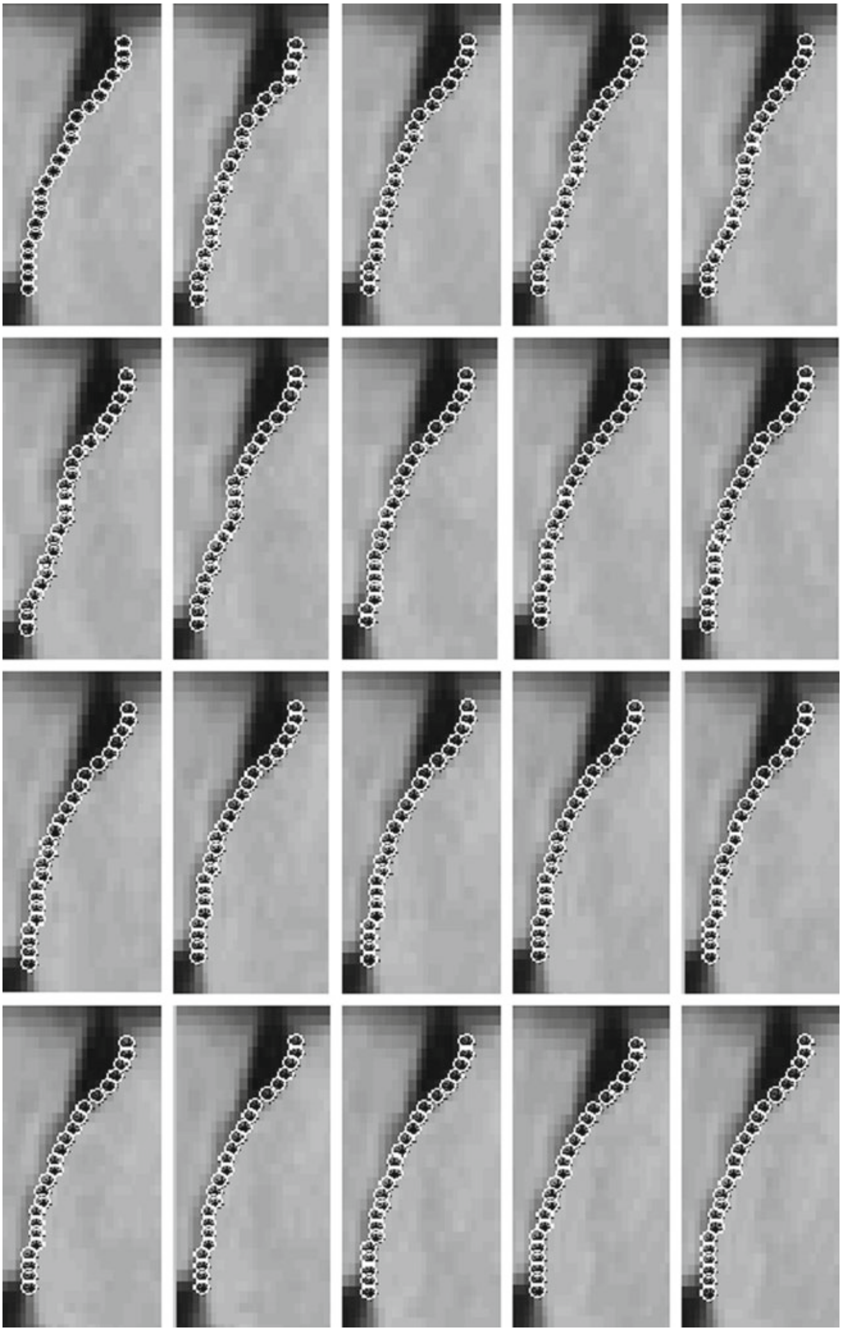


Fig. 5.12 Illustration of the first image of the original sequence with its first ROI and tracking results of the Extended Kalman Filter algorithm for the first region of interest ROI1

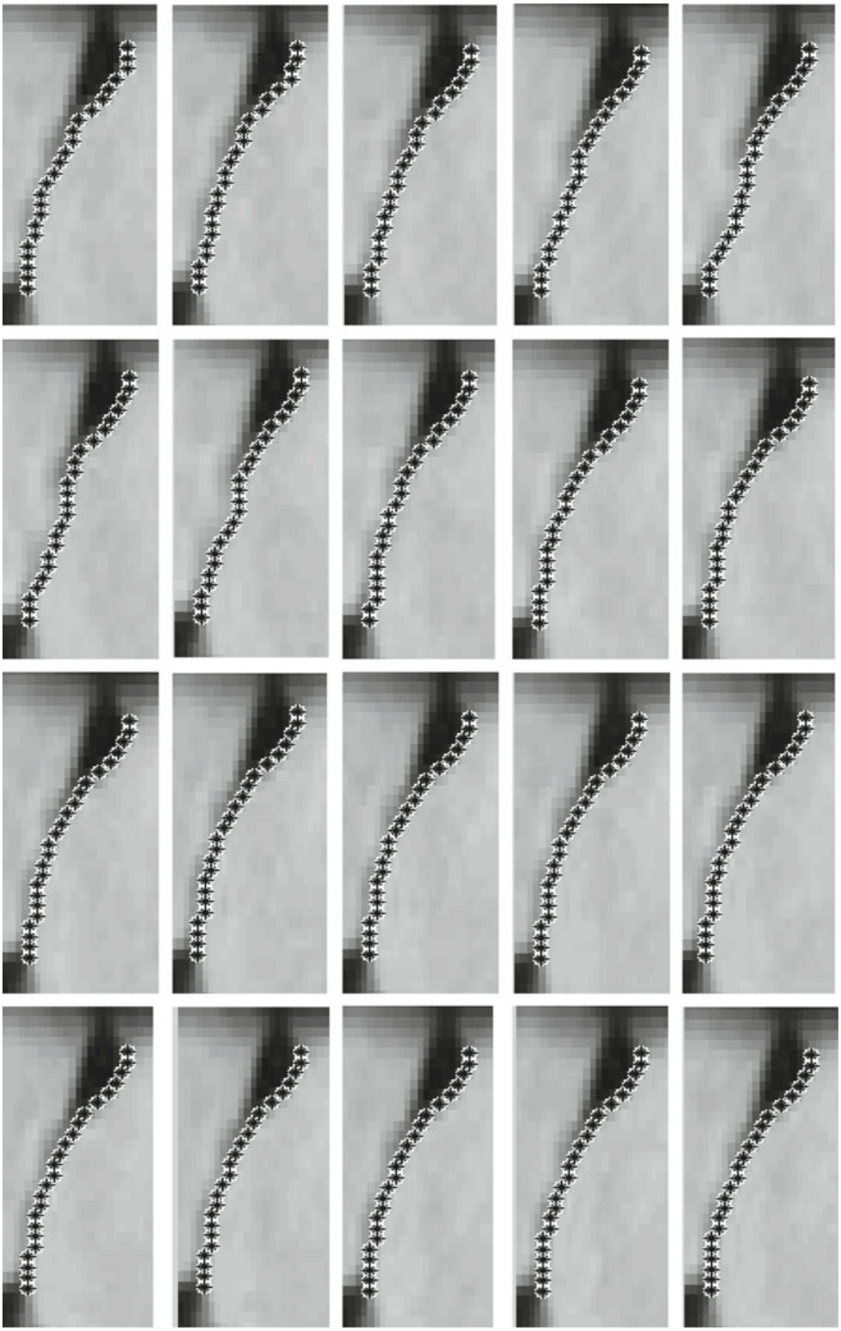


Fig. 5.13 Illustration of the first image of the original sequence with its first ROI and tracked outputs of the Evidential Particle Filter algorithm for the first region of interest ROI1

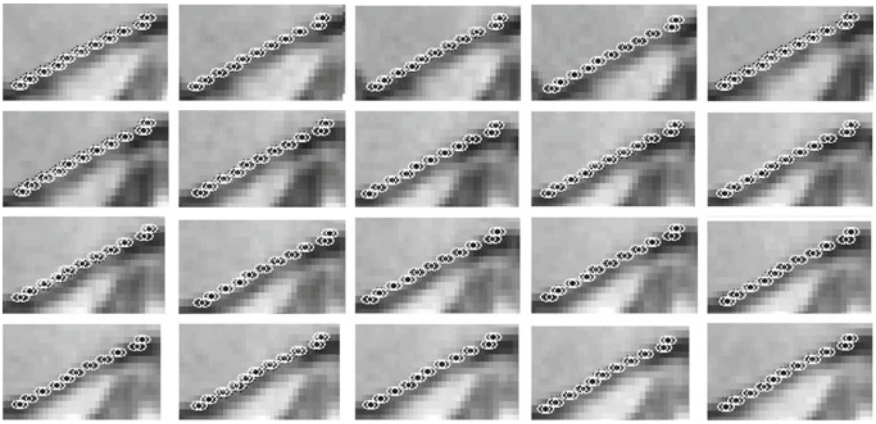


Fig. 5.14 Illustration of the first image of the original sequence with its second ROI and tracking results of the Kalman Filter algorithm for the second region of interest ROI2

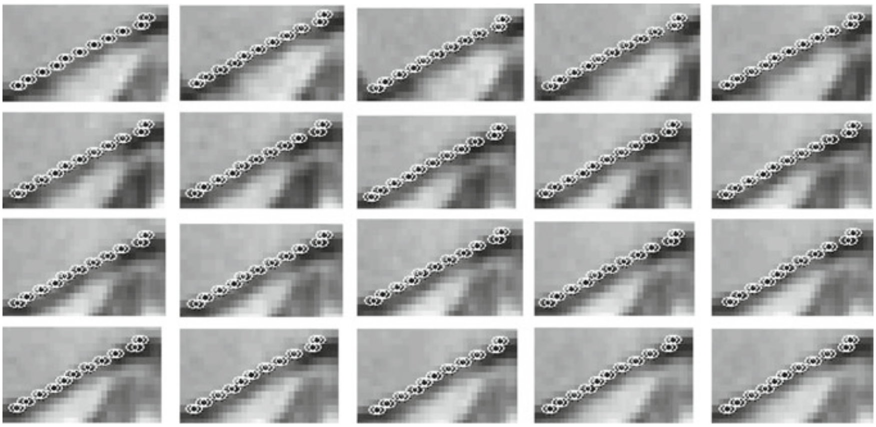


Fig. 5.15 Illustration of the first image of the original sequence with its second ROI and tracking results of the Extended Kalman Filter algorithm for the second region of interest ROI2

stays closest to the ground truth till the very end and more accurate than the Kalman Filter's results (Figs. 5.14 and 5.15).

5.5.4 Tracking Accuracy

To show the efficiency and accuracy of our tracking solution, we have calculated the Mean Square Error (MSE), the Mean error, the Median error and the Standard Deviation (SD) error of the estimated state vector in *X* direction and *Y* direction for

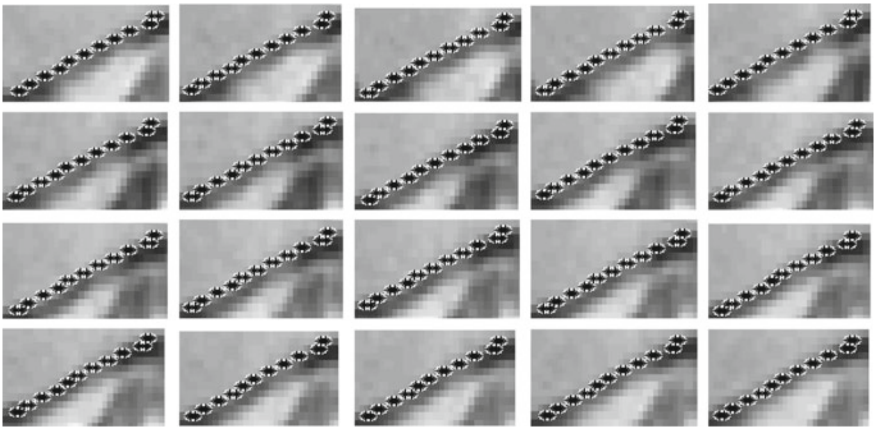


Fig. 5.16 Illustration of the first image of the original sequence with its second ROI and tracked outputs the Evidential Particle Filter algorithm for the second region of interest ROI2

each data set. Each one has its own number of point to track in each region of interest. For the first region, 24 points is associated to the first data set DS1, 34 points for the data set 2 DS2 and 36 points for the data set 3 DS3, while the second region has 24 points for the data set 1 DS1 36 points for the data set 2 DS2 and 34 points for the data set 3 DS3. We estimated the state using our proposed method, Evidential Particle Filter for Complex Shape Tracking (EPF-CST) shown in Black, Extended Kalman Filter (EKF) shown in blue, red and green for DS1, DS2 and DS3 respectively, and the Kalman Filter (KF) in orange, magenta and light green (Fig. 5.16).

For each moving region (Data Set (DS)), in both direction, a state error vector is calculated at t in the following manner:

$$X^t_{error} = \begin{pmatrix} \frac{\hat{x}_1 - x_1}{x_1} \\ . \\ . \\ \frac{\hat{x}_i - x_i}{x_i} \\ . \\ . \\ \frac{\hat{x}_n - x_n}{x_n} \end{pmatrix} Y^t_{error} = \begin{pmatrix} \frac{\hat{y}_1 - y_1}{y_1} \\ . \\ . \\ \frac{\hat{y}_i - y_i}{y_i} \\ . \\ . \\ \frac{\hat{y}_n - y_n}{y_n} \end{pmatrix}$$

where (x_i, y_i) are the right coordinates of the i^{th} point at t and $(\hat{x}_i, \hat{y}_i)$ are the estimated coordinates at t.

Figures 5.17 and 5.18 show the mean error in state estimation. The mean error in X direction (Y direction) at t is the arithmetic average of X^t_{error} elements (Y^t_{error} elements), computed by dividing the total of all values by the number of values. As we can see, at all time-instants, the state estimation error using our method remains null. On the other hand, the KF remains close to the true state in most time. In X

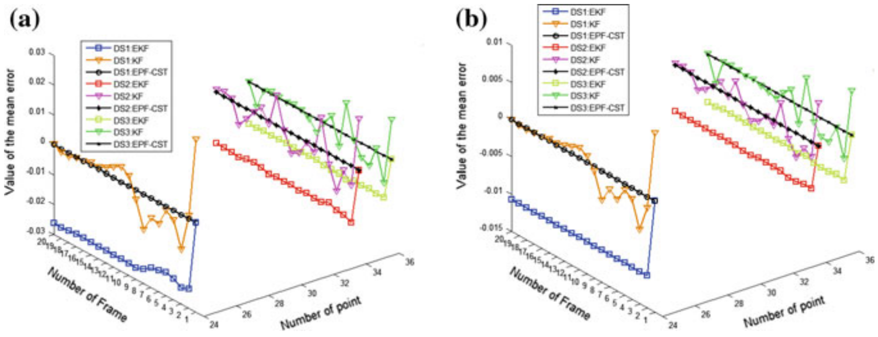


Fig. 5.17 Mean state estimation error of ROI1 in **a** X direction **b** Y direction across three data sets (DS1: Data Set 1, DS2: Data Set 2, DS3: Data Set 3) using our proposal the Evidential Particle Filter for Complex Shape Tracking (EPF-CST), Extended Kalman Filter (EKF) and Kalman Filter (KF)

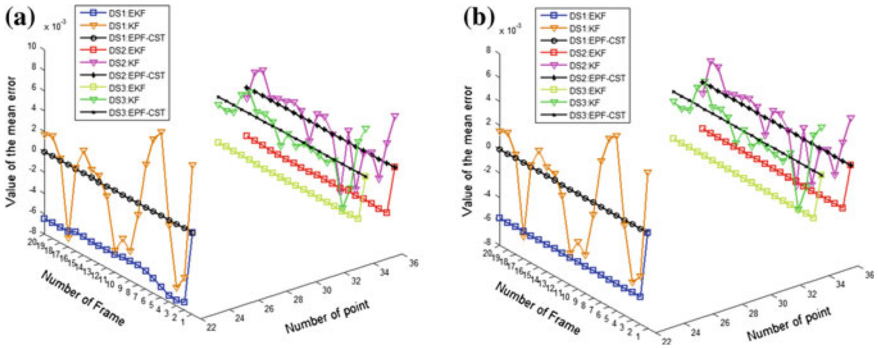


Fig. 5.18 Mean state estimation error of ROI2 in **a** X direction **b** Y direction across three data sets (DS1: Data Set 1, DS2: Data Set 2, DS3: Data Set 3) using our proposal the Evidential Particle Filter for Complex Shape Tracking (EPF-CST), Extended Kalman Filter (EKF) and Kalman Filter (KF)

direction and for the ROI1, the mean error didn't exceed ±0.02 for the three data sets. In Y direction, the mean error is more reduced and it is in ±0.005. For the second region ROI2, the KF performs with the same manner. In fact, the mean error in X direction is greater than the mean error in Y direction. The EKF yields a higher mean error in both directions compared to the KF for all the tested moving region.

Similar to the above evaluation, Figs. 5.19 and 5.20 show the median error in state estimation, for the different initializations of the initial state, x_0, for the same dynamic contour regions. As we can see, the norm of the state error is very large initially for the Kalman Filter, in some case it is equal to 0.02 for the X direction, but for our proposed method, as time proceeds the state error is steady and equal to zero. In beginning the Kalman Filter diverge from the true state then, as time proceeds, it converges to the right state. In most cases, the EKF have the median error value

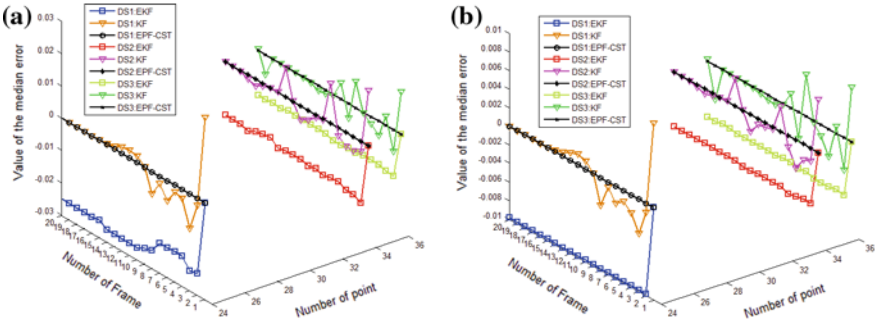


Fig. 5.19 Median state estimation error of ROI1 in **a** X direction **b** Y direction across three data sets (DS1: Data Set 1, DS2: Data Set 2, DS3: Data Set 3) using our proposal the Evidential Particle Filter for Complex Shape Tracking (EPF-CST), Extended Kalman Filter (EKF) and Kalman Filter (KF)

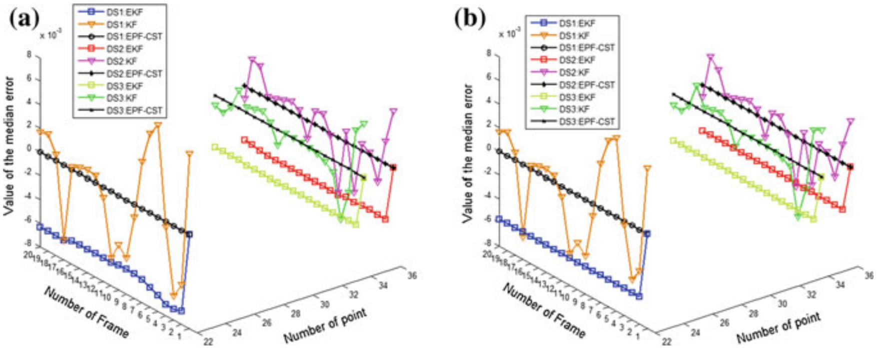


Fig. 5.20 Median Error state estimation error of ROI2 in **a** X direction **b** Y direction across three data sets (DS1: Data Set 1, DS2: Data Set 2, DS3: Data Set 3) using our proposal the Evidential Particle Filter for Complex Shape Tracking (EPF-CST), Extended Kalman Filter (EKF) and Kalman Filter (KF)

remains constant for each separate case in both directions. The values varied in the gap of $[-0.03, 0]$ for the first region ROI1 and in the gap of $[-0.008, 0]$ for the second region ROI2

The plots of the MSE for ROI1 are shown in Fig. 5.21 and for ROI2 they are shown in Fig. 5.22. The MSE is given by:

$$MSE(n) = 1/n \sum_{k=1}^{n} \|x(k) - \hat{x}(k)\|^2 \tag{5.9}$$

where $\hat{x}(k)$ is the predicted state and the $x(k)$ is the true state. Then we have compared them together.

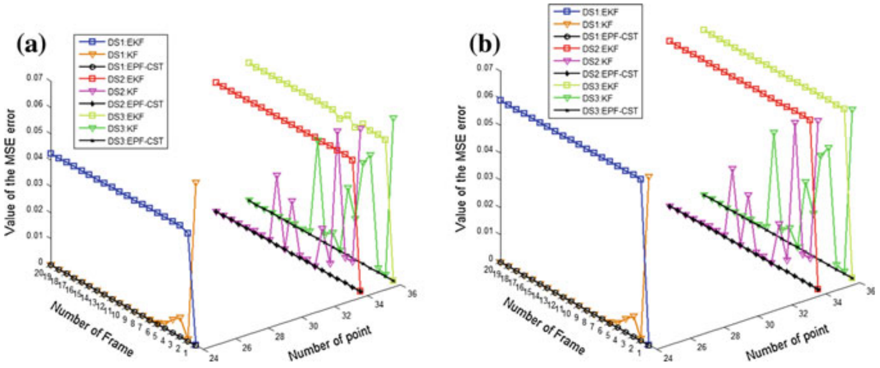


Fig. 5.21 Mean Square Error state estimation error of ROI1 in **a** X direction **b** Y direction across three data sets (DS1: Data Set 1, DS2: Data Set 2, DS3: Data Set 3) using our proposal the Evidential Particle Filter for Complex Shape Tracking (EPF-CST), Extended Kalman Filter (EKF) and Kalman Filter (KF)

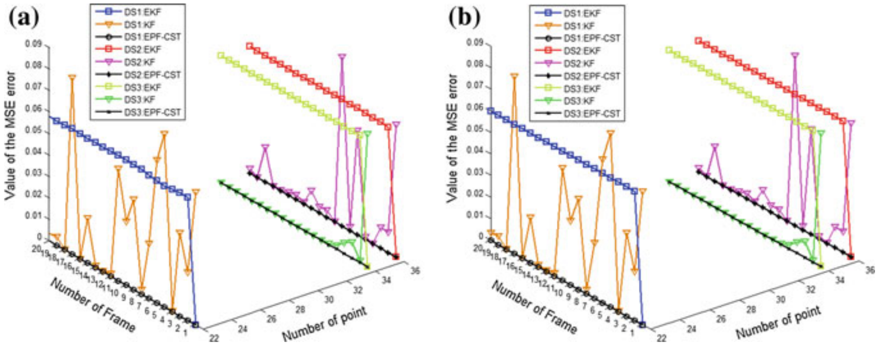


Fig. 5.22 Mean Square Error state estimation error of ROI2 in **a** X direction **b** Y direction across three data sets (DS1: Data Set 1, DS2: Data Set 2, DS3: Data Set 3) using our proposal the Evidential Particle Filter for Complex Shape Tracking (EPF-CST), Extended Kalman Filter (EKF) and Kalman Filter (KF)

In these plots, it can be observed that for the models described before, our method has the smallest MSE. Kalman Filter may also have a good estimation as ours but the difference may be clear in the state estimation of the second and third data set DS2 and DS3 in both directions for the first region ROI1. For the second region ROI2, the difference is clear in the state estimation of the first and third data set DS1 and DS3 in both directions. In the other cases, the KF diverge at beginning from the true state then the its MSE error converges to zero.

On the other hand and for ROI1, the Extended Kalman Filter starts up with very good estimation of the state, then it diverges to a constant value in the gap of [0.04, 0.06]. For the second region ROI2, the EKF failed in all cases also. It fact, it was exactly behaving as its behaviour for the first region ROI1.

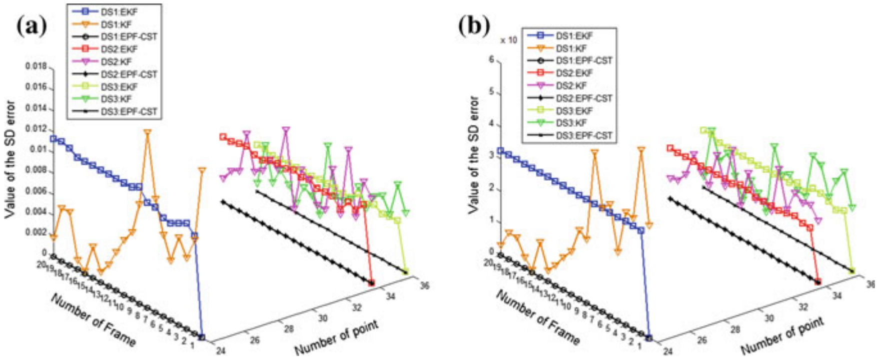


Fig. 5.23 Standard Deviation state estimation error of ROI1 in **a** X direction **b** Y direction across three data sets (DS1: Data Set 1, DS2: Data Set 2, DS3: Data Set 3) using our proposal the Evidential Particle Filter for Complex Shape Tracking (EPF-CST), Extended Kalman Filter (EKF) and Kalman Filter (KF)

The standard deviation error s of the data vector X^t_{error} is defined as follow:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{5.10}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and n is the number of elements in the data vector X^t_{error}. It provides an indication of how spread out the individual in X^t_{error} (also Y^t_{error}) i.e. are they concentrated around the mean, or scattered far and wide.

For the first class of moving region ROI1 Fig. 5.23 shows the shape of distribution of the error values in X^t_{error} (also Y^t_{error}). From these shapes we note that: for the data set 1 in both directions, the individual data values of the KF estimation is closer than the individual data value of the EKF to the mean value. This doesn't contradict the fact that the error value of the KF at the first frames scatter far and wide the mean value then it becomes more contracted around the mean value. For the two other data sets, we can't affirm the same interpretation. Indeed, the individual data value of the EKF more or less constant compared to the individual data values of the KF estimation. Likewise, for the second class of the moving region ROI2, Fig. 5.24 the same interpretation can be done but for both KF and EKF the values of the SD are more concentrated around the mean value. we can interpret also that the individual data values of the EKF estimation is closer than the individual data value of the KF to the mean value in Y direction. Summarizing, the above evaluation shows that the proposed tracker performs favorably against the state of the art algorithms (Kalman Filter and Extended Kalman Filter) as it carries out the best performance and in practice it performs very well and always converge to the true state. The Extended Kalman Filter performs worse than the Kalman Filter almost in all runs.

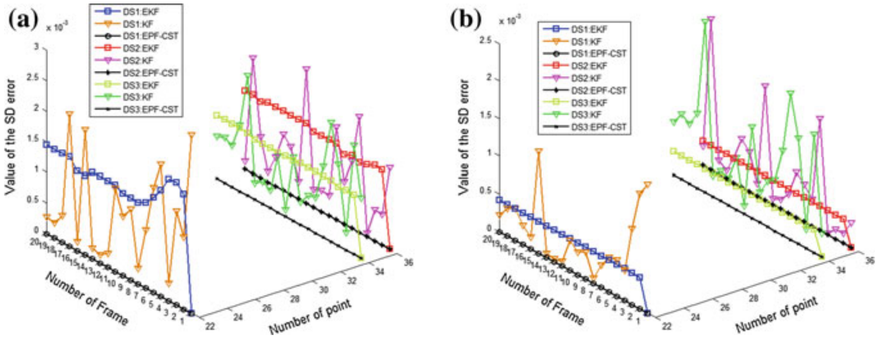


Fig. 5.24 Standard Deviation Error state estimation error of ROI2 in **a** X direction **b** Y direction across three data sets (DS1: Data Set 1, DS2: Data Set 2, DS3: Data Set 3) using our proposal the Evidential Particle Filter for Complex Shape Tracking (EPF-CST), Extended Kalman Filter (EKF) and Kalman Filter (KF)

5.6 Conclusion

In this paper, we proposed a new approach the evidential particle filter for objects'contour tracking with complex shapes, specifically the contour of the third cerebral ventricle tracking. The nature of our sequences induces specific sources of information issues regarding notably highly imprecise or unreliable cine-RMI images. The proposed particle filter algorithm for complex shape tracking makes use of the conception of the Particle Filter and evidence theory. First of all, the integration of the curve evolution in the update step makes a good use of the observation model at current time t. Second, an evidential modeling step is achieved. It is based on the integration of the fusion of two heterogeneous features which are the locality sensitive histograms and the energy functional. Throughout our successive experiments, we noticeably demonstrated that our tracking method outperforms state-of-the art approaches in an Cine-RMI and synthetic sequences. Besides our method is robust against noise in the sequences related to the RMI images. Hence, for future works, it is intended to use the evidential particle filter for objects'contour tracking with complex shape complex in a multiple object tracking context. The latter would be used for the different regions of interest which are the ROIs of the third ventricle cerebral contour and some other tracking objects, such as person tracking and vehicle tracking.

5.7 Annex

5.7.1 Bayesian Filtering

In the Bayesian filtering, the state of the object at time t is modeled by $X_t, t = 1, .., n$ which contains all the object properties such as the position, velocity, acceleration. In order to estimate the state X_t, the Bayesian filtering recursively calculates some degree of belief using different measurement information given up to time t and prior state values. It requires so the construction of two stochastic models:

- The dynamic model: describes the state evolution via:

$$X_t = f_t(X_{t-1}) + W_t \tag{5.11}$$

  where W_t is a white noise.
- The measurement model: describes the relationships between the measurement and the state via:

$$Y_t = h_t(X_t, N_t) \tag{5.12}$$

  where N_t is another white noise independent of W_t.

Besides these models, a set of densities is defined as follow:

- $P(X_t|X_{t-1})$: the transition density function which describes the dynamic model
- $P(Y_t|X_t)$: the observation density function conditionally independent given the states which describes the measurement model.
- And the distribution of interest is the filtering distribution P($X_t|Y_{1..t}$) which can be computed using two recursion steps:

1. The prediction step: here is the use of the dynamic model and the probability density function at $t - 1$ $P(X_t|Y_{1..t-1})$ to derive the prior probability density function of the current state t P($X_{1..t}|Y_{1..t}$) based on:

$$P(X_t|Y_{1..t}) = \int_{X_{t-1}} P(X_t|X_{t-1})P(X_{t-1}|Y_{1..t-1})d_{X_{t-1}} \tag{5.13}$$

2. The update step: employs the prior probability density function arising from the prediction step and the likelihood function P($Y_t|X_t$) of the current measurement at t to calculate the posterior probability density function P($X_t|Y_{1..t}$).

$$P(X_t|Y_{1..t}) = \frac{P(Y_t|X_t, Y_{1..t-1}) * P(X_t|Y_{1..t-1})}{P(Y_t|Y_{1..t-1})} \tag{5.14}$$

5.7.2 Kalman Filter

In [12], R.E Kalman published the origin of the concepts Linear filtering and prediction problems. Then his research led him to describe a process that will be known as the Kalman filter.

The Kalman filter is a set of mathematical equations that provides a better estimate of the future state of a system despite the imprecision measurements and modeling.

The Kalman filter is very powerful through the use of the control in feedback: it considers the state of the system and improves its results with noisy information provided by the observations system. In the prediction phase, it estimate the state x_t at t using the state's updated estimation x_{t-1} at $t-1$ using

$$\tilde{x}_t = F * x_{t-1} + G * u_t + \xi_t \tag{5.15}$$

where ξ_t is sequences of white, zero mean, Gaussian noise, F, G are know $p \times p$ and $p \times r$ matrices. F is the state transition matrix. G is the control input matrix converts the control vector u_t into state space. u_t is the vector containing any control inputs. Besides to the in the prediction step, Kalman filter calculate the Error Covariance Prediction using

$$P_t^- = F * P_{t-1} * F^T + Q \tag{5.16}$$

where P_{t-1} is a matrix representing the error covariance in the state prediction at $t-1$. Q is the process noise covariance matrix. In Sect. 5.5.3, we will mention the chosen model of moving object and the values used for the both matrix.

After predicting the state $\tilde{x}_t$ and its error covariance. The Kalman filter use the Kalman Gain K_t to correct the state estimated $\tilde{x}_t^-$ in the last step using:

$$K_t = P_t^- * H^T (H * P_t^- * H^T + R_t)^{-1} \tag{5.17}$$

where R is the measurement noise covariance and H is a matrix converting the state space into measurement space

The state is then updated using

$$\tilde{x}_t = \tilde{x}_t^- + K_t (y_t - H * \tilde{x}_t^-) \tag{5.18}$$

where y_t is the measurement, generally derived from sensors:

$$y_t = H * x_t + \delta_t \tag{5.19}$$

where δ_t: is sequences of white, zero mean, Gaussian noise.

The final step of the kalman filter is to update the error covariance P_t^- using:

$$P_t = (I - K_t * H) P_t \tag{5.20}$$

The performance of the kalman filter depends on these factors:

1. The value of the prediction error covariance: In fact, the higher is this later, the more efficient is the predicted state.
2. The modeling of the process act on the value of the Prediction error covariance. This last will be high if the process is not precisely modeled, so the entries of Q are fairly high.
3. The value of the updated error covariance: In fact, this later decrease if the measurements are accurate.

As such we can interpret that the equations of the Kalman filter can be classified in two classes:

1. Time update equations or predictor equations: are responsible to obtain the a priori estimates for the next time step.
2. Measurement update equations or corrector equations: are responsible to obtain an improved a posteriori estimate by incorporating a new measurement into the a priori estimate.

5.7.3 Extended Kalman Filter

As we mention before, when we want to apply a Kalman filter to estimate the parameters of a system, the first thing to do is to model the problem. It turns out that in some cases the equations that model the problem are not linear. Thus, the Kalman filter is not applicable as it is. The Extended Kalman Filter is the non-linear version of the Kalman Filter. This filter makes it possible to locally linearize the problem and therefore to apply the equations of classical Kalman filter. The state equations and the equations linking the previous state to the next one at $t + 1$ which were linear in the case of conventional Kalman filter is now nonlinear. It is impossible to write them in a matrix form and are replaced by:

$$x_t = f(x_{t-1}, u_{t-1}) + w_{t-1} \tag{5.21}$$

$$y_t = h(x_t) + v_t \tag{5.22}$$

where w_t and v_t are the process and the observation noises which are both assumed to be zero mean multivariate Gaussian noise with covariance Q_t and R_t, respectively. In the prediction phase, the state is predicted using:

$$\tilde{x}_t^- = f(x_{t-1}, u_{t-1}) \tag{5.23}$$

and the predicted covariance matrix is estimated by using:

$$P_t = F_{t-1} P_{t-1} F_{t-1}^T + Q_{t-1} \tag{5.24}$$

The predicted state is updated using

$$\hat{x}_t = \hat{x}_t^- + K_t \tilde{z}_t \tag{5.25}$$

where K_t is the optimal Kalman gain, $\tilde{z}_t$ is the measurement residual and S_t is the residual covariance

$$K_t = P_t H_t^T S_t^{-1} \tag{5.26}$$

$$\tilde{z}_t = y_t - h(\hat{x}_t) \tag{5.27}$$

$$S_t = H_t P_t H_t^T + R_t \tag{5.28}$$

To use these equations, it is necessary that the functions f and h are locally linearized. Thus, the transition matrix and the observation matrix are obtained by taking the partial derivative of the nonlinear equations. The state transition and observation matrices are defined to be the following Jacobians

$$F_{t+1} = \frac{\partial f(t, x_t)}{\partial x}|_{x=\hat{x}_t} \tag{5.29}$$

$$H_t = \frac{\partial h(t, x_t)}{\partial x}|_{x=\hat{x}_t} \tag{5.30}$$

As mentioned before, the linearisation of the equations is local which causes a local convergence of the extended Kalman filter. This filter doesn't guarantee, hence, a global convergence unlike the Kalman filter. Therefore, it is harder to reach the stability of the Extended Kalman Filter, and it usually depends on a convenient initialization.

5.7.4 Particle Filter

Recently Particle Filter has become more and more popular. The Particle Filter is a sequential MC approach. It was first introduced in [5] as the Bayesian Bootstrap filter and it was first applied to object tracking in computer vision and known as the Condensation algorithm [9].

The Monte Carlo (MC) method [2] constitutes the basis for most sequential MC filters. This Sequential MC is a method to implement a recursive Bayesian filter by MC simulation. The main objective is to represent the conditional state density P($X_{1..t}|Y_{1..t}$) at time t by a set of particles with weights $\{x_t^i, w_t^i\}_{i=1}^N$.

The weights define the importance of a particle, that is, its observation frequency. The new set of particles at time t are drawn from $X_{t1} = \{(x_{t-1}^i, w_{t-1}^i i = 1, .., N\}$ and a cumulative weight c_{t-1} to decrease the computational complexity at the previous time $t-1$ step based on different sampling schemes [20]. The Importance sampling is one of the most common sampling schemes and it can be resumed in three steps as follows:

- Selection: Select N random particles from X_{t-1} by generating a random number $r \in [0, 1]$, finding the smallest j such that $c_{t-1}^{(j)} > r$ and $\hat{x}_t^{(n)} = x_{t-1}^{(j)}$
- Prediction: Generate a new particle for each particles $\hat{x}_t^{(n)}$ using $x_t^{(n)} = f(\hat{x}_t^{(n)}, W_t^{(n)})$
- Correction: Computing the weights $w_t^{(n)} = p(y_t|x_t = x_t^{(n)})$ where y_t is the measurement at t and $p(.)$ can be modeled as a Gaussian density.

References

1. D. Angelova, L. Mihaylova, Extended object tracking using monte carlo methods. Trans. Sig. Proc. **56**(2), 825–832 (2008)
2. M.S. Arulampalam, S. Maskell, N. Gordon, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans. Signal Process. **50**, 174–188 (2002)
3. D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
4. T. Denux, An evidence-theoretic neural network classifier, in *IEEE International Conference on Systems, Man and Cybernetics* (1995), pp. 712–717
5. N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/non-gaussian bayesian state estimation. IEEE Proc. F Radar Signal Process. **140**(2), 107–113 (1993)
6. F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.-J. Nordlund, Particle filters for positioning, navigation, and tracking. Trans. Sig. Proc. **50**(2), 425–437 (2002)
7. S. He, Q. Yang, R.W.H. Lau, J. Wang, M.-H. Yang, Visual tracking via locality sensitive histograms, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 2427–2434
8. J. Hodel, P. Decq, A. Rahmouni, S. Bastuji-Garin, A. Maraval, C. Combes, B. Jarraya, C. Le Guérinel, A. Gaston, Brain ventricular wall movement assessed by a gated cine mr truefisp sequence in patients treated with endoscopic third ventriculostomy. Eur. Radiol. **19**(12), 2789–2797 (2009)
9. M. Isard, A. Blake, Condensation - conditional density propagation for visual tracking. Int. J. Comput. Vis. **29**, 5–28 (1998)
10. J.D. Jackson, A.J. Yezzi, S. Soatto, Tracking deformable moving objects under severe occlusions, in *IEEE CDC* (2004)
11. U.K. Jaliya, H.S. Parekh, D.G. Thakore, A survey on object detection and tracking methods, in *International Journal of Innovative Research in Computer and Communication Engineering*, 2, Feb 2014
12. R.E. Kalman. A new approach to linear filtering and prediction problems, in *ASME Journal of Basic Engineering* (1960)
13. N.K. Kanhere, S.T. Birchfield, Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features, in *IEEE Transactions on Intelligent Transportation Systems* (2008), pp. 148–160
14. M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models. Int. J. Comput. Vis. **1**(4), 321–331 (1988)
15. R. Kennes, P. Smets, Computational aspects of the mobius transform, in *CoRR* (2013). arXiv:1304.1122

16. V. Kurtcuoglu, M. Soellinger, P. Summers, K. Boomsma, D. Poulikakos, P. Boesiger, Y. Ventikos, Computational investigation of subject-specific cerebrospinal fluid flow in the third ventricle and aqueduct of sylvius. J. Biomech. **40**(6), 1235–1245 (2007)
17. V. Kurtcuoglu, M. Soellinger, P. Summers, D. Poulikakos, P. Boesiger, Mixing and modes of mass transfer in the third cerebral ventricle: a computational analysis. J. Biomech. Eng. **129**(5), 695–702 (2007)
18. S. Lankton, S. Member, A. Tannenbaum, A.: localizing region-based active contours, in *IEEE Transaction on Image Processing* (2008), pp. 2029–2039
19. J. MacCormick, A. Blake, A probabilistic exclusion principle for tracking multiple objects. Int. J. Comput. Vis. **39**(1), 57–71 (2000)
20. D.J.C. MacKay, Introduction to monte carlo methods, in *Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models* (Kluwer Academic Publishers, Norwell, MA, USA, 1998), pp. 175–204
21. A. Mayer, H. Greenspan, An adaptive mean-shift framework for mri brain segmentation. IEEE Trans. Med. Imaging **28**(8), 1238–1250 (2009)
22. A. Nakib, F. Aiboud, J. Hodel, P. Siarry, P. Decq, Third brain ventricle deformation analysis using fractional differentiation and evolution strategy in brain cine-mri, in *SPIE medical imaging* (International Society for Optics and Photonics, 2010), pp. 76232I–76232I
23. A. Nakib, P. Siarry, P. Decq, A framework for analysis of brain cine mr sequences. Comput. Med. Imaging Graph. **36**(2), 152–168 (2012)
24. M. Niethammer, A. Tannenbaum, Dynamic geodesic snakes for visual tracking, in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, vol. 1 (IEEE, 2004), pp. I–660
25. Dr. D. Nitnawwre, R. Mishra, M.K. Chouhan, Multiple object tracking by kernel based centroid method for improve localization, in *International Journal of Advanced Research in Computer Science and Software Engineering* (2012), pp. 137–140
26. L.M. Novak, Optimal target t designation techniques. IEEE Trans. Aerosp. Electron. Syst. **5**, 676–684 (1981)
27. N. Paragios, R. Deriche, Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Trans. Pattern Anal. Mach. Intell. **22**(3), 266–280 (2000)
28. M. Pupilli, A. Calway, Real-time camera tracking using a particle filter, in *Proceedings of British Machine Vision Conference* (2005), pp. 519–528
29. Y. Rathi, N. Vaswani, A. Tannenbaum, A. Yezzi, Particle filtering for geometric active contours with application to tracking moving and deforming objects, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 2 (IEEE, 2005), pp. 2–9
30. S Saravanakumar, A Vadivel, CG Saneem Ahmed, Multiple human object tracking using background subtraction and shadow removal techniques, in *IEEE International Conference on Signal and Image Processing (ICSIP), 2010* (IEEE, 2010), pp. 79–84
31. G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, 1976)
32. C. Shen, J. Kim, H. Wang, Generalized kernel-based visual tracking. IEEE Trans. Circuits Syst. Video Technol. **20**(1), 119–130 (2010)
33. P. Smets, Quantifying beliefs by belief functions: an axiomatic justification, in *Proceedings of the 13th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'93* (Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993), pp. 598–603
34. P. Smets, Decision making in the tbm: the necessity of the pignistic transformation. Int. J. Approx. Reason. **38**(2), 133–147 (2005)
35. P. Smets, R. Kennes, The transferable belief model. Artif. Intell. **66**(2), 191–234 (1994)
36. P. Smets, R. Kruse, The transferable belief model for belief representation, in *Uncertainty Management in Information Systems* (1996), pp. 343–368
37. H. Tao, H.S. Sawhney, R. Kumar, Object tracking with bayesian estimation of dynamic layer representations. IEEE Trans. Pattern Anal. Mach. Intell. **24**(1), 75–89 (2002)
38. D. Terzopoulos, R. Szeliski, Tracking with kalman snakes, in *Active vision* (MIT press, 1993), pp. 3–20

39. N. Vaswani, A. Roy Chowdhury, R. Chellappa, Activity recognition using the dynamics of the configuration of interacting objects, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003*, vol. 2 (IEEE, 2003), pp. II–633
40. A. Yilmaz, X. Li, M. Shah, Contour-based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans. Pattern Anal. Mach. Intell. **26**(11), 1531–1536 (2004)
41. A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey. ACM Comput. Surv. **38**(4), 13 (2006)
42. L.M. Zouhal, T. Denoeux, An adaptive k-nn rule based on Dempster–Shafer theory, in *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns (CAIP'95* (Springer, 1995), pp. 310–317

Chapter 6
Microscopic Image Segmentation Based on Based Branch and Bound and Game Theory

Amira Kouzana, Amir Nakib and Narjes Dogaz

Abstract In this work a new family of image segmentation algorithms is proposed. This paper is a generalization of the model proposed, called: Power Watershed segmentation framework. Indeed, we extended it for cases: $2 < q < \inf$ and $p \to \inf$. To do so, we explore the segmentation a new formulation of the segmentation problem based on game theory is proposed optimization energy function as a game theory problem. In this new formulation, The minimization can be, then, optimization process is seen as a search of the Nash equilibrium of a non-cooperative strategic game. Indeed, the computation of Nash equilibrium in finite game is equivalent to a non linear optimization problem afterward. As the optimization problem thus formulated the computation of the Nash equilibrium is an NP-hard problem, then, we propose the use of the Branch and Bound method is used to solve it to find it in reasonable time. In this study moreover, the uniqueness of the Nash equilibrium is demonstrated using a potential game-theoretic approach. Then we propose a new family of segmentation approach with $2 < q < \inf$ and $p \to \inf$, named Game-based PW. The obtained results of the proposed approach, show are better than those given by the original Power Watershed $q = 2$.

6.1 Introduction

Segmentation is a fundamental problem in image processing. It consists in partitioning an image into several meaningful objects. A great amount of literature on image segmentation has been published and a lot of techniques have been applied. Among these techniques, the graph-theory was applied to image segmentation problem. The segmentation problem is then solved in a spatially discrete space by the efficient tools from the graph theory [32]. Graph-based segmentation views the image as a graph $G = (V, E)$ where $V = v_1, \ldots, v_n$ is a set of nodes corresponding to the image elements and $E \subseteq V \times V$ is a set of edges, with $n = |V|$ and $m = |E|$. Each edge

A. Kouzana · A. Nakib (✉) · N. Dogaz
Laboratory of LISSI, Université Paris Est, 122 Rue P. Armangot, 94400 Vitry-sur-Seine, France
e-mail: nakib@u-pec.fr

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_6

$e_{ij} = (v_i, v_j)$, connects two neighboring nodes, has a real weight $w_i j = w(v_i, v_j)$ which measures the degree of the connectivity between two nodes such that high value means strong connection between the linked nodes and vice versa. Methods in image segmentation can be divided into two categories: automatic (non-supervised) image segmentation and interactive (supervised). Most segmentation algorithms have focused on automatic segmentation grouping the elements of the image into disjoint subsets according to a criterion such as color or homogeneity. While, interactive methods, or seeded methods, allows to the user the ability to provide prior knowledge about the result to improve the segmentation process. In this case, the user has to mark areas of the image i.e. background and/or object(s) providing. Indeed, many seeded image segmentation methods consider the image as a weighted graph and minimize an energy function on a graph to produce a segmentation result. Among these methods, we can cite: the watershed [3, 12, 13, 28, 34, 41], Kruskal [33], Prim [4] algorithms for finding maximum spanning forest (MSF); Graph-Cuts (GC) based methods were firstly proposed by Boykov and Jolly [4], Random Walker (RW) based is presented in [20], shortest paths [1, 2, 14, 16] introduced by Bai and Sapiro [2] and, recently the Power Watershed (PW) model [11].

It is known that in the topographical surface, the watershed of a function is formed by the locations from which a drop of water could flow towards different minima. In the seeded image segmentation, the gradient of the image can be considered as a relief map and seeds used to specify the segmentation of the image into desired regions. The maximum spanning forest (MSF) is a watershed if the seeds correspond to the maxima [13]. The MSF based algorithm computes trees spanning among all the nodes of the graph, each tree is connected to one connected seed component, and the weight of the set of the trees is maximum. In the case of the Graph Cuts (GC) based algorithms, the produced segmentation is determined by finding the minimum cut between the foreground and the background seeds via a maximum flow computation. The random walker (RW) algorithm can be interpreted as assigning the unlabeled pixels to the seeds where there is a minimum diffusion of the distance [9], i.e. the semi-supervised transduction learning algorithm [15], and the interactive version of normalized cuts [21, 39]. In the case of the shortest path algorithm, each pixel is assigned to the foreground label if the shortest path is from the current pixel to the foreground seed. These paths are weighted by image content in as in the GC and RW approaches.

Several image segmentation works, use game theory to deal with the optimization problem. For instance, in [7], the game-theoretic integration was used to solve an image segmentation problem. A two-person non-zero sum non-cooperative game is, then, used for an integration region-based segmentation and gradient-based boundary finding. This game is played out by a set of decision makers (or players), which correspond to a two segmentation modules to integrate. This leads to a system where the two modules operate in parallel. So, the outputs of each of the modules are feedback to each other after every decision-making step. The game stops when none of the modules can improve their positions, and the final solution pair constitutes the Nash Equilibrium (NE). We recall that a Nash equilibrium is a set of strategies such that each players strategy is an optimal response to the other players strategies [25, 29]. It can be seen as the point where none of the players involved in the game can change their strategies for minimizing the payoff function [42].

In [22], the authors used the Nash-Game approach to joint image restoration and segmentation. The game is a static game with complete information and the considered solution is the Nash equilibrium, which is computed by an iterative method with relaxation. The authors define two players; one is restoration, with the image intensity as strategy, and the other is segmentation with contour as strategy to find the Nash equilibrium. Zeng et al. [24] used the game theory in image segmentation to solve a multi-labeled segmentation problem. The input of this algorithm includes the image to be segmented, as well as multi-labeling preferences called, seeds. The output is the labeling of the input image. In this work, the authors design m parallel games played on m separate layers. Each player on layer l is randomly assigned a strategy of whether cooperate of defect, where one player is assigned to each label. The players play the game on a graph to obtain the minimum energy of the graph via maximize the total payoff.

Shen et al. [38] apply the evolutionary cluster game to image segmentation. In their clustering game image segmentation (CGIS) approach, pixels clusters are derived from the competition of individuals playing the clustering game. This game is a symmetric evolutionary game involving two players. Each player simultaneously selects pixels that should be clustered and, after having revealed his choice, he receives a payoff according to the similarity that the selected pixels have with respect to the opponents' ones. The evolutionary stable equilibrium (ESS) can determine a cluster of pixels in the associated image.

In this work, a new formulation of the segmentation problem based on game theory is proposed. In this new formulation, the optimization process is seen as a search of the Nash equilibrium of a non-cooperative strategic game. Afterwards, the optimization problem thus formulated is an NP-hard combinatorial problem, then, we propose the Branch and Bound method to solve it. Moreover, the uniqueness of the Nash equilibrium is demonstrated using a potential game-theoretic approach. In this formulation, the segmentation problem is considered on a graph involving two players. Each player selects pixels that should be clustered and, after having revealed his choice, he receives a payoff according to the similarity that the selected pixels have with respect to the adversaries ones. The segmentation model considered is a generalization of the model called Power Watershed (PW) [11], to the case of $2 < q < \infty$ and $p \to \infty$. Hence, the segmentation problem is formulated as a non-cooperative strategic-game and the energy minimization problem is solved by Nash equilibrium.

The outline of the paper is as follows: in Sect. 6.2, we recall as preliminaries, the Power Watershed (PW) graph-based segmentation algorithm based on energy minimization. Section 6.3 describes the proposed non-cooperative strategic game formulation of the segmentation problem. In Sect. 6.4, we present the optimization algorithm used to solve the posed problem, called branch and bound. In Sect. 6.5, the experimentations and evaluation of the proposed approach are presented. Finally, we conclude and give suggestions for future work in Sect. 6.6.

6.2 Image Segmentation as a Graph-Cut Problem

The unified graph based segmentation model is known as Power Watershed (PW) and was proposed by Couprie et al. [11]. It extends the common framework for seeded images [40] that included Graph-Cuts, random Walkers and Shortest Path Optimization algorithms. These algorithms consider the image as a graph (G), where each pixel is considered as a node and the edges are weighted to reflect the gradient of intensity, color or other features in the image. In the rest of the paper, we consider the 4-connected case. The PW segmentation model was obtained by introducing a new parameter p in the generalized energy proposed in [40]. Then, the new segmentation model is defined by:

6.3 Formulation of the Segmentation Problem as Non-cooperative Strategic Game

In this section, a new formulation of the segmentation using graph cut and game theory is presented. So, to solve the energy optimization problem formulated in the equation (Eq. 2.2) as a non-cooperative game, the search of the Nash Equilibrium must be done. Indeed, the computation of Nash equilibrium in an n-player non cooperative finite game is equivalent to a non linear optimization problem. In this problem the constraints and objective functions are polynomials. Moreover, the number of variables is equal to the sum of the number of players and the number of the available strategies in the game [8].

Before detailing our approach, some preliminary notions are introduced for more clarity.

A Normal-Form Game:

A (finite, n-person) normal-form game is a tuple (N, X, C), where:

- N is a finite set of n players, indexed by i;
- $X = X_1 \times X_2 \times \ldots X_n$ where X_i is a finite set of actions available to player i. Each vector $x = (x_1, x_2, \ldots, x_n) \in X$ is called: action profile;
- $C = (C_1, C_2, \ldots, C_n)$ where $C_i : X \rightarrow \Re$ is a real-valued utility (or payoff) function for player i.

The game $G = (N, X, C)$ is called finite, if the set N and all sets are finite. Simple non-cooperative games assume that players have complete information. In other terms, all players know the strategies and payoffs available to all other players. Each rational player will select strategy that minimize his own payoff, depending on the strategies selected by other players. This leads to an equilibrium state where the gain of a player does not correspond to a loss of the other players. This kind of a non-cooperative game, called non-zero sum game, can be solved by finding the Nash equilibrium [5, 26].

6.3.1 Formulation of the Segmentation Problem

Let state our segmentation problem that consists of optimizing Eq. 2.2 as a non-cooperative strategic game. We have then, to define the tuple, where N is the set of nodes (pixels) of the plateau. Thus, each node of the plateau is considered as a player. To play the game each player $i \in N$ selects a strategy $x_i \in X_i$. In our case, a strategy for a player i, consists in selecting a value $x_i \in [0, 1]$. This value allows the classification of the node i as a foreground or a background.

Usually, the game theory is based on cost analysis for the payoff function of player i depends on its strategy and those of other players. Then, for the payoff model, we propose to use the global energy defined above as a local energy function $C_i(x)$ specific to the player i that involves its direct neighbors. This choice was motivated by the fact that in game theory the participants follow a strategic interactions, and according to the global energy function of the interaction, and the influence carried on the node i by its direct neighbors. We propose, then, as a payoff function the equation (Eq. 3.1):

Minimizing the equation (Eq. 3.1a) consists, then, in finding the Nash equilibrium. Nash equilibrium [29] is established to be the most important solution concept in Game theory [31]. It represents a steady state of the play of a strategic game where each player has an accurate opinion about the behavior of other players and acts rationally. This equilibrium is a game of strategy, one per player, such that each player minimizes its cost, depending on the activities of the other players. In other words, the Nash equilibrium can be seen as the optimal operating point of the game. Although, there are cases where we can find multiple Nash equilibrium, making the choice of the best one difficult, it was proven that when the game is an exact potential, and its corresponding potential function admits a global minimum; then the convergence to a unique Nash-equilibrium is guaranteed [37].

A strategic-form game is an exact potential game if the incentive of all players to change their strategy can be expressed in one global function, called the games potential [5].

Definition of an Exact Game:

A game $G = (N, X, C)$ is an exact potential game [27, 42], if it exists a function P such that:

$$\begin{aligned} P &: X \to \Re \\ C_i(y_i, x_{-i}) - C_i(z_i, x_{-i}) &= P(y_i, x_{-i}) - P(z_i, x_{-i}) \end{aligned} \tag{6.1}$$

where $x_i \in X_{-i}$ all strategies except that of player i, $\forall x_{-i} \in X_{-i}$, $y_i \in X_i$, $z_i \in X_i$, and $i \in N$.

The proposed potential function P or the energy function to minimize is defined by:

$$P(X) = \sum_{e_{ij} \in plateau} \left| x_i - x_j \right|^q, q > 2 \tag{6.2}$$

This approach allowed us to extend the generalized model for image segmentation by adding, a new segmentation method for the cases $2 < q < \infty$ and $p \to \infty$. Moreover, one can see that this case deals with the finite values of q and $p = 0$.

6.4 Branch and Bound Algorithm to Achieve the Nash Equilibrium

The computation of a Nash equilibrium by classical methods is an NP-hard problem [10, 17, 19]. Then, to find the Nash equilibrium, we propose to use the Branch and Bound method. Indeed, the Branch and Bound is an effective technique used for solving NP-hard combinatorial optimization problems [18, 23]. A Branch and Bound algorithm for a minimization problem consists of three main components:

1. a branching rule,
2. a selection strategy,
3. a bounding function.

The branching rule

Allows determining if and how a subset of X can be split into subsets. Each subset is split into a finite number of proper subsets that represents sub-problems to solve. The bounding function provides for a given subspace a lower bound for the best solution value reachable in the subspace.

The selection strategy

Allows the selection of the active subspace to be explored. In the literature, we can find some applications [30] of the Branch and Bound method used to partition a graph into sub graph with minimum cost knowing the potential of every node in the graph [30]. In our case, we have to assign potentials to the nodes of the plateau corresponding to the Nash equilibrium of the proposed game. Thus, for the Branching part, we propose to split the strategy set X into two subsets, X_i' and X_i'' with $\left|X_i'\right| = \left|X_i''\right| \pm 1$. In general case (more than 2 objects to extract), the strategy set is split to M subsets, where M is the total number of objects to extract from the image and the background.

The bounding step

The problem is relaxed and the upper and the lower estimated bounds are checked. The proposed Branch and Bound algorithm is presented in Algorithm 6.1. It takes as input a set of strategies for every player i. Then, to reach the Nash equilibrium, this algorithm requires 7 iterations at most for every player. Consequently, if there is N players in our game, the algorithm needs between $N \times 7$ iterations at most to find the Nash equilibrium.

Algorithm 6.1: Power Game based on B&B

```
Input:
• A plateau $P(V, E)$, containing a set of foreground $F$, set of background $B$ seeds, and a set
  of unseeded nodes $x_i$.
• The set of strategies $X = X_1 \times X_2 \cdots \times X_N = 0, 0.01, 0.02, \ldots, 1^N$.
• $x_i = 1$ if the node $i \in F$, $x_i = 0$ if the node $i \in B$, and the potential $x_i$ is unknown if
  $i \notin (B \cup F)$.

Output: A potential $x_i$ for every node of the plateau $P$.
while $\exists X_i \subset X$ with $|X| > 1$ do
    1. Branching: Branch the subset $X_i$ into two subsets $X_i'$ and $X_i''$, where:
       $|X_i'| = |X_i''| \pm 1$ and $x_i' < x_i'' \; \forall x_i' \in X_i', \forall x_i'' \in X_i''$.
       $X' = X_i' \times X_j \; \forall i \neq j$
       $X'' = X_i'' \times X_j \; \forall i \neq j$
    2. Bounding:
       • Bound $X'$ looking for:
         $m_1 = \underset{x_i, x_j \in Inf(X'), Sup(X')}{Min} C_i(x)$
       • Bound $X''$ looking for:
         $m_2 = \underset{x_i, x_j \in Inf(X''), Sup(X'')}{Min} C_i(x)$
       • Search strategy
         if $m_1 < m_2$ then
             $X_i = X_i'$
         end
         else
             $X_i = X_i''$
         end
end
if $x_i > 0.5$ then
    $s_i = 1$
end
else
    $s_i = 0$
end
Result: print out the final segmentation.
```

6.5 Results and Discussions

6.5.1 Evaluation on Synthetic Images

To test and evaluate our approach, we compare our results with those obtained by PW $q = 2$, MSF Kruskal, MSF Prim. These methods belong to the same family of the general scheme of Couprie with $P \rightarrow \infty$. We have tested our method on different types of images. Thus, we have segmented some synthetic images, and microscopic images. The different tests show that the proposed method outperforms PW $q = 2$, MSF Kruskal and MSF Prim.

To evaluate objectively the performance of the tested methods, the Misclassification Error (ME) (Eq. 6.3) is used. The ME reflects the percentage of background pixels wrongly assigned to foreground and foreground pixels wrongly assigned to background [36]:

$$ME = \left(1 - \left(\frac{|B_{GT} \cap B_s| + |F_{GT} \cap F_s|}{|B_{GT}| + |F_{GT}|}\right)\right) \times 100 \qquad (6.3)$$

where B_{GT} is the background, F_{GT} the foreground of the Ground truth image, and B_s and F_s are the Background and the foreground of the segmented image, respectively.

Figure 6.1 shows the results on a synthetic image to which we add a Signal to Noise ratio SNR = 9.9533. Results provided by PG are better than those given by PW q = 2, MSF Kruskal and MSF Prim. Those results are confirmed by the ME measure in Tables 6.1, 6.2, and 6.3 from. Indeed, the values given by Game based PW method are the lowest ones.

We have also applied our method on microscopic images (Fig. 6.2). The extracted cells with PG are visibly better than those extracted with MSF Kruskal, MSF Prim and PW q = 2. The ME measures confirm our observations. One can remark also that the difference between segmentation results increase and is more noticeable if the object to extract has small size compared with the size of the image.

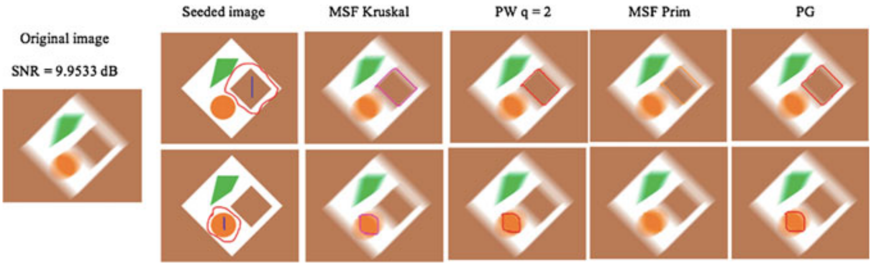


Fig. 6.1 Illustration of the segmentation results on noised synthetic image

Table 6.1 Optimal values x_i's at the corresponding energy, x_s, at the Nash equilibrium with a classical method for different values of q

q	3	4	5	10	25	50	100
x_1	0.69	0.68	0.68	0.67	0.67	0.67	0.67
x_2	0.71	0.69	0.69	0.68	0.67	0.67	0.67
x_3	0.34	0.34	0.34	0.33	0.33	0.33	0.33
x_4	0.41	0.38	0.37	0.35	0.34	0.34	0.34
Value	0.284023	0.09512002	0.03181364	0.0013201	9.38e−12	1.25e−23	3.23e−47
Iteration	101^4	101^4	101^4	101^4	101^4	101^4	101^4

Table 6.2 Optimal values x_i's at the corresponding energy at the Nash equilibrium given by a classical descent algorithm x_{NE} for different values of q

q	3	4	5	10	25	50	100
x_1	0.69	0.68	0.68	0.67	0.67	0.67	0.67
x_2	0.71	0.69	0.69	0.68	0.67	0.67	0.67
x_3	0.34	0.34	0.34	0.33	0.33	0.33	0.33
x_4	0.41	0.38	0.37	0.35	0.34	0.34	0.34
Value	0.284023	0.09512002	0.03181364	0.0013201	9.38e−12	1.25e−23	3.23e−47
Iteration	14965	14454	14352	13943	13739	13739	13739

Table 6.3 Optimal values x_i's at the corresponding energy at the Nash equilibrium with the Branch and Bound algorithm, $x_{NE-B\&B}$ for different values of q

q	3	4	5	10	25	50	100
x_1	0.689	0.689	0.689	0.689	0.689	0.689	0.657
x_2	0.686	0.686	0.686	0.679	0.674	0.672	0.665
x_3	0.686	0.686	0.686	0.679	0.674	0.672	0.665
x_4	0.375	0.375	0.373	0.356	0.345	0.344	0.329
Value	0.286621	0.0946326	0.03196730	0.0013787	1.21e−11	3.15e−23	1.36e−46
Iteration	14965	14454	14352	13943	13739	13739	13739

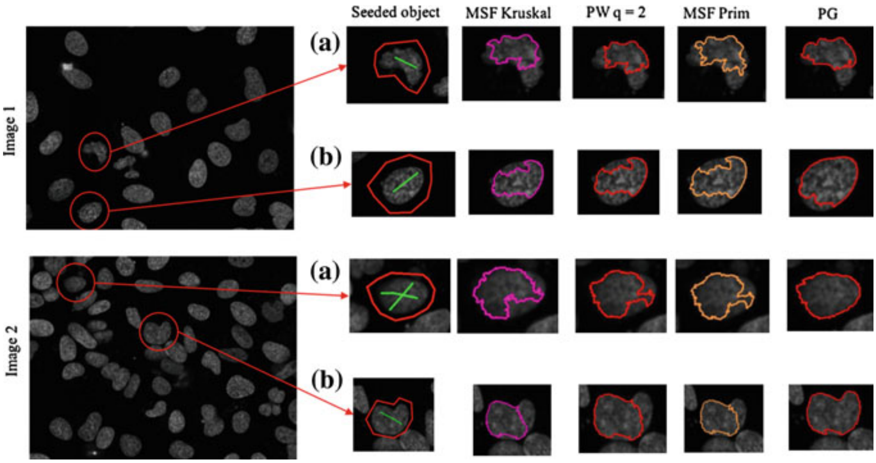


Fig. 6.2 Illustration of the segmentation results on two microscopic images

In order to illustrate the performance of our algorithm, different kind of microscopic images were considered from the cell library database [6]. For instance, the image in Fig. 6.3 is well segmented and its area can be easily calculated.

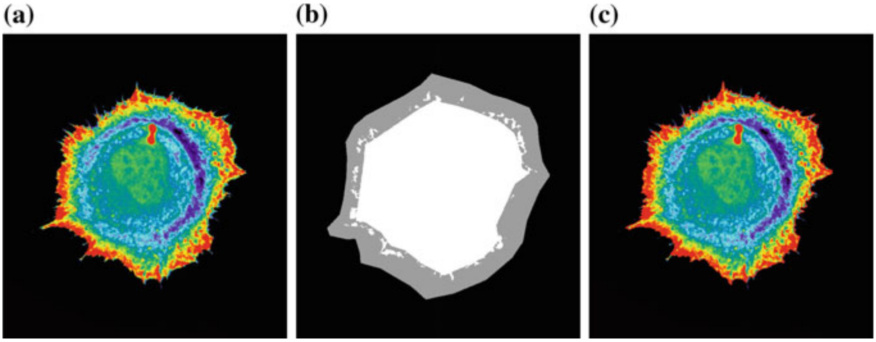


Fig. 6.3 Illustration of the segmentation results on microscopic images. **a** original image [6], **b** segmentation seed, **c** superimposition of the segmentation result and the original image

6.6 Conclusion

In this paper, we extended the recent work of Power Watershed $q = 2$, and $p \to \infty$. We studied the case of $2 < q < \infty$ and $p \to \infty$. The optimization problem is not linear for $2 < q < \infty$ and $p \to \infty$. The energy minimization was formulated as a non-cooperative strategic game problem and solved using B&B. Indeed, the resolution of a non linear optimization problem is equivalent to the computation of Nash equilibrium in non-cooperative strategic game. In addition, the proposed game is proven to be an exact potential game which guarantees the convergence to unique Nash equilibrium. As, the computation of Nash equilibrium is an NP-hard problem, the Branch and Bound method allows to solve it in a reasonable time. In work under progress, changing the final decision s_i and make it adaptive, in order to enhance the segmentation results. A second direction would be to study the case of q finite and p finite; this case can open a way to other segmentation results method.

References

1. C.V. Alvino, G.B. Unal, G.G. Slabaugh, B. Peny, T. Fang, Efficient segmentation based on Eikonal and diffusion equations. Int. J. Comput. Math. **84**(9), 1309–1324 (2007)
2. X. Bai, G. Sapiro, A geodesic framework for fast interactive image and video segmentation and matting, in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Rio de Janiero, 14–21 October 2007, pp. 18
3. G. Bertrand, On topological watersheds. J. Math. Imaging Vis. **22**(2–3), 217–230 (2005)
4. Y. Boykov, M.P. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Vancouver, BC, USA, vol. 1, 7–14 July 2001, pp. 105–112
5. O. Carbonell-Nicolau, R.P. McLean, Refinements of Nash equilibrium in potential games. Theor. Econ. **9**, 555–582 (2014)
6. Cell database: http://www.cellimagelibrary.org/

7. A. Chakraborty, J.S. Duncan, Game-theoretic integration for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **21**(1), 12–30 (1999)
8. B. Chatterjee, An optimization formulation to compute Nash equilibrium in finite games, in *International Conference on Methods and Models in Computer (ICM2CS 2009)*, Delhi, India, 14–15 December 2009, pp. 1–5
9. R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, S.W. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. Proc. Natl. Acad. Sci. **102**(21), 7426–7431 (2005)
10. V. Conitez, T. Sandholm, New complexity results about Nash equilibria. Games Econ. Behavior **63**, 621–641 (2008)
11. C. Couprie, L.J. Grady, L. Najman, H. Talbot, Power watershed: a unifying graph-based optimization framework. IEEE Trans. Pattern Anal. Mach. Intell. **33**(7), 1384–1399 (2011)
12. J. Cousty, G. Bertrand, L. Najman, M. Couprie, Watershed cuts: minimum spanning forests and the drop of water principle. IEEE Trans. Pattern Anal. Mach. Intell. **31**(8), 1362–1374 (2009)
13. J. Cousty, G. Bertrand, L. Najman, M. Couprie, Watershed cuts: thinnings, shortest path forests, and topological watersheds. IEEE Trans. Pattern Anal. Mach. Intell. **32**(5), 925–939 (2010)
14. A. Criminisi, T. Sharp, A. Blake, Geos: geodesic image segmentation, in *Proceedings of the 10th European Conference on Computer Vision ECCV 08: Part I*, Marseille, France, 12–18 October 2008, pp. 99–112
15. O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, F. Sgonne, Segmentation by transduction, in *Proceedings of IEEE Conference in Computer Vision and Pattern Recognition (CVPR)* (2008)
16. A.X. Falcao, J. Stolfi, R. de Alencar Lotufo, The image foresting transform: theory, algorithms, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **26**(1), 1929 (2004)
17. D. Fotaskis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, P. Spirakis, The structure and complexity of Nash equilibria for a selfish routing game. Theor. Comput. Sci. **410**, 3305–3326 (2009)
18. B. Gendron, T.G. Crainic, Parallel Branch and Bound algorithms: survey and synthesis. Oper. Res. **42**, 1042–1066 (1994)
19. I. Gilboa, E. Zemmel, Nash and correlated equilibria: some complexity consideration. Games Econ. Behavior **1**, 80–93 (1989)
20. L.J. Grady, Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **28**(11), 1768–1783 (2006)
21. L.J. Grady, J.R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science* (Springer, New York, 2010)
22. M. Kallel, R. Aboulaich, A. Habbal, M. Moakher, A Nash-game approach to joint image restoration and segmentation. Appl. Math. Model. **38**, 3038–3053 (2014)
23. E.L. Lawler, D.W. Wood, Branch-and-Bound methods: a survey. Oper. Res. **14**, 699–719 (1966)
24. J. Li, G. Zeng, R. Gan, H. Zha, L. Wang, A game-theoretical approach to image segmentation, *Computational Visual Media*, vol. 7633, Lecture Notes in Computer Science (Springer, Berlin, 2012), pp. 33–42
25. R.D. McKelvey, A Liapunov function for Nash equilibria, Technical Report, California Institute of Technology (1991)
26. J.D. Molina, H. Rudnick, Transmission expansion investments: cooperative or non cooperative game?, in *General Meeting of Power and Energy Society 2010*, Minneapolis, MN, USA, 25–29 July 2010, pp. 1–7
27. D. Monderer, L.S. Shapley, Potential games. Games Econ. Behavior **14**, 124–143 (1996)
28. L. Najman, M. Schmitt, Watershed of a continuous function. Signal Process. **38**(1), 99–112 (1994)
29. J.F. Nash, Noncooperative games. Ann. Math. **54**, 289–295 (1951)
30. J. Nossak, E. Pesch, A branch-and-bound algorithm for acyclic partitioning problem. Comput. Oper. Res. **44**, 174–184 (2014)
31. M.J. Osborne, A. Rubinstein, *A Course in Game Theory* (The MIT Press, Cambridge, 1994)
32. B. Peng, L. Zhang, D. Zhang, A survey of graph theoretical approaches to image segmentation. Pattern Recognit. **46**, 1020–1038 (2013)

33. R.C. Prim, Shortest connection networks and some generalizations. Bell Syst. Tech. J. **36**(6), 1389–1401 (1957)
34. J.B.T.M. Roerdink, A. Meijster, The watershed transform: definitions, algorithms and parallelization strategies. Fundam. Inform. **41**(1–2), 187–228 (2000)
35. G. Scutari, S. Barbarossa, P. Palomar, Potential games: a framework for vector power control problems with coupled constraints, in *IEEE International conference of Acoustics, Speech and Signal Processing*, Vol. 4 (2006), pp. 241–244
36. M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance. J. Electron. Imaging **13**(1), 146–165 (2004)
37. A. Shahid, S. Aslam, H. Soek Kim, K.G. Lee, Distributed joint source and power allocation in self-organized femtocell networks: a potential game approach. J. Netw. Comput. Appl. **46**, 280–292 (2014)
38. D. Shen, G. Chen, Y. Zheng, E. Blasch, K. Pham, Game theoretic approach to similarity based image segmentation. Signal Data Process. Small Targets **8137**, 1–12 (2011)
39. J. Shi, J. Malik, Normalized cuts and image segmentation. IEEE Trans. PAMI **22**(8), 888–905 (2000)
40. A. K. Sinop, L. Grady, A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm, in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Rio de Janiero, 14–21 October 2007, pp. 18
41. L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. **13**(6), 583–598 (1991)
42. M. Voorneveld, Best response potential games. Econ. Lett. **66**(3), 289–295 (2000)

Chapter 7
Dynamic Metaheuristics for Brain Cine-MRI

Amir Nakib

7.1 Introduction

Recently, a new technique for obtaining brain images of cine-MR (Magnetic Resonance) type has been developed by Hodel et al. [8]. The principle of this technique is to synchronize the MRI signal with the ECG (Electrocardiographic) signal. The MRI signal provides three dimensional images and cuts of high anatomical precision, and the ECG signal is obtained from the heart activity. An image of brain cine-MRI is therefore built by making the average of the MRI signals acquired during the R-R period of the ECG signal. This technique allows to have a good visualization of the movements of the walls of the third ventricle during the cardiac cycle. For more details about this method see. In this work, we were interested in the automation of the assessment of the movements of the walls, which allows a better understanding of physiological brain functioning and the provision of aid to diagnosis and therapeutic decision. Here, we are not be interested in the cerebrospinal fluid (CSF), the reader can have more details about analysis of the motion using CSF in [9, 10].

Several methods for the movement quantification have been proposed in the literature for myocardium images. This image processing application requires the partitioning of the image into homogeneous regions. Then, the fundamental process used is called image segmentation, that plays an important role in entire image analysis system [14]. It is generally performed before the analysis and the decision-making process in many image analyses, such as the quantification of tissue volumes, diagnosis, and localization of disease, the study of anatomical structures, the matching and motion tracking. Since the segmentation is considered by many authors to be an essential component of any image analysis system, this problem has received a great deal of attention; thus any attempt to completely survey the literature would be too space consuming. However, surveys of most segmentation methods may be

A. Nakib (✉)
Laboratoire LISSI, Université Paris Est, 122 Rue P. Armangot, 94400 Vitry-sur-Seine, France
e-mail: nakib@u-pec.fr

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_7

found in [7, 19]; another review of thresholding methods is in [21]. In [4, 7, 22] the authors proposed level set based methods to assess cardiac movements. These methods cannot be applied directly to our brain sequences. Indeed, firstly, the amplitude of movements in the heart is much greater than that of the walls of the third ventricle. Secondly, due to the presence of cerebrospinal fluid, one cannot properly segment the entire ventricle and quantify its movements in the sequence, because the stopping criteria are adapted to our images and the algorithmic complexity is high.

The goal in this work was to develop a low computation complexity method to assess and quantify these movements to be used as a clinical routine. Then, we proposed a framework that consists of two phases:

1. the extraction of contour of the walls of the 3rd ventricle is performed.
2. the contours' registration is done to achieve the deformation model.

In the segmentation step (phase 1), images are segmented by a fractional differentiation based method in [16, 17]. It is followed by a technique aimed at optimizing the differentiation order. This phase allows to enhance the contrast and to extract the contours of a selected region of interest (ROI) at different times of the cardiac cycle. In the second step, the information provided by these contours is combined through their mapping. This registration procedure allows us to track the movements of each point of the contour of the ROI over time, and to obtain a better mathematical modeling of the movement. The parameters of this model are calculated over all the sequence. As we perform optimization several times for every couple of images, then the landscape of the function to be optimized (objective function) changes. Thus, we talk about dynamic optimization.

7.2 Problem Formulation

The goal behind this work is to build an atlas of the movements of the healthy ventricles in the context of the hydrocephalus pathology. Then, in this paper, our objective is to segment the walls of the third ventricle, and quantify their movements. We have tested several image segmentation methods based on edge detection approach: the method of Canny, derivative methods and more robust methods such as the Level-set approach [7]. All these methods give similar unconvincing results because they do not reproduce precisely the contours of the third ventricle on the images of the used sequences. This is due to the fact that they are very noisy, due to the presence of CSF, and variation of their contrasts on the same subject (third ventricle).

For our application, the segmentation of the entire walls is not possible (differences between the pathologic and sane cases) and is not necessary. Consequently, we decided to work only on a region of interest (ROI). In Fig. 7.1a, we present an example of an ROI: Lamina Terminalis (LT). It is the first image of a brain cine-MRI sequence in a pathologic case (hydrocephalus). The sane case is presented in Fig. 7.1b.

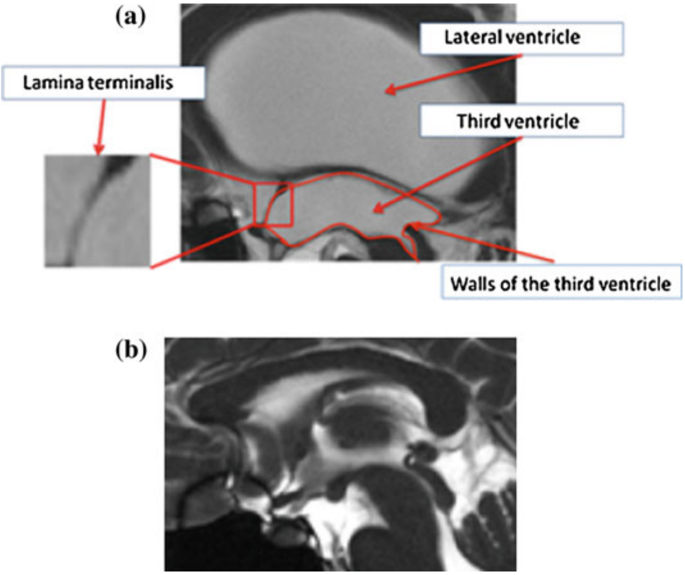


Fig. 7.1 Illustration of the first image of the original sequence. Pathologic case (Hydrocephalus), **b** sane case

Hydrocephalus is usually due to the blockage of CSF outflow in the ventricles. Patients with hydrocephalus have abnormal accumulation of CSF in the ventricles. This may be source of the increase of the intra-cranial pressure inside the skull and progressive enlargement of the head, convulsion, and mental disability.

The goal of developing this framework is to contribute to the study of the diagnosis of endoscopic third ventriculostomy (ETV) patency. To validate the proposed method, sixteen age-matched healthy volunteers were explored with the same MR protocol (12 women, 4 men; average age 38.5 years, interquartile range: 25.554). This study was approved by the local ethics committee; written informed consent was obtained for all the patients.

7.3 Data Acquisition

Data were obtained using 1.5-T MR (Siemens, Avanto, Erlangen, Germany). The protocol included a retrospectively gated cine true/Fast Imaging with Steady-state Precession Magnetic Resonance (cine True FISP MR) sequence: mid-sagittal plane was defined on a transverse slice from the center of the Lamina Terminalis to the cerebral aqueduct [8]. This acquisition technique provides only a sequence of 2D image, then a 3D image segmentation technique cannot be applied.

7.4 Proposed Framework

In this section we present the proposed framework that consists in two phases: the first phase is the extraction of contours of the image ROI with a segmentation technique, based on two dimensional digital-fractional integration (2D-DFI). In the second phase, we combine the information provided by the contours of the image sequence through a registration. The steps of our framework are illustrated in Fig. 7.2.

7.4.1 Segmentation Problem

7.4.1.1 Segmentation Based on Fractional Differentiation

In order to segment every image of the sequence, a segmentation method based on two dimensional digital fractional integration (2D-DFI), where the fractional order is negative (in the interval $(-1, 0]$). The fractional integration (also called fractional differentiation (FD) with negative order) is based on the works of Leibniz and Hospital in 1965. The applications of this method are numerous, it is used in automatics [2, 12], in signal processing [15] and in image processing [13, 16, 17]. The FD of Riemann–Liouville is defined as follows:

$$D^{-\alpha} f(x) = \frac{1}{\Gamma(\alpha)} \int_c^x (x-\xi)^{\alpha-1} f(\xi)\, \mathrm{d}x. \tag{7.1}$$

where $f(x)$ is a real and causal function, $x > 0$, α the order of the FD, c the interval of the integral and the function of Euler–Lagrange. In the case of integration, the order α is negative and in the discrete domain, the approximation of the DFI is given

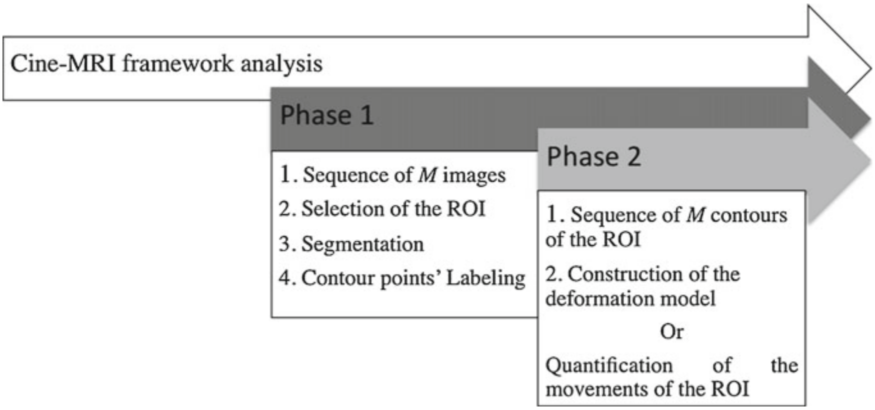

Cine-MRI framework analysis
Phase 1
1. Sequence of M images
2. Selection of the ROI
3. Segmentation
4. Contour points' Labeling
Phase 2
1. Sequence of M contours of the ROI
2. Construction of the deformation model
Or
Quantification of the movements of the ROI

Fig. 7.2 Overall scheme of the cine-MRI framework analysis

```
Optimal DFD order
Inputs: $\alpha = -0.9$
for $i = -0.9$ to $-0.1$ do
    Apply 2D-DFI at the order $\alpha$ to the original Image $(I)$
    Compute t that corresponds to the order $\alpha$
    Calculate the segmentation quality measure criterion corresponding to $t$
end
Find $t_{opt}$ optimizes the segmentation quality measure criterion.
Print results: optimal order $\alpha_{opt}$, optimal threshold $t_{opt}$ and segmented image.
```

Fig. 7.3 Search for the optimal α_{opt}

by:

$$g^{\alpha}(x) = D^{-\alpha} f(x) \approx \frac{1}{h^{\alpha}} \sum_{k=0}^{M} ((\beta_k)(\alpha) f(x - kh)) \tag{7.2}$$

where h is the sampling step, M the number of samples, $x = M \times h$ and $\beta_k(\alpha)$ are defined by:

$$\beta_0(n) = 1, \beta_{k+1}(\alpha) = \frac{(k+1) - \alpha - 1}{(k+1)} \beta_k(\alpha), k = 0, 1, 2, M - 1 \tag{7.3}$$

The Eq. (9.3) is equivalent to the Riemann–Liouville equation when h tends to zero. The 2D-DFI is given, for a real and bounded function $f(x, y)$ by:

$$D^{-\alpha} f(x, y) = \left(\frac{\partial}{\partial x}\right)^{\alpha} \left(\frac{\partial}{\partial y}\right)^{\alpha} f(x, y) \approx \frac{1}{h^{2\alpha}} \sum_{k=\lfloor -M/2 \rfloor}^{\lfloor -M/2 \rfloor} \sum_{l=\lfloor -N/2 \rfloor}^{\lfloor -N/2 \rfloor} (p(k, l) f(x - hk, y - lh)) \tag{7.4}$$

M and N are the number of elements of f taken into account for calculating the differential image, $M \times N$ represents the size of the mask, $p(k, l) = \beta_k(\alpha) \times \beta_l(\alpha)$ are the elements of the matrix, $p^{\alpha}_{M,N}(p(k, l))$ calculated from the Eq. (7.3), which corresponds to the horizontal and vertical components, respectively. $\lfloor x \rfloor$ denotes the integer part of x.

The optimal segmentation of an image corresponds to finding the optimal 2D-DFI order. In order to find the best value, a criterion that characterizes the best segmentation was used, then the best value is that providing a segmentation that optimizes the defined criterion (Fig. 7.3). Indeed, a good result means that all regions (connected components) are homogeneous and their number will not be under 2 and greater than 3. Less than 2 (equal to 1), it means that there is only one region; in this case, the segmentation result is bad. Greater than 3, it means that there are too many regions and the image is over segmented, thus the segmentation result is bad.

In order to obtain the optimal order for the segmentation process, the well known uniformity criterion is used [5].

7.4.1.2 Segmentation Based on Dynamic Optimization

Before using this criterion we must fit the histogram of the image to be segmented to a sum of Gaussian probability density functions (pdf's). This procedure is named Gaussian curve fitting, more details about it are given below. The pdf model must be fitted to the image histogram, typically by using the maximum likelihood or mean-squared error approach, in order to find the optimal threshold(s). For the multimodal histogram $h(i)$ of an image, where i is the gray level, we fit $h(i)$ to a sum of d probability density functions. The case where the Gaussian pdf's are used is defined by:

$$p\left(x\right) = \sum_{i=1}^{d} P_i \exp\left[-\frac{(x-\mu_i)^2}{\sigma_i^2}\right] \tag{7.5}$$

where P_i is the amplitude of Gaussian pdf on μ_i, μ_i is the mean and σ_i^2 is the variance of mode i, and d is the number of Gaussian used to approximate the original histogram and corresponds to the number of segmentation classes. Given an image histogram $h(j)$ (observed probability of gray level j), it can be defined as follows:

$$h(j) = \frac{g\left(j\right)}{\sum_{i=0}^{L-1} g\left(i\right)} \tag{7.6}$$

where $g(j)$ denotes the occurrence of gray-level j over a given image ranges $[0, L-1]$. Our goal is to find a vector of parameters, Θ, that minimizes the fitting error J, given by the following expression:

$$\text{Minimize } J = \sum_i |h(i) - p(\Theta, x_i)|^2 \tag{7.7}$$

where i ranges over the bins in the measured histogram. Here, J is the objective function to be minimized with respect to Θ, a set of parameters defining the Gaussian pdf's and the probabilities, given by $\Theta = \{P_i, \quad \mu_i, \quad \sigma_i; \quad i = 1, \ 2, \ldots, \ d\}$. After fitting the multimodal histogram, the optimal threshold could be determined by minimizing the overall probability of error, for two adjacent Gaussian pdf's, given by:

$$e(T_i) = P_i \int_{-\infty}^{T_i} p_i\left(x\right) dx + P_{i+1} \int_{T_i+1}^{\infty} p_{i+1}\left(x\right) dx \tag{7.8}$$

with respect to the threshold T_i, where $p_i(x)$ is the i^{th} pdf and $i = 1, \ldots, d-1$. Then the overall probability to minimize is:

$$E(T) = \sum_{i=1}^{d-1} e(T_i) \tag{7.9}$$

where T is the vector of thresholds: $0 < T_1 < T_2 < \cdots < T_{(d-1)} < L - 1$. In our case L is equal to 256.

The criterion in (7.9) has to be minimized for each image, as we are in the case of a sequence, then the fitting criterion becomes:

$$\text{Minimize } J(t) = \sum_{i} |h(i,t) - p(\Theta(t), x_i)|^2 \tag{7.10}$$

where t is the number of the current image in the cine MRI sequence, $\Theta(t)$ is same as Θ defined before but here is dependent on the image. $h(i,t)$ is the histogram of the image number t.

7.4.2 Geometric Matching of the Contours

The obtained contours in the segmentation phase will be used to assess the movement of the contours of the ROI over time. This step requires matching these contours. However, several false matches appear. To eliminate this problem, we have separated the obtained contours after segmentation and indexing and we have only kept the contour corresponding to the third ventricle (in the case of the Lamina Terminalis, this is in the right side). To evaluate the ventricle deformation, a contour matching operation is required after the segmentation phase in order to track the position of points belonging to the contours of the region of interest (ROI) over time. This operation is carried out in two steps: first, a rigid registration, called alignment, takes into account the displacement of the global membrane. Then, a morphing process performs accurate elastic matching of the successively aligned contours.

7.4.2.1 Contours' Alignment

This step consists in looking, for each point in the curve of a reference image, at the nearest point in the curve in the destination image, based on a predefined minimum distance. The different steps of that phase matching are summarized as follows: For each point of the source curve:

1. Calculate the distance between this point and all points of the destination curve.
2. Match this point with the nearest point of the destination curve.

In this step, the goal is to associate each point of the initial contour to one point of the second contour. We assume that each point of the first contour can be associated with at least one point of the second contour. The alignment procedure realizes a registration between these two curves through a geometrical transformation controlled by a dissimilarity criterion based on 3 parameters. A non linear model can also be considered as a deformation model.

In the matching phase, a point is mapped to the nearest point, but the direct application of the matching method to the ROI contour without indexing produces false matches. Other false matches are due to the presence of several equidistant points from the point to match. Zhang [23] proposed a method using an adaptive statistical threshold, which is based on the statistical characteristics of the distances between matched points, such as the average and the standard deviation. We used this method to eliminate false matches and to assess the quality of the registration obtained. False matches of this type appear only if an alignment takes the first curve as a reference, i.e. by matching the contour of each sequence with the first points of the contour of the first image.

7.4.2.2 The Deformation Model

In order to approximate the deformation model, we use a registration technique. The idea of the registration is to combine information from two (or more) images of the same scene, but obtained at different moments, with different views or different acquisition methods. Then, the aim of a registration system is to determine the mapping information (positions, gray-scale, structures, etc.) representing a physical reality on these different images.

The goal of image registration is to find the best transformation T' among all the transformations T applied to an image I that looks at best like the image J. It is quite difficult to review all the image registration methods that have been proposed in the literature, a survey of many methods is presented in [24]. According to the used primitives (attributes), in our work we considered the geometric approach of image registration.

In this approach an extract of the geometric primitives (points, curves or surfaces) is performed, and then a geometric similarity criterion is used to find the best correspondence. In the case where the primitives are points, the Euclidean distance is usually the most used [18]. In the case of surfaces and curves, the most widely used algorithm is the Iterative Closest Point (ICP) algorithm [3, 6, 11, 23]. Other criteria are based on geometric calculations cards, Chamfer distances or Hausdorff distance [1, 18]. Geometric approaches have the advantage of holding high-level information, but remain vague regarding the extraction of primitives. The optimization is a very important step in image registration. It aims at determining an optimum processing according to a similarity criterion. The optimization process is iterative. It must find at each stage the parameters T of the processing that ensure the best match between the two images until the convergence to the optimal solution. Broadly speaking, the optimization problem is formulated by:

$$T_{opt} = arg_{\tau \epsilon \Omega} max \zeta(J, T(I)) \tag{7.11}$$

where: I is the original image, J the image to register, Ω:search space for possible transformations, ζ: similarity criterion chosen, T_{opt}: optimum processing.

Among the optimization methods, we find numerical methods (without use of the gradient), such as Simplex and Powell [1] methods, and gradient based methods, such as gradient descent, conjugate gradient, the method of Newton and Levenberg-Marquardt [1, 18]. In our works, we proposed to use dynamic optimization metaheuristics to solve this optimization problem. Recently, static evolutionary algorithms were also used to find the optimal registration model parameters' [20].

In order to assess the deformation model, the contours of the indexed images of the sequence were matched by taking at each time two contours of two successive images. The similarity criterion that measures the distance between two successive contours must be minimized.

Considering the distortion models that exist, we assume, for instance, that the movements of the third ventricle are governed by an affine transformation. This model is characterized by a rotation θ, two translations (t_x, t_y) and two scaling factors (s_1, s_2) according to x and y:

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} S_1 \cdot \cos\theta & -S_2 \cdot sin\theta & t_x \\ S_1 \cdot \sin\theta & S_2 \cdot \cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \tag{7.12}$$

$P_1(x_1, y_1)$ is a point of the reference primitive and $P_2(x_2,y_2)$ is the point obtained with the geometric model.

The parameters of this transformation are defined by minimizing the squared error among all points of the curve and those obtained with the model given by this function:

$$SE(X) = \sum_{i=1}^{N_1} ([C_2^{'}(i) - T^x(C_1(i))]^2) \tag{7.13}$$

N_1 is the cardinal of all the points of the contour C_1 and $x \equiv (s_1, s_2, \theta, t_x, t_y)$ is the vector of parameters. Several authors such as [23] use an iterative local search algorithm to solve this problem. To avoid local minima, the optimization criteria are modified by adding weight terms inversely proportional to the distances between matched points. This makes the optimization algorithm more complex.

7.4.3 Cine-MRI Registration as a DOP

The registration of a cine-MRI sequence can be seen as a dynamic optimization problem. Then, the dynamic objective function optimized by a dynamic optimization algorithm changes according to the following rules:

- The criterion in (9.6) has to be minimized for each couple of contours, as we are in the case of a sequence, then the optimization criterion becomes:

$$MSE(\Phi(t), t) = \sum_{j=1}^{L_t} \frac{\left[C_t'(j) - T_{\Phi(t)}(C_t(j))\right]^{\mathrm{T}} \left[C_t'(j) - T_{\Phi(t)}(C_t(j))\right]}{L_t} \tag{7.14}$$

where t is the index of the contours on which the transformation $T_{\Phi(t)}$ is applied, also equal to the index of the current couple of contours in the sequence. $\Phi(t)$, $C_t'(j)$, $C_t(j)$ and L_t are the same as Φ, C_1', C_1 and L_1 defined before, respectively, but here are dependent on the couple of contours.

- Then, the dynamic optimization problem is defined by:

$$\text{Min } MSE(\Phi(t), t) \tag{7.15}$$

- If the current best solution (transformation) found for the couple t cannot be improved anymore (according to a stagnation criterion), the next couple $(t + 1)$ is treated.
- The stagnation criterion of the registration of a couple of successive contours is satisfied if no significant improvement in the current best solution is observed.
- Thus, the end of the registration of a couple of contours and the beginning of the registration of the next one constitute a change in the objective function.

This formulation (introduction of the time variable t to get an objective function that changes over the time) allows the use of dynamic optimization algorithms to solve this problem, rather than having to restart a static algorithm to register a sequence of images. Then, information acquired on the objective function during the registration of several couples of contours, in a sequence, can be used by the dynamic optimization algorithm to accelerate the registration of the next couples (the correlations between the images of the sequence can be taken into account).

7.5 Results and Discussions

In this section, we illustrate the results of the extraction of the ROIs contours (for the Lamina Terminalis), followed by the results for the registration and the quantification of the movement.

Finally, we will present the results obtained for the quantification of other ROIs of the sequences. The results of the proposed quantification method have been clinically validated by an expert and compared to those in [8] obtained using a manual segmentation.

The first phase of the framework consists in the extraction of the ROI. It is carried out in two steps: the segmentation and the indexation. We proposed two algorithms to segment the different images of the sequence.

Then, only perimeters of the segmentation results are considered. In order to illustrate this phase, we consider the original sequence in Fig. 7.4. The obtained segmentation results are presented in Fig. 7.5.

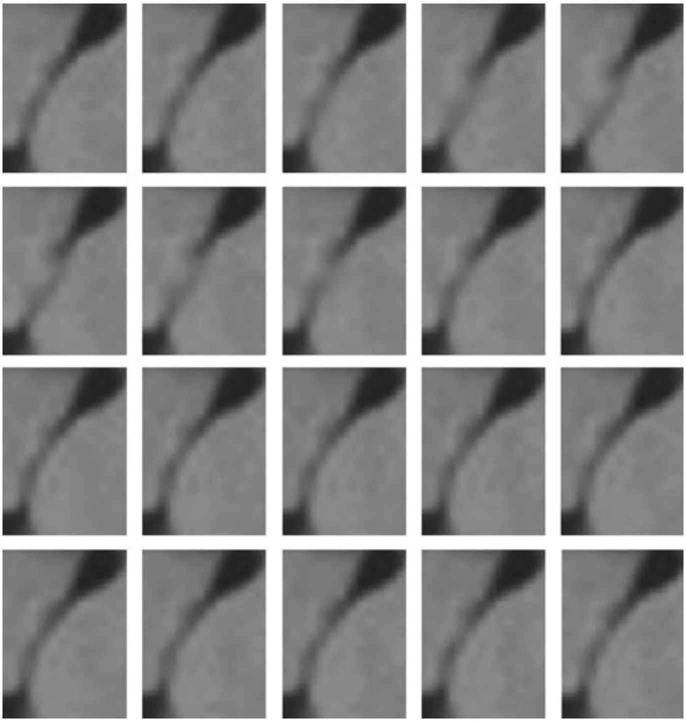


Fig. 7.4 Original sequence

From the segmentation results and after the matching process, the representation of movement of the ROI can be shown. In Fig. 7.7, we present the case of the example in Fig. 7.6.

The registration process allows to have all parameters of the deformation for all the different couples of images (i.e. deformation of the contour in image1 towards that of image 2, etc.).

To quantify the movement of the ROI, we have aligned all the contours of the sequence with the contour of the first image. Figure 7.7 shows the amplitudes of displacements of each point over time in the case of the Lamina Terminalis (Fig. 7.4).

The movements that we are interested in this work are those that have maximum amplitude. In the sequence used, the maximum movement is of 2.57 mm. This has been clinically validated by an expert in this study and is in the same range as the results published in [8]. The horizontal and vertical displacements are given in Table 7.1.

Further tests were made on other image sequences of patients and sequences of witnesses with no abnormalities in the third ventricle. We found that our method depends on the spatial resolution of images used. When the resolution is low, the quantification result is better. After several tests, we noticed that, for a good quantification

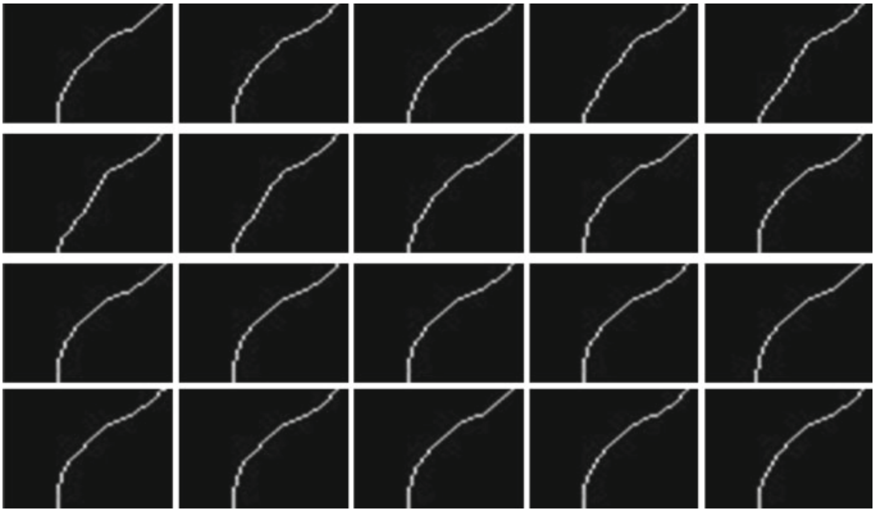


Fig. 7.5 Illustration of the segmentation result for the sequence in Fig. 7.4

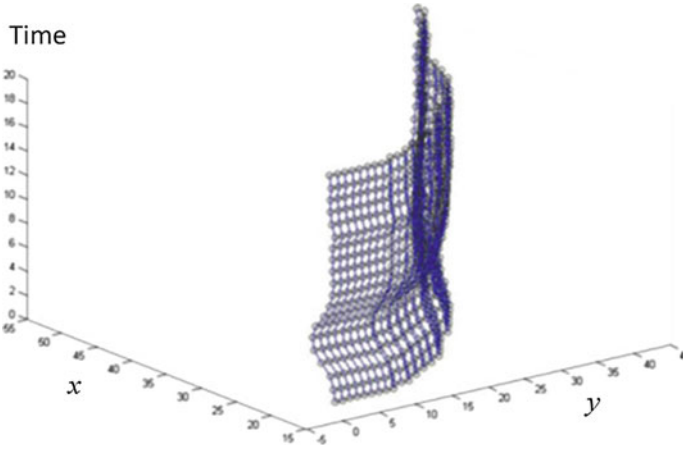


Fig. 7.6 Overall scheme of the cine-MRI framework analysis

result of the movement of the walls of the third ventricle, it is necessary that the spatial resolution of a pixel is smaller than or equal to 0.6 mm. More details about these contributions can found in the publications listed in the Sect. ??.

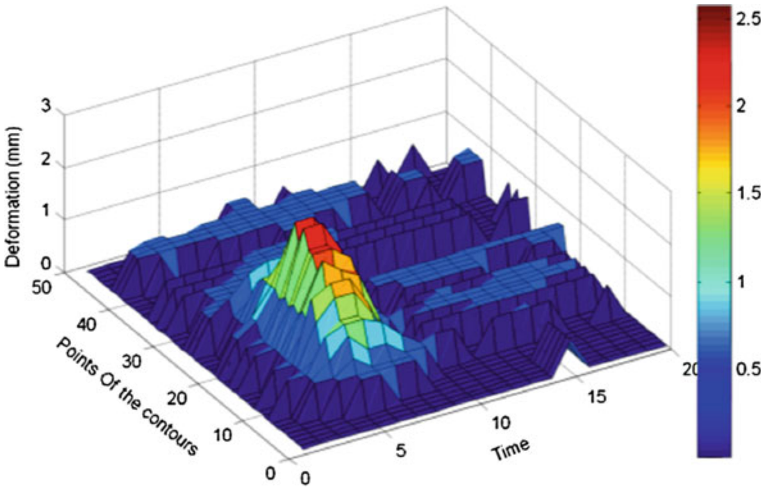


Fig. 7.7 Overall scheme of the cine-MRI framework analysis

Table 7.1 Obtained set of parameters of the deformation model

Images	s	θ	t_x	t_y
imag01-02	0.9917	0.0023	0.615	1.4813
imag02-03	0.9999	0.0015	0.2151	−0.639
imag03-04	1.0044	0.0042	0.2884	0.4406
imag04-05	0.9985	0.0025	−0.2611	0.4406
imag05-06	0.9977	0.0028	0.0736	0.1055
imag06-07	1.0018	0.0117	0.5290	0.1088
imag07-08	0.9984	0.0076	0.3330	−0.1900
imag08-09	0.9985	−0.0016	0.1376	−0.2963
imag09-10	0.9967	−0.0011	0.1136	−0.1026
imag10-11	0.9959	−0.0015	0.2835	−0.3654
imag11-12	1.0018	−0.0004	−0.2613	−0.5670
imag12-13	1.0004	−0.0010	0.1992	−0.4103
imag13-14	1.0087	0.0085	0.1123	−0.6622
imag14-15	0.9843	0.0070	0.4257	0.7800
imag15-16	1.0033	−0.0033	−0.5332	−0.3712
imag16-17	1.0086	−0.0086	0.371	0.1734
imag17-18	1.0047	0.0047	0.1351	−0.1494
imag18-19	1.0028	0.0028	−0.1303	0.9025
imag19-20	1.0073	−0.0073	0.2596	0.9034

7.6 Conclusion

In this work, we were interested in quantifying the movements of the third cerebral ventricle in a cine-MRI sequence. To solve the problem of quantifying the movements of the third cerebral ventricle, we proposed a method for quantification of movement based on fractional integration and dynamic metaheuristics. A new segmentation methods based on the fractional integration and a new formulation of the segmentation problem as a dynamic optimization problem were proposed to segment "*quickly*" all sequences. In this step, the thresholding method provided good results for detecting the contours of the images. For the registration step, a covariance matrix adaptation evolution strategy, called D-CMAES and MLSDO were proposed to allow, first, to build a distortion model representing the distortion of the ROI and, secondly, to quantify its movements, without restarting the optimization process from the beginning. The obtained results were considered good and represent the movement of the ROI rather well. In order to take into account the third dimension, the design of a new acquisition technique is under progress to enhance the quality the cine-MRI.

References

1. X. Baty, C. Cavaro-Menard, J.-J. Le Jeune, Automatic multimodal registration of gated cardiac PET, CT and MR sequences, in *6th IFAC Symposium on Modelling and Control in Biomedical Systems, Reims (France)* (2006)
2. A. Benchellal, T. Poinot, J.C. Trigeassou, Approximation and identification of diffusive interfaces by fractional models. Signal Process. **86**(10), 2712–2727 (2006)
3. P.J. Besle, N.D. McKay, A method for registration of 3d shape. IEEE Trans. Pattern Anal. Mach. Intell. **14** (2) (1992)
4. M.J. Budoff, N. Ahmadi, G. Sarraf, Y. Gao, D. Chow, F. Flores et al., Determination of left ventricular mass on cardiac computed tomographic angiography. Acad. Radiol. **16**(6), 726–732 (2009)
5. C.A. Cocosco, A.C. Evans, A fully automatic and robust brain MRI tissue classification. Med. Image Anal. **7**(4), 513–527 (2003)
6. B. Delhay, P. Clarysse, S. Bonnet, P. Grangeat, I.E. Magnin, Combined 3d object motion estimation in medical sequences, in *IEEE International Conference on Image Processing, Singapore* (2004)
7. L. He, A comparative study of deformable contour methods on medical image segmentation. Image Vis. Comput. **26**(2), 141–163 (2008)
8. J. Hodel, P. Decq, A. Rahmouni, S. Bastuji-Garin, C. Combes, Brain ventricular wall movement assessed by a gated cine MR true FISP sequence in patients treated with endoscopic third ventriculostomy. **19**(12) (2009)
9. V. Kurtcuoglu, M. Soellinger, P. Summers, D. Poulikakos, P. Boesiger, Mixing and modes of mass transfer in the third cerebral ventricle: a computational analysis. J. Biomech. Eng.**129**, 695–702 (2007)
10. V. Kurtcuoglu, V. Soellinger, M. Summers, P. Boomsma, K. Poulikakos, D. Boesiger et al., Computational investigation of subject-specific cerebrospinal fluid flow in the third ventricle and aqueduct of Sylvius. J. Biomech. **40**, 1235 (2007)
11. G. Malandin, E. Bardinet, K. Nelissen, W. Vanduffel, Fusion of autoradiographs with an MR volume using 2-d and 3-d linear transformations. NeuroImage **23**(1), 111 (2004)

12. D. Mansouri, D. Bettayeb, Optimal low order model identification of fractional dynamic systems. Appl. Math. Comput. **206**, 543 (2008)
13. B. Mathieu, P. Melchior, A. Oustaloup, C. Ceyral, Fractional differentiation for edge detection. Signal Process. **83**(11), 2421–2432 (2003)
14. A. Mayer, H. Greenspan, An adaptive mean-shift framework for MRI brain segmentation. IEEE Trans. Med. Imaging **28**(8), 1238–1250 (2009)
15. A. Nakib, Y. Ferdi, Entropy reduction in fractionally differentiated electrocardiogram signals, in *European Medical and Biological Engineering Conference EMBEC'02*, vol. 3 (2002)
16. A. Nakib, H. Oulhadj, P. Siarry, Fractional differentiation and non-Pareto multiobjective optimization for image thresholding. Eng. Appl. Artif. Intell. **22**(2), 236–249 (2009)
17. A. Nakib, H. Oulhadj, P. Siarry, A thresholding method based on two-dimensional fractional differentiation. Image Vis. Comput. **27**(9), 1343–1357 (2009)
18. V. Noblet, C. Heinrich, F. Heitz, J.-P. Armspach, 3-d deformable image registration: a topology preservation scheme based on hierarchical deformation models and interval analysis optimization. IEEE Trans. Image Process. **14**(5), 553–566 (2006)
19. N. Paragios, S. Osher, *Geometric Level Set Methods in Imaging Vision and Graphics* (Springer, New York, 2003)
20. J. Santamariaa, O. Cordonb, S. Damasb, A comparative study of state-of-the-art evolutionary image registration methods for 3d modeling. Comput. Vis. Image Underst. **115**(9), 1340–1354 (2011)
21. M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation. J. Electron. Imaging **13**(1), 146–168 (2004)
22. H. Sundar, H. Litt, D. Shen, Estimating myocardial motion by 4d image warping. Pattern Recognit. **42**(11), 2514–2526 (2009)
23. Z. Zhang, Iterative point matching for registration of free-form curves and surfaces. Int. J. Comput. Vis. **13**(2), 119–152 (1994)
24. B. Zitová, J. Flusser, Image registration methods: a survey. Image Vis. Comput. **21**(11), 977–1000 (2003)

Chapter 8
Lexicographic Approach Based on Evidence Theory for Blood Cell Image Segmentation

Ismahan Baghli and Amir Nakib

8.1 Introduction

The analysis of microscope cell blood images can provide useful information concerning health of patients; the main different components of blood are White Blood Cells (WBCs), Red Blood Cells (RBCs) and platelets. When a disease and foreign materials infect human bodies, the number of WBCs increases to respond and defend infection. The decrease, of RBCs may for example, indicates a low rate of vitamins. Thus, the blood cell image analysis is an important diagnostic tool that must be done accurately.

In practice, cells are analyzed by an expert during a screening stage. Yet, visual screening stage is a very difficult task, and the low number of abnormal cells compared to high number of cells shows the difficulty of this task for an operator. Consequently, false-negative detection can appear due to the subjective aspect of screening. It is obvious that the use of an efficient image processing method will improve the effectiveness of the analysis, the diagnostic, and save time.

Cell segmentation from background involves subtraction of blood cells and other objects merged in the microscopic images. Nevertheless, illumination inconsistencies, and cell occlusion have made cell segmentation very challenging. Many segmentation methods were proposed in the literature to solve the cell segmentation problem since 1960 [22]. However, most of them are based on few basic approaches: thresholding, feature detection, deformable model fitting, morphological filtering, and region accumulation. For example, the use of Mean Shift and GVF (Gradient Vector Flow snake) in [17], the Gram–Schmidt orthogonalization with a snake algorithm in [27] and a snake algorithm and Zack's threshold in [28], in order to segment

I. Baghli
Laboratory of GBM, University of Tlemcen, Tlemcen, Algeria

A. Nakib (✉)
Laboratoire LISSI, Université Paris Est, 122 Rue P. Armangot, 94400 Vitry-sur-Seine, France
e-mail: nakib@u-pec.fr

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology,*
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_8

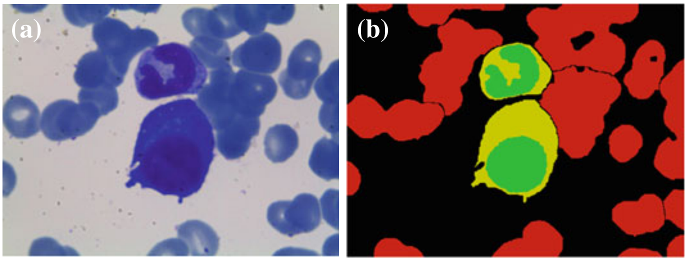


Fig. 8.1 Cell's extraction. **a** Original image **b** Segmented image

nucleus and cytoplasm separately. Two thresholds to remove RBC and background and separate the detected WBC into nucleus and cytoplasm in [25], FCM (Fuzzy C-Means) and evidence theory to segment cells in [4], and a segmentation of WBC with an RBC size estimation and Otsu threshold preceded by nucleus enhancer in [14]. In this paper, we have been more interested by the watershed transform, which makes part of region accumulation approach. The watershed transformation segmentation is considered as a powerful method. Its principle consists in iteratively adding connected points from labelled region, initially marked by seed points. However, one problem of this technique is the over-segmentation,which is caused by the existence of numerous local minima in the image normally due to the presence of noise, and thus, requires further processing [9, 34].

In this work, an hybridization of the watershed transformation and the classification under uncertainty in a lexicographic order to extract from a microscopic image blood cells, the Nucleus, the cytoplasm (both components of the WBC), and RBCs (Red Blood Cell) from the background (see Fig. 8.1; Nucleus, Cytoplasm, RBC and background are colored with green, yellow, red and black respectively). A short description of this work was presented in [3]. A similar problem was already treated in literature where a marker controlled watershed is applied to segment RBC in [30], and to segment WBC in [11], the obtained regions were merged by granularity criterion to segment WBC. However, the performance of the method depends on the quality of the post processing procedure of binary image in [24]. A color watershed growing after a clustering of color's plane for cells segmentation was proposed in [19], where the use of a hybrid gradient (Texture and intensity) as a marker for the watershed transformation to segment WBC in [10], here in, the database should be extended to have more concluded results. In [23] a hybrid method based on the saturation of gradient extraction, the morphological reconstruction and the watershed transform, was used, however, it requires the regularization of gradient magnitude to locate the WBC. In [7, 8] two schemes for nucleus segmentation (watershed transform and level set method) and two schemes of cytoplasm segmentation (granulometric analysis and morphological operators), the estimation of the cytoplasm contour and the cytoplasm mask needs improvement. In [29] Otsu's method was used to segment RBC and the watershed transform to separate overlapping cells. The marker-controlled watershed by internal and external markers to segment RBC from images acquired by a DHM camera (Digital Holographic Microscopy) was proposed in [35], however, the regularization of the RBC storage parameters is need. In [33] the internal marker was obtained by morphological operators (opening, erosion and dilation) on binary image and the external marker was extracted by the distance and the watershed transform; the morphological watersheds and the regions merging are based on the dissimilarities of watershed regions to segment cells. This method requires a threshold fit to avoid under-segmentation. In [31] three different masks are used on three different watershed transform in order to segment WBC, RBC and platelets, however, these masks need to be well fitted to have a good performance. In [20, 21] the authors proposed a watershed based on map distances (closing and erosion operations on binary image) for the final division of the image into catchment's basins, each corresponding to one cell.

In this work, the use information coming from different sources: watershed energy, color information in different colors spaces' in order to segment the blood cell images is proposed. In this work, the problem related to the acquisition techniques are not treated, we only focus on processing the image database at hand. The processed data base is real images acquired from four different patients suffering from blood cancer. In blood smear, number of red cells is many more than white blood cells. For example an image may contain up to 100 red cells and only 1 to 3 white cells, which lead to a great overlap of red blood cells. Moreover, image quality is dependent on blood fixation over the blade (cells may be shredded), also microscope quality, i.e a bad adjustment and/or maintenance gives noised images. The use of the lexicographic allows to mix the different information hierarchically, therefore, at each step the search of the optimal solution (optimal segmentation) is reduced. Moreover, we use the evidence theory to add uncertainty about the solution provided by the different sources' before ending the segmentation process.

The rest of the paper is organised as follows. The motivation of our method is introduced in Sect. 8.2. The Proposed method is exposed in Sect. 8.3, while the experimental results, discussion, and comparison study are presented in Sect. 8.4. We conclude this paper in Sect. 8.5.

8.2 Background

Segmentation is an important processing step in many medical researches and clinical applications where decision making is critical. The watershed segmentation consider the minima of the image as the objects of interest and the maxima as the separation boundaries between objects. However, the existence of numerous local minima cause an over-segmentation. Color features of the watershed regions offers the possibility of merging those similar. The different color representation led to the handling of the uncertainty resulting from imprecise, incomplete, vague and complementary information. The data fusion process permit to take advantage of color uncertainty related to the oversegmented region, and hence, get the best decision among the considered hypothesis.

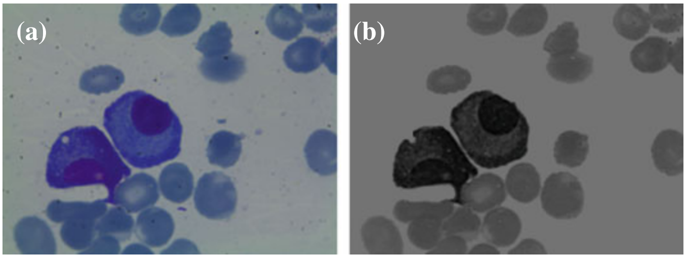


Fig. 8.2 **a** Original image **b** Image projection on Magenta channel (CMYK space)

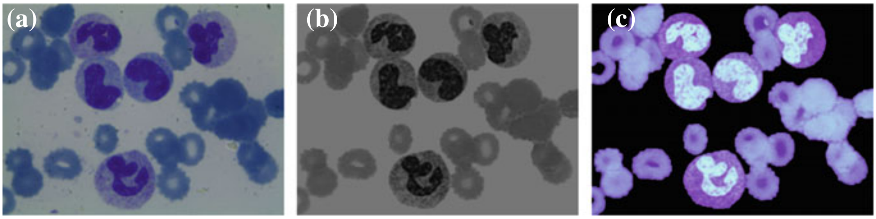


Fig. 8.3 Illustration of the projection of the microscopic dataset in the different color spaces. **a** Original image **b** Image projection on Magenta channel **c** Image projection on HSV color space

8.2.1 Projection on Different Color Spaces

Color descriptors are among the most important features used in image processing. They provide powerful information that often simplifies object detection and identification. Each pixel of a color image is specified in a color space, which serves as a color coordinate system, and the commonly used color space is the RGB color space.

It's always difficult to predict the behavior of the method on a given color space, since each color space has an interest with different stimulus [18].

In cell blood images, the RBCs have always the same appearance, however, WBCs have diverse classes such as basophil, eosinophil etc. Those differences mainly concern the shape of Nuclei and the color of Cytoplasm. We take advantage of different color's representation, which are RGB,HSV and CMYK, to made distinction between WBCs (nucleus and cytoplasm) and RBCs. See Fig. 8.2: RBCs are lighter than the WBC on the Magenta channel. And see Fig. 8.3: all WBC's cytoplasm are similar to RBC in the Magenta Channel, but the difference is clearly made in the HSV projection.

8.2.2 Recall on Evidence Theory

The aim of data fusion is to improve the quality of decisions by increasing the amount of global information while decreasing its uncertainty, using the information redundancy and complementariness between sources. Therefore, it is a technique to integrate heterogeneous data from different sources, in order to optimize the estimation. The fusion is based on uncertain reasoning, developed in bayesian probability theory, Dempster–Shafer evidence theory and other. The Bayesian approach has a decision-making theory, but it requires complete knowledge of combined conditional probabilities and specification of the priori knowledge of probability distribution proving that a piece of evidence is present. Besides, the main limitation of the Bayesian approach is that it cannot model imprecision. That is, the Bayesian probability theory cannot measure a body of evidence with an imprecision on probability measurement. The evidence theory is a general extension of Bayesian theory which offers a number of advantages compared to bayesian theory that can robustly deal with incomplete data [26]. There are three main reasons why the evidence theory should be taken into account when it comes to information fusion. First of all, since the evidence theory supports the representation of both imprecision and uncertainty, it is considered to be a more flexible and general approach than the traditional probability theory. Secondly, the evidence theory offers the possibility of coming up with the probabilities of a collection of hypotheses, whereas a classical probability theory only deals with one single hypothesis. Finally, the major strength of the theory is its ability to deal with ignorance and missing information.

The Dempster–Shafer evidence theory was originally developed by Dempster, who concerned about the lower and upper probabilities, and later Shafer made his contribution by offering belief functions to model uncertain knowledge on the basis of mathematical foundations [13]. It is based on two main ideas: obtaining a degree of belief on a hypothesis (called the "mass function") from subjective probabilities, and Dempster's rule (orthogonal sum) for combining such degrees of beliefs when they are based on independent items of evidence [12]. In a more formal way:

Let Ω be a finite set of mutually exclusive and exhaustive hypotheses, called the frame of discernment:
$\Omega = \{H_1, \ldots, H_n\}$

The power set $P(\Omega)$ composed with the 2^n propositions of Ω defined by:
$P(\Omega) = \{\phi, \{H_1\}, \{H_2\}, \ldots, \{H_n\}, \{H_1 \cup H_2\}, \{H_1 \cup H_3\}, \ldots, \Omega\}$

When the frame of discernment is determined, the Basic Probability Assignment (BPA) or mass function, is defined as a mapping of the power set $P(\Omega)$ to a number between 0 and 1, i.e.: $m : P(\Omega) \rightarrow [0, 1]$, that satisfies conditions below:

$$\begin{cases} m(\phi) = 0 \\ \sum_{A \subseteq \Omega} m(A) = 1 \end{cases} \tag{8.1}$$

If A is considered as a subset of $\Omega(A \subseteq \Omega)$, $m(A)$ indicates the degree of belief that is assigned to the exact set A and not to any subsets of A.

There are also the belief and plausibility function, which are sometimes referred to as the lower bound and upper bound on the probability of a subset, respectively, such that:

$$Bel(A) \ \leq Prob(A) \ \leq Pl(A) \tag{8.2}$$

Those two functions are derived from mass function:

$$\begin{cases} Bel(A) = \sum_{B\subseteq\Omega} m(B) \\ Pl(A) = \sum_{A\cap B\neq\phi} m(B) \end{cases} \tag{8.3}$$

Let a part from the ability of the evidence theory to compute a degree of uncertainty, it is also used by combining different evidences to increase the certainty value, which allows more accurate decisions.

The combination of the different evidences using Dempster's rule in which two independent information sources (m_1 and m_2) are fused to create a new belief function, is presented below:

$$m(C) = \frac{\sum_{A\cap B=C} m_1(A) \times m_2(B)}{1 - \sum_{A\cap B=\phi} m_1(A) \times m_2(B)} \tag{8.4}$$

Where the denominator can be interpreted as a conflict criterion between independent evidences to be combined [12]. The increase of $m(A)$ corresponds to stronger the belief on the proposition A.

An important key point when using evidence theory is to well model the knowledge given by different sources of information S in order to initialize the associated BPA. Moreover, modeling the belief on power sets may result in exponential complexity of operations made on BPAs with respect to the number of classes. However, the BPA is often interpreted as a piece of confidence associated with a certain hypothesis. Since the definition of the BPA function in evidence theory remains a largely unsolved problem and there is not one general solution.

8.3 The Proposed Framework

In this section, we present the proposed framework (Fig. 8.4), called ESA (Evidential Segmentation Algorithm). Herein, we use the power of the evidence theory to classify the pixels of a given watershed transformed image. The well known watershed segmentation gives an over-segmentation caused by the existence of numerous local minima. However, considering both over-segmented regions and the different color projection as shown in Sect. 8.2.1, i.e. combine the information coming from different sources: watershed energy, color information in different colors spaces would allow to enhance the segmentation of the blood cell images. Moreover, Evidence theory is widely used in data fusion and pattern recognition because it provides strong and native modeling of imprecision. The completeness and effectiveness of the transformed representations for the underlying information of the source images are crucial to the fusion quality. Therefore, the information extracted from any source in favor of, or against, a given class assignment must be combined with that of available sources to infer the likelihood of this assignment.

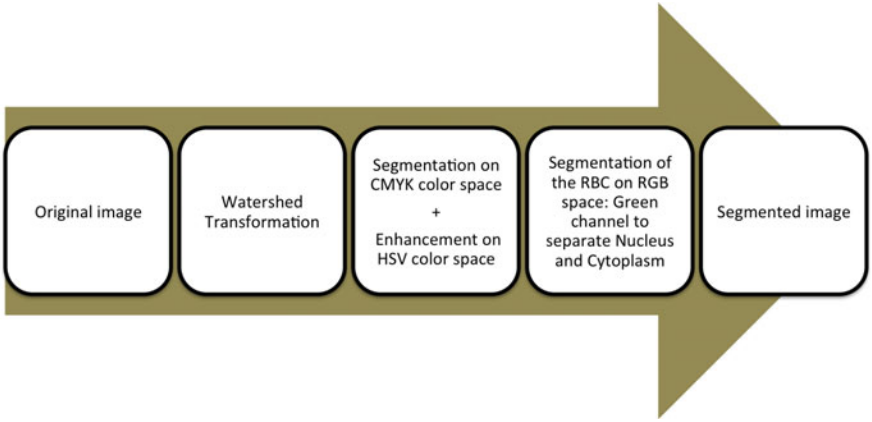

Original image
Watershed Transformation
Segmentation on CMYK color space + Enhancement on HSV color space
Segmentation of the RBC on RGB space: Green channel to separate Nucleus and Cytoplasm
Segmented image

Fig. 8.4 The flowchart of the evidential segmentation algorithm

Thus, the added value of using the evidence theory instead of a traditional classification method, lies in its ability to integrate the colors and lightning uncertainties, and hence, increasing the belief on a given hypothesis by computing the degree of belief between the color distance.

8.3.1 Initial Segmentation Using Watersheds

At this step of the proposed algorithm, our goal is have an initial segmentation that provides connected contours. To do so, the morphological watershed transform is applied to the original image. Indeed, this method allows having an initial portioned image into regions. In the following, the fast watershed detection algorithm [32].

Let I be the original image. The lines separating the catchment basins that belong to different local minima define the Watersheds. In other terms, a local optimum M at intensity level l in I is a connected set of pixels with intensity with intensity l, such that it is impossible to reach a pixel of intensity l' without having to pass from a pixel of intensity l'', where $l' \leq l \leq l''$. The catchment basin $D(M)$ associated with the minimum M is a set of pixels, such that, if a drop of water falls at any pixel in $D(M)$, then it will flow down to the minimum. The watersheds computation algorithm used here is based on immersion simulations [32], that is, on the recursive detection and fast labeling of the different catchment basins using queues. The algorithm consists of two steps: sorting and flooding. At the first step, the image pixels are sorted in increasing order according to their intensities. At the second step, the pixels are quickly accessed in increasing intensity order (immersion) using the sorted image and labels are assigned to catchment basins. The label propagation is based on queues constructed using neighborhoods [32]. After the application of the watersheds algorithm a tessellation of the original image into its different catchment basins is obtained, where each one described by a unique label. Among the image watershed points, only those located exactly half-way between two catchment basins are given a special label [32]. Then, to obtain the final image tessellation, the watersheds are removed by assigning their corresponding points to the neighboring catchment basins.

The high sensitivity of the watersheds algorithm to the intensity variation yields a very large number of catchment basins, leading to oversegmentation.

In the literature, most of the earlier attempts in oversegmentation reduction use markers to select regional minima prior to the application of the watershed transform. Although markers have been successfully used in segmenting many types of images, their selection requires either careful user intervention or explicit prior knowledge on the image structure. In our approach, image oversegmentation is viewed as an initial image partition to which an evidential region-merging procedure is applied (see next section). The larger the initial oversegmentation, the higher the probability of false region merges during merging. In addition, It is obvious that the computational complexity of region merging depends on the size of this initial partition, and consequently the smallest possible oversegmentation size is suitable. One way to limit the size of the initial image partition is to prevent oversegmentation in homogeneous regions, where the gradient magnitude is low.

8.3.2 Evidential Regions Merging

In this work, some models from the literature were tested: Denoeux [5], Shafer [12], and Appriou [2]. Then, the Denoeux model (distances between samples, and class center model the knowledge, on patterns) provided the best results for our blood cell images.

Thus, the considered mass function is defined by:

$$m(H_i) = \alpha_i \times \exp(-\gamma_i \times dist(x, x_i)) \tag{8.5}$$

with $\alpha_i \in [0, 1]$ is the weakening parameter, $\gamma_i > 0$ related to the hypothesis H_i, $\gamma_i = \frac{1}{d_i}$ with, d_i the average distance between representatives samples of (H_i), and $dist(x, x_i)$ the euclidian distance between average values of the processed zone x, and x_i the average value of a representatives samples of H_i.

The proposed mergin region procedure is based on a lexicographic order that allows to segmentation the different components. The three steps of the proposed procedure are described below:

1. We associate to each labeled region a degree of belief for the hypothesis WBC and RBC. In most cases, we noticed that the magenta component from the CMYK color space allows to differentiate RBCs from others cells (see Fig. 8.2: RBCs are lighter than the WBC). However, in some cases the cytoplasm is confused with RBC.

2. New projection is performed on the HSV (Hue, Saturation, Value) color space and an evidential classification is applied. To do so, a new mass function that consists in representing each region by its average value in the HSV space is considered:

$$m(H_i) = \frac{1}{\lfloor R_i \rfloor} \sum_{j \in R_i}^{\lfloor R_i \rfloor} I(j) \tag{8.6}$$

   where H_i is the hypothesis on the region R_i, $\lfloor R_i \rfloor$ is the cardinal of the region R_i, and $I(j)$ is the intensity of the pixel j belonging to the considered region. (see Fig. 8.3: Cytoplasm of cell on top is similar to RBC, but difference is clearly made in the HSV projection). Then, the following heuristic is applied:
   if $m(H_i) > 0.5$, then the region is reclassified as WBC, **Else**, it is left as RBC region.
3. For each identified WBC, the average value of the green channel (observed that the green channel represent as well the nucleus) is computed to separate Nucleus and Cytoplasm.

8.4 Results and Discussion

The aim of our experiments is to apply the proposed method on a real image database (87 images). The blood smears blades have been fixed by the MGG (May-Granwald Giemsa) coloration. We have obtained 24-bits RGB images of 1024 × 768 pixels by the use of LEICA environment (camera and microscope with a 100x magnification).

Eight blood smears images (Fig. 8.5a-1–a-8) are presented to illustrate the obtained results of the proposed algorithm. Note that Nucleus, Cytoplasm, RBC and background are colored with green, yellow, red and black, respectively; as it can be seen in Fig. 8.5b-1–b-8 and c-1–c-8.

In these experiments, the goal is to identify the red blood cell regions and the white blood cell regions, especially, the white blood cells since, it remains important for example for the diagnosis of the blood cancerous (our current project). Thus, we do not deal with separating touching red blood cells. We handle with several kinds of white blood cells, the difference may concern the shape of nucleus and the color of cytoplasm. In Fig. 8.5, the shape of nucleus is circular or oval in images (a)-1, bottom (a)-2, (a)-4, top and middle (a)-5, (a)-6, (a)-7, (a)-8, and has kidney shape in top (a)-2 and (a)-3. Regarding the color of cytoplasm, one can notice the dark and the light cytoplasm, in some cases within the same image (for example, image (a)-2).

The obtained segmentation results compared to the ground truth are good enough, especially in the case of images (b)-1, (b)-3, (b)-4 and (b)-6: cells were totally recognized. The proposed algorithm provides good results in the case overlapping cells in all images (RBC vs. WBC and WBC vs. WBC), except in image (b)-5 where the WBC in the middle is misclassified. Furthermore, some limitations can be observed: a small part nucleus has been identified as cytoplasme in images (b)-2 (the top one)

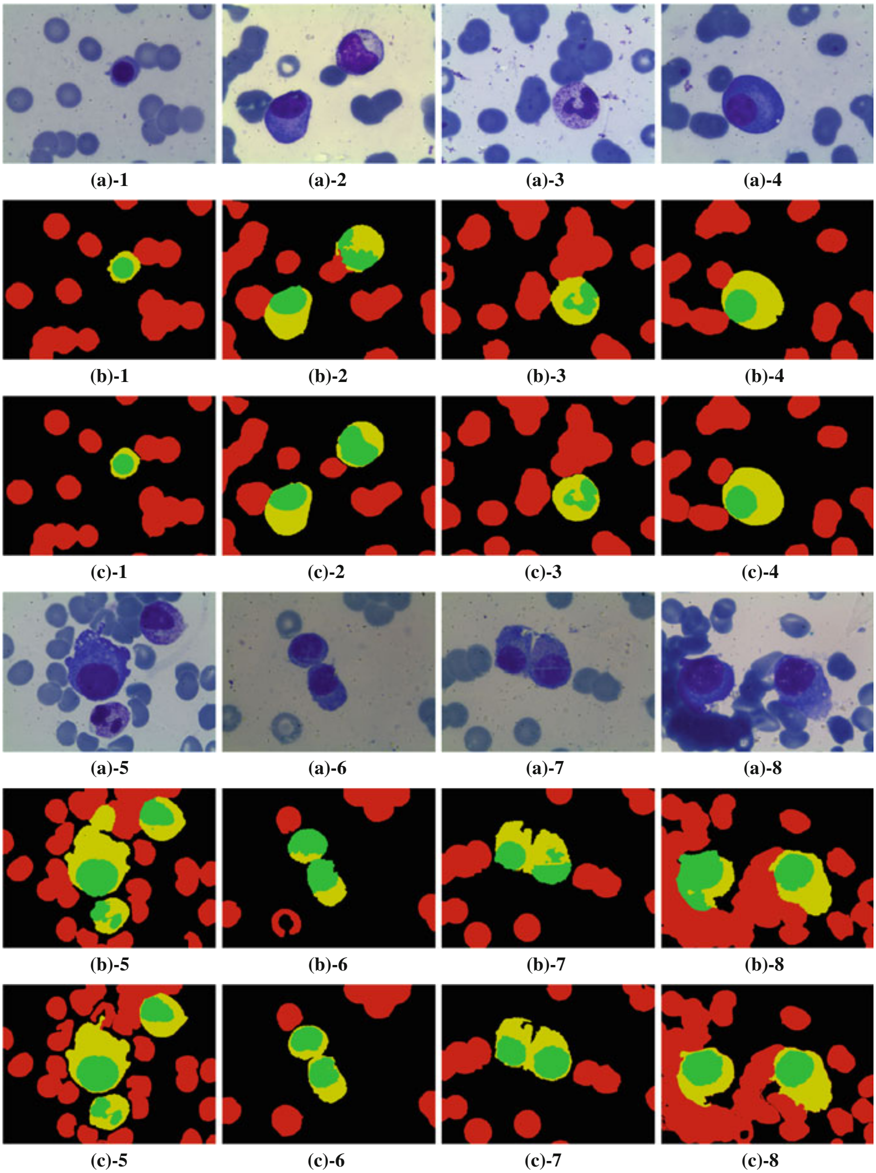
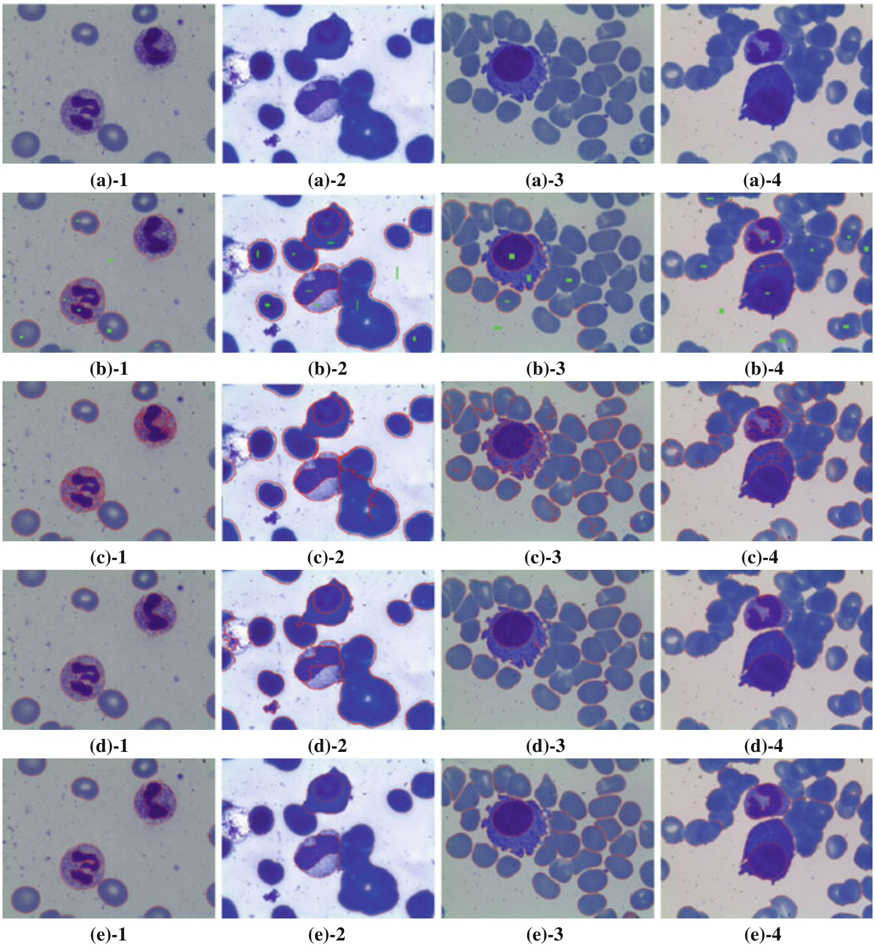


Fig. 8.5 ESA results: **a** Original image **b** ESA segmentation **c** Ground truth

and (b)-7 (the right one), and in the left of image (b)-8 a part of cytoplasm has been identified as Nucleus. However, it does not affect the identification of the WBC.

One can remark that the same kind of WBC is present in images bottom of (a)-2, (a)-4, middle of (a)-5, bottom of (a)-6, (a)-7 and (a)-8, this white blood cell,


(a)-1 (a)-2 (a)-3 (a)-4

(b)-1 (b)-2 (b)-3 (b)-4

(c)-1 (c)-2 (c)-3 (c)-4

(d)-1 (d)-2 (d)-3 (d)-4

(e)-1 (e)-2 (e)-3 (e)-4

Fig. 8.6 Comparison results: **a** Original image **b** Controlled watershed (*green* markers) **c** Stochastic watershed **d** ESA segmentation **e** Ground truth

commonly named *Plasma cell*, is present in normal and distorted shape, fortunately, the two shapes were well recognized, which is a promising results.

We have compared the performance of the ESA to those of others methods: stochastic watershed [1] and controlled watershed [16]. To illustrate the comparison of the performance, we present in Fig. 8.6 an example on four blood cell images ((a)-1 to (a)-4). The obtained edges of the controlled watershed, the stochastic watershed, the ESA segmentation and the ground truth are superimposed on the originals images of Fig. 8.6 on (b), (c), (d) and (e), respectively.

In order to obtain a good segmentation with the controlled watershed, the user must mark each region, manually, sometimes several times: in Fig. 8.6b-1, the two WBCs are segmented differently; the top one (one marker) is segmented as one region, i.e, cytoplasm and nucleus are not separated, while the nucleus of the bottom one (two markers) is well segmented, unlike the cytoplasm whose the right part is merged with the RBC below. So, many cells were skipped because they had not a marker on them (clearly shown on Fig. 8.6b-1–b-4), thus, the segmentation quality of the controlled watershed is closely linked to position, and to the number of markers. The stochastic watershed (Fig. 8.6c-1–c-4), and even if it is not our aim) seems to be suited to separate touching red blood cells, as it can bee seen also in image (c)-3. However, it oversegments cytoplasm regions (see the two WBCs on image (c)-1)). When applying the ESA (results in Fig. 8.6d-1–d-4), one can notice that the contours are well detected, even if a slight overtaking between nucleus and cytoplasm on the WBC in the bottom of image (d)-2, nevertheless, the big overlap between RBCs on images (a)-(2) and (a)-4 are not confused with nucleus. Moreover, it is detected as an RBC region.

It should be noticed that the images (a)-2 and (a)-4 of Fig. 8.6 are hard to be segmented because of the big overlapping of red blood cells, which lead to be confused with nucleus of white blood cells. This can be solved using geometric priori approaches.

In summary, to obtain a good segmentation with the controlled watershed, it is necessary to mark each region manually, sometimes several times, thus, this technique cannot be non-supervised, and depends on the user expertise. In the other hand, for the comparison with the stochastic watershed transform, we used 20 markers and 20 iterations. Then, we kept the most redundant arcs in order to achieve the segmentation. The true edges are, therefore, enhanced and the false lines are weakened. However, the number of realizations and the number of markers affect the result. So, if the execution time of both techniques is similar after our optimization, increasing them will dramatically increase the execution time when using a stochastic watershed.

8.4.1 Performances Evaluation

A performance metrics are required to determine the similarity between the ground truth data and the segmentation results. We have evaluate our algorithm in terms of Dice score [6] and Jaccard similarity [15]. Those metrics, were first developed to measure similarities in ecological studies, but their nature of set operations made them applicable for segmentation similarity.

The Dice score is a spatial overlap index and a reproducibility validation metric it also known as the mean overlap. The Jaccard similarity is the ratio of the intersection of segmented area A and ground truth area B, to the union of segmented area A and ground truth area B. It takes the qualitative aspect overlap into account. The corresponding formulas are given below:

$$Dice\ score = 2 * \frac{TP}{((TP + FP) + (TP + FN))} \tag{8.7}$$

$$Jaccard\ similarity = \frac{TP}{(FP + TP + FN)} \tag{8.8}$$

where, TP (True Positives): intersection between segmentation and ground truth), FP (false positives): segmented parts not overlapping the ground truth, FN (false negatives): missed parts of the ground truth, and TN (true negatives): part of the image beyond the union between segmentation and ground truth).

These two metrics values range from 0, indicating no spatial overlap between two sets of segmentation results, to 1, indicating complete overlap. When applied to all our data, the Dice score and the Jaccard similarity have achieved a promising results: 0.93 and 0.87, respectively.

Figure 8.7 points out a sensitivity analysis of segmentation quality against the variations of the parameter $\gamma_i = d_i^{-1}$ defined in Eq. 8.5, where d_i is the average distance between the samples of the class i. As it can be seen, the fitting of this parameters is more important in the case of the magenta channel than in the HSV space. Indeed, the HSV projection is made only to check if the quality of the RBC classification.

The illustration of the analysis is presented in Fig. 8.7b–k where all cases are reported. A summery of the experiments is below:

- Case (c) to case (e) the γ_i for RBC on Magenta channel is equal to 10×10^{-3} and γ_i for WBC is superior to 4.8×10^{-3}, equal to 4.8×10^{-3} and inferior to 48×10^{-4}, respectively.
- Case (e) to case (g), the γ_i for WBC on Magenta channel is fixed on the used value 4.8×10^{-3} and $\gamma_i for$ RBC is superior to 16×10^{-3}, between 11×10^{-3} and 15×10^{-3} and inferior to 9×10^{-3}, respectively.
- Case (h) to case (k) the γ_i for RBC on HSV space is superior to 0.1, equal to 0.09, between 0.05 and 0.08 and inferior to 0.04, respectively.

In visual terms, bad segmentations are clearly observed on cases (b), (d), (e), (f), (g), (h) and (k). More on Magenta channel than on HSV space. Further more, case (j) is the same to ours, illustrated on case (c) and case (i). And, in qualitative terms, we have calculated TP rate and accuracy on both WBC data and RBC data of all cases (Fig. 8.8). As we can see, on case (b), (d), (e), (f), (g), (h) and (k) accuracy is at the cost of rate, i.e a good recognition but a bad accuracy and vice versa. However case (f) is comparatively acceptable.

Taking the conclusion on magenta channel into account, we have analyzed all possible values of degree of belief. To this end, we have calculated degrees of belief on both examples of white blood cell and red blood cell with respect of all possible values of the processed region. Figure 8.9 show the different changes. As one can see, the degrees of belief of red blood cell are decreasing faster than the degrees of belief of white blood cell, we explain that by the observed closeness of nucleus appearence on almost all images.

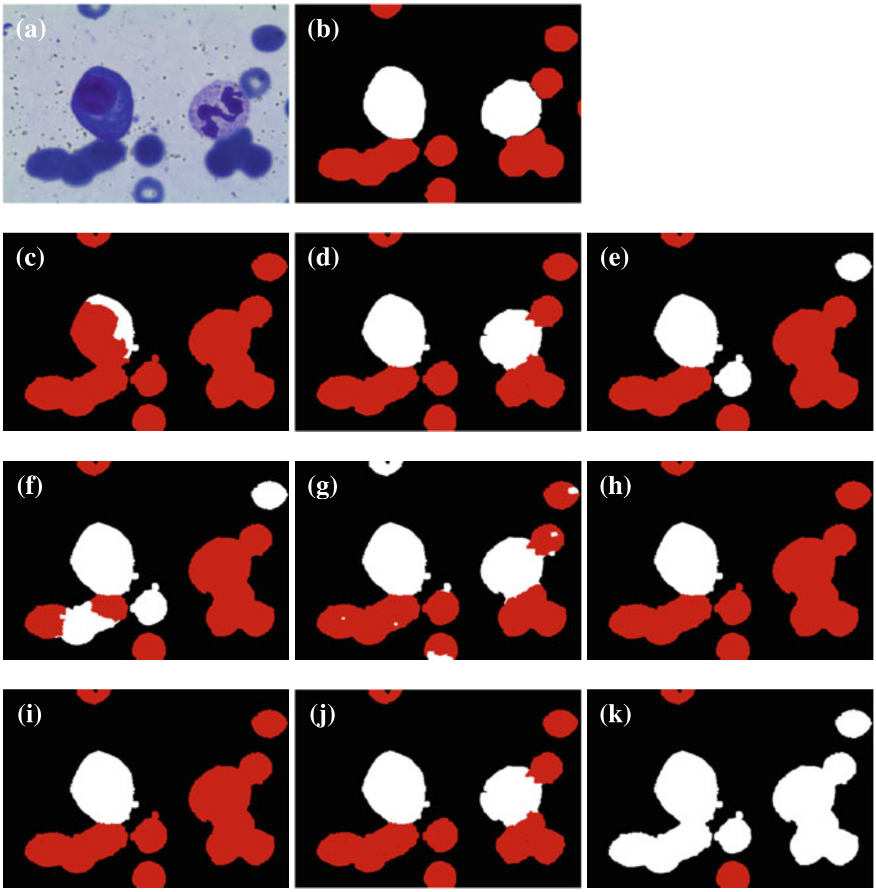


Fig. 8.7 Illustration of the sensitivity against the parameters γ_i RBC and γ_i for WBC: **a** Original image, **b** Ground truth image, **c**–**k** Segmentation results for different values of γ_i for RBC, and γ_i for WBC

The study of this case shows that evidence theory seems to be well suited for blood cell segmentation problem. Moreover, the segmentation results are promising and close to expert segmentation. However, some drawbacks were observed: errors mostly consist of classification of some cytoplasms as RBCs. This is due mainly to the diverse classes of WBCs (shape of Nucleus and the color of Cytoplasm), unlike the RBCs which always have the same appearance. In our work, we focus on the distinction made by color uncertainties. To this end, the power of evidence theory was used on both the Magenta channel and the HSV color space to distinguish red blood cells from white blood cells. Afterwards, the white blood cells were segmented, using the green channel, into Nucleus and Cytoplasm.

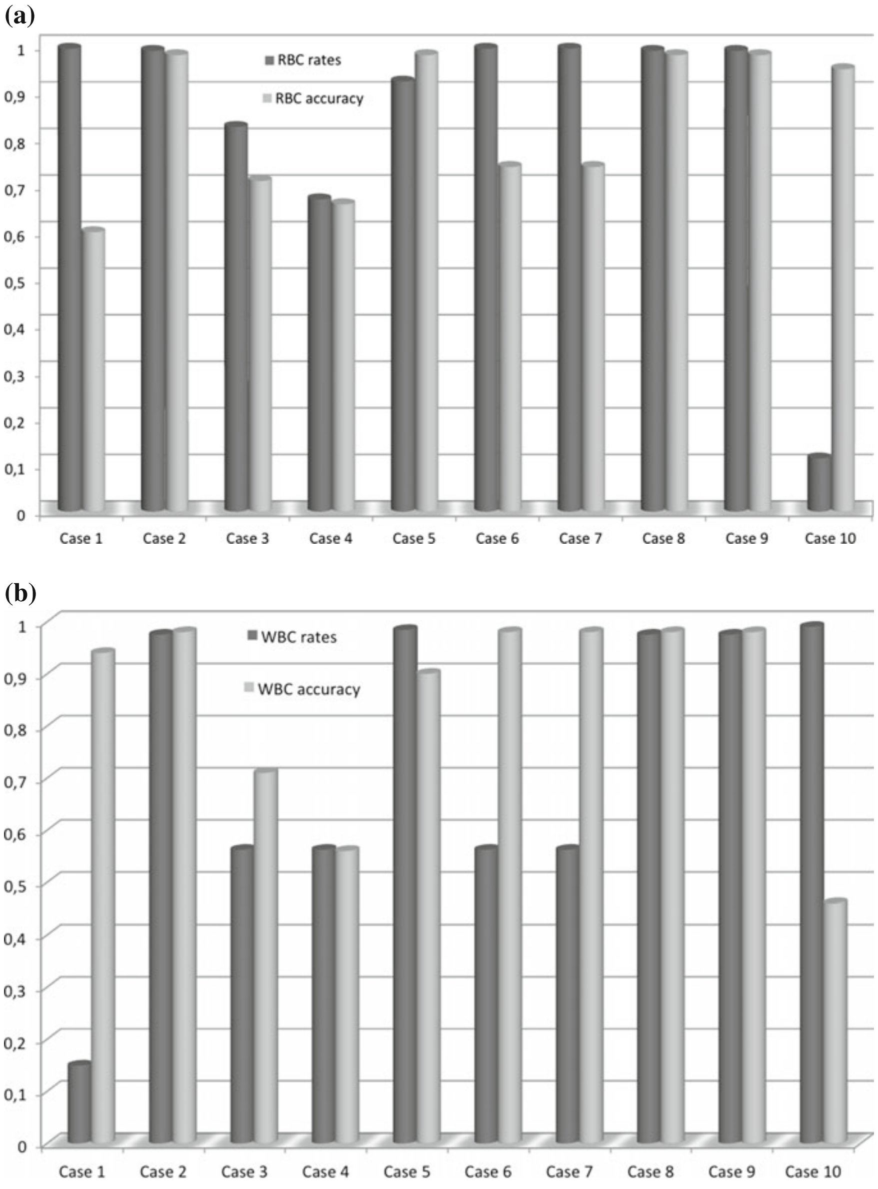


Fig. 8.8 Rate and accuracy: **a** White Blood Cell data **b** Red Blood Cell data

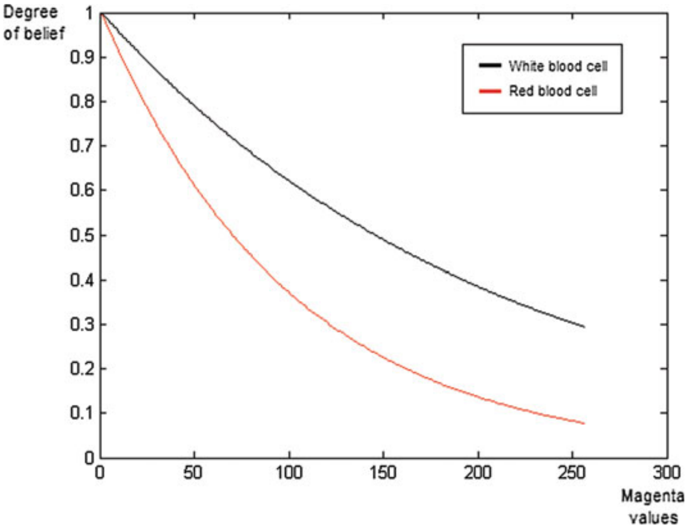


Fig. 8.9 The impact of distances on the degree of the belief

8.5 Conclusion

In this work, a new lexicographic based segmentation approach, called ESA, was proposed. This proposed method has allowed solving the segmentation problem of blood microscopic images. In the application at hand, the goal is to extract cells' component. The efficiency of the evidence theory was used to fuse information from watershed transformation and different colour spaces. In work under progress the use of geometric a priori to solve overlapping problem. Another issue that we dealing with is to find BPA models.

References

1. J. Angulo, D. Jeulin, Stochastic watershed segmentation, in *Proceedings of the 8th International Symposium on Mathematical Morphology (ISMM'2007)* (2007), pp. 265–276
2. A. Appriou, Multisensor signal processing in the framework of the theory of evidence, *Application of Mathematical Signal Processing Techniques to Mission Systems*, vol. 216, NATO/RTO - Lecture Series (1999), pp. 5–31
3. I. Baghli, A. Nakib, E. Sellam, M. Benazzouz, A. Chikh, E. Petit, Hybrid framework based on evidence theory for blood cell image segmentation, in *Proceedings of the SPIE 9038, Medical Imaging 2014, San Diego, (USA)*, 15–19 Feb 2014. doi:10.1117/12.2042142
4. S. Ben Chaabane, M. Sayadi, F. Fnaiech, E. Brassart, Dempster-shafer evidence theory for image segmentation: application in cells images. Int. J. Inf. Commun. Eng. **5**(2), 126–132 (2009)
5. T. Denoeux, A k-nearest neighbor classification rule based on dempster-shafer theory. IEEE Trans. Syst. Man Cybern. **25**(5), 804–813 (1995)

6. L.R. Dice, Measures of the amount of ecologic association between species. Ecology **26**(3), 297–302 (1945)
7. L.B. Dorini, R. Minetto, N.J. Leite, White blood cell segmentation using morphological operators and scale-space analysis, in *Proceedings of the 20th Brazilian Symposium Computer Graphics Image Processing* (2007), pp. 100–107 (2007)
8. L.B. Dorini, R. Minetto, N.J. Leite, Semiautomatic white blood cell segmentation based on multiscale analysis. IEEE J. Biomed. Health Inform. **17**(1), 250–256 (2013)
9. O. Dzyubachyk, W.A. Van Cappellen, J. Essers, W.J. Niesen, E. Meijering, Advanced level-set based cell tracking in time-lapse fluorescence microscopy. IEEE Trans. Med. Imaging **29**(3), 852–867 (2010)
10. W. Gao, Y. Tang, X. Li, Segmentation of microscopic images for counting leukocytes, in *Proceedings of the 2nd International Conference on Bio-informatics and Biomedical Engineering (ICBBE'08)* (Shangai, China, 2008), pp. 2609–2612
11. M. Ghosh, D. Das, S. Mandal, C. Chakraborty, M. Pal, A.K. Maity, S.K. Pal, A.K. Ray, Statistical pattern analysis of white blood cell nuclei morphometry, in *Proceedings of the 2010 IEEE Students Technology Symposium, IIT Kharagput*, pp. 59–66, 3–4 April 2010
12. S. Glenn, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, 1976)
13. J.W. Guan, D.A. Bell, *Evidence Theory and Its Applications* (North-Holland, New York, 1991)
14. D.-C. Huang, K.-D. Hung, Y.-K. Chan, A computer assisted method for leukocyte nucleus segmentation and recognition in blood smear images. J. Syst. Softw. **85**, 2104–2118 (2012)
15. P. Jaccard, Etude comparative de la distribution florale dans une portion des alpes et des jura. Bulletin de la sociÈtÈ Vaudoise des Sciences Naturelles **37**, 547–579 (1901)
16. M. Kaur, G. Jindal, Medical image segmentation using marker controlled watershed transformation. IJCST **2**(4), 548–551 (2011)
17. B.C. Ko, J.-W. Gim, J.-Y. Nam, Automatic white blood cell segmentation using stepwise merging rules and gradient vector flow snake. Micron **42**, 695–705 (2011)
18. A. Koschan, A. Mongi, *Digital Color Image Processing* (Wiley, New York, 2008)
19. O. Lezoray, Supervised automatic histogram clustering and watershed segmentation. Application to microscopic medical color images. Image Anal. Stereol. **22**, 113–120 (2003)
20. T. Markiewicz, S. Osowski, B. Marianska, L. Moszczynski, Automatic recognition of the blood cells of myelogenous leukemia using SVM, in *Proceedings of the IEEE International Joint Conference on Neural Networks. IJCNN'05*, vol. 4 (IEEE, 2005), pp. 2496–2501
21. T. Markiewicz, S. Osowski, B. Mariańska, White blood cell automatic counting system based on support vector machine, *Adaptive and Natural Computing Algorithms* (Springer, Berlin, 2007), pp. 318–326
22. E. Meijering, Cell segmentation: 50 years down the road. IEEE Signal Process. Mag. **29**(5), 140–145 (2012)
23. L.H. Nee, M.Y. Mashor, R. Hassan, White blood cell segmentation for acute leukemia bone marrow images, in *International Conference on Biomedical Engineering (ICoBE'12), Penang, Malaysia*, pp. 357–361, 27–28 Feb 2012
24. J.B. Nemane, V.A. Chakkarwar, A novel method of white blood cell segmentation and counting. Int. J. Adv. Comput. Eng. Commun. Technol. **1**(1), 44–49 (2012)
25. R. Nisha, D. Bryan, E. Salama Mohammed, T. Tasdizen, Isolation and two-step classification of normal white blood cells in peripheral blood smears. J. Pathol. Inform. **3**(1), 13 (2012)
26. A. Rakar, D. Juricic, P. BallÈ, Transferable belief model in fault diagnosis. Eng. Appl. Artif. Intell. **12**, 555–567 (1999)
27. S.H. Rezatofighi, H. Soltanian-Zadeh, Automatic recognition of five types of white blood cells in peripheral blood. Comput. Med. Imaging Graph. **35**, 333–343 (2011)
28. F. Sadeghian, Z. Seman, A.R. Ramli, B.H. Abdul Kahar, M.I. Saripan, A framework for white blood cell segmentation in microscopic blood images using digital image processing. Biol. Proced. Online **11**(1), 196–206 (2009)
29. S.S. Savkare, S.P. Narote, Automatic system for classification of erythrocytes infected with malaria and identification of parasite's life stage, in *Procedia Technology: 2nd International Conference on Communication, Computing and Security (ICCCS'12)* (2012) pp. 405–410

30. J.M. Sharif, M.F. Miswan, M.A. Ngadi, M.S.H. Salam, M.M.B.A. Jamil, Red blood cell segmentation using masking and watershed algorithm: a preliminary study, in *Proceedings of ICoBE, Penang, Malaysia*, pp. 258–262, 27–28 Feb 2012
31. H. Tulsani, S. Saxena, N. Yadav, Segmentation using morphological watershed transformation for counting blood cells. Int. J. Comput. Appl. Inf. Technol. **2**(3), 28–36 (2013)
32. L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. **13**, 583–598 (1991)
33. W. Wang, H. Song, Q. Zhao, A modified watersheds image segmentation algorithm for blood cell. Int. Conf. Commun. Circuits Syst. Proc. **1**, 450–454 (2006)
34. Q. Wu, F.A. Merchant, K.R. Castleman, *Microscopic Image Processing* (Academic Press, Burlington, 2008)
35. F. Yi, I. Moon, B. Javidi, D. Boss, P. Marquet, Automated segmentation of multiple red blood cells with digital holographic microscopy. J. Biomed. Opt. **18** (2013). doi:10.1117/1.JBO.18.2.026006

Chapter 9
Medical Image Denoising Using Metaheuristics

Serdar Kockanat and Nurhan Karaboga

9.1 Introduction

In recent years, metaheuristic optimization techniques have attracted much attention from researchers and practitioners and they have been widely used to solve complex or specific optimization problems in all fields, from engineering area to finance [2]. The considerable advantage of metaheuristic algorithms is that they don't need to be adapted in detail for each optimization problem. Therefore, they can be easily used to solve various optimization problems. Metaheuristic techniques play an important role in optimization, because they are very useful in the usage of limited resources, such as time, money and computational complexity. In metaheuristics, to date, a number of algorithms have been developed in the literature, such as the genetic algorithm (GA) [5], simulated annealing (SA) [16], differential evolution (DE) [26], particle swarm optimization (PSO) [15] and the artificial bee colony algorithm (ABC) [8]. In biomedical imaging systems, the common and most studied problem is noise [6]. Noise can occur due to the image acquisition systems or techniques used in biomedical images. Noise elimination is of the almost importance in order to extract correct and useful details from biomedical images for medical experts. To date many studies have dealt with medical image denoising in order to eliminate different types of noise on noisy biomedical images. In the literature, many filtering and transformation methods have been proposed [1, 19].

Spatial domain filtering is one of the methods used for image denoising in the bio-medical imaging area. Spatial filters can be classified as linear and nonlinear filters, such as linear Wiener, nonlinear median or mean filters [21, 28]. However,

S. Kockanat · N. Karaboga (✉)
Sivas Vocational School, Cumhuriyet University, Sivas, Turkey
e-mail: nurhan_k@erciyes.edu.tr

S. Kockanat · N. Karaboga
Department of Electrical and Electronics Engineering,
Erciyes University, Kayseri, Turkey

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_9

the disadvantage of linear filters is that they can cause blurring effects on the filtering images and a number of studies have tried to overcome this situation [20, 24]. Another alternative approach is wavelet based methods which are more successful than spatial filtering for noise elimination [4, 25]. Recently, two dimensional (2D) digital filters have become increasingly important and they have been used in a wide range of applications, such as image enhancement, biomedical image processing, noise reduction and radar signal processing [22]. Normally, 2D digital filters are grouped into two categories: finite impulse response (FIR) and infinite impulse response (IIR) digital filters. The design of the FIR digital filter is easier than that of the IIR filter and it is always stable. On the other hand, the stability is a very important and troublesome problem for IIR digital filters. However, 2D IIR filters can achieve very sharp frequency responses. In the literature, many studies have been conducted in order to design 2D digital filters and they have been used frequently for image denoising [3, 17, 18, 23]. Mastorakis et al. proposed a novel constrained optimization method to design 2D IIR filters. The design problem was transformed into a constrained minimization problem and then it was solved using the genetic algorithm [23]. Das and Konar used particle swarm optimization to optimize the constrained optimization problem in the design of a 2D IIR filter [3]. Kockanat et al. proposed an approach to optimize the filter coefficients of a 2D FIR digital filter using the artificial bee colony algorithm for image denoising [17, 18]. Tzeng introduced an effective genetic algorithm for the design of 2D FIR filters with specified magnitude and group delay responses [27]. In this work, four well-known metaheuristics are used for ultrasound image denoising and their performances compared. Two of these metaheuristics are GA and DE which are evolutionary based algorithms and the other two are swarm intelligence algorithms which are PSO and ABC. The chapter is organized as follows. Section 9.2 explains the proposed noise elimination approach. Section 9.3 describes the parameter adaptation algorithms. Section 9.4 shows the results of the proposed noise elimination approaches. Finally, Sect. 9.5 gives the conclusions.

9.2 Noise Elimination Approach

Noise elimination is one of the most common research problems in digital image processing. It can be explained as the process of removing noise from a noisy image. Figure 9.1 shows the scheme of the proposed noise elimination approach.

As shown in Fig. 9.1, $d(n_1, n_2)$, $x(n_1, n_2)$ and $s(n_1, n_2)$ are the noisy, noise and original signals, respectively. $x(n_1, n_2)$ and $x_1(n_1, n_2)$ are the Gaussian noise signals and they are produced synthetically. $x_1(n_1, n_2)$ is achieved by passing $x(n_1, n_2)$ through the 2D lowpass filter and both signals are correlated with each other. The error signal $e(n_1, n_2)$, is obtained by subtracting the 2D FIR digital filter output signal $y(n_1, n_2)$ from the noisy signal $d(n_1, n_2)$. In Fig. 9.1, the relationship between the input and output signals of the 2D FIR digital filter is expressed as in Eq. 9.1

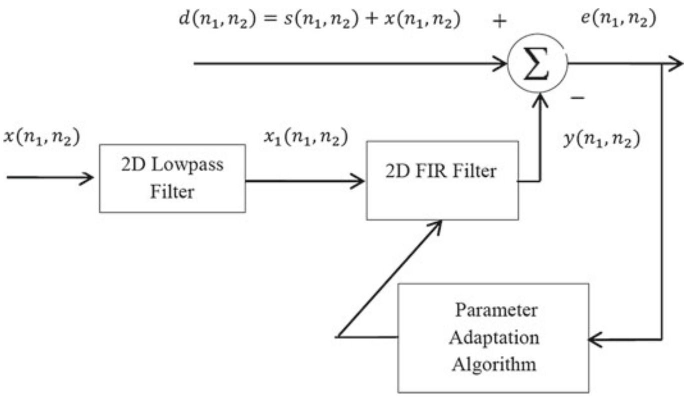


Fig. 9.1 Block scheme of the proposed noise elimination approach

$$y(n_1, n_2) = \sum\nolimits_{m=0}^{M-1} \sum\nolimits_{n=0}^{N-1} h(m, n) x_1(n_1 - m, n_2 - n) \tag{9.1}$$

where $x_1(n_1, n_2)$ and $y(n_1, n_2)$ are the input and output signals of the 2D FIR digital filter, respectively. $h(m, n)$ is the filter coefficient which is represented generally as a matrix and M and N are the filter orders. The noise signal is selected as Gaussian noise due to the specification of the power spectrum. Gaussian noise is an additive, statistical and uncorrelated noise with constant power spectral density. In theoretical study, it has an infinite power which lies in the infinite frequency range. Random fluctuations in the signal introduce the Gaussian noise. As seen in Fig. 9.1, the parameter adaptation algorithm has been used to optimize the coefficient matrix of the 2D FIR digital filter by minimizing the objective function. The objective function is defined as

$$MSE = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (d(n_1, n_2) - y(n_1, n_2))^2 \tag{9.2}$$

where N_1 and N_2 are the size of the input digital images. Mean square error (MSE) is employed as the error function. In order to analyze the quality after the denoising approach, peak signal to noise ratio (PSNR) is calculated between the original signal $s(n_1, n_2)$ and error signal $e(n_1, n_2)$. The formula of PSNR is given as

$$PSNR = 10 \log_{10} \left(\frac{R^2}{MSE} \right) \tag{9.3}$$

where R is the maximum possible value of the signal.

9.3 Parameter Adaptation Algorithm

In this study, the artificial bee colony, particle swarm optimization, genetic algorithm and differential evolution are employed as the parameter adaptation algorithm. These algorithms optimize the coefficient matrix of the 2D FIR digital filter and find the best solutions for the noise elimination problem. The employed algorithms are referred to as metaheuristic methods and they are frequently used in many research areas, such as electronics, construction and biology. The basic features of the ABC, PSO, GA and DE are briefly discussed in following sections.

9.3.1 Artificial Bee Colony Algorithm

Karaboga introduced the artificial bee colony (ABC) for numerical optimization problems in 2005 [8]. The ABC algorithm is a new swarm intelligence algorithm based on the foraging behavior of honey bees for food. In later years, Basturk and Karaboga compared the performance of ABC with that of some other well-known population-based optimization algorithms [12]. To date, a number of studies have shown the use of the ABC algorithm in solving different problems from digital filter design to civil engineering problems [9–11, 13, 14]. The ABC algorithm is very flexible and it has been found to be quite robust.

Three types of bees are considered in the algorithm: employed, onlooker and scout bees. Employed bees exploit their food sources. Onlooker bees wait in the hive and decide which food source should be exploited. Scout bees carry out random searches for exploring new food sources. In the ABC algorithm, each solution in the problem under consideration is called a food source and represented by an n-dimensional real value vector.

In the ABC algorithm, we can describe procedures with an equation. Assume that W_i is the position of the i^{th} food source(i^{th} solution to the problem) and $f(W_i)$ is the nectar amount (the quality solution). The population number of food source points $P(c) = \{W_i(c)\,|i = 1, 2, ..., s\}$ (c: cycle, s: number of food sources around the hive) is equal to the number of the solutions. We can calculate p_i, the probability value of the food source chosen by an onlooker bee, as

$$p_i = \frac{f(W_i)}{\sum\limits_{k=1}^{s} f(W_k)} \tag{9.4}$$

In Eq. 9.4, s is the number of food sources or the number of employed bees in the colony since each food source has only one employed bee. If there is a new neighbor food source which is better than the previous one in bee's memory the neighbor source is memorized and the previous one is forgotten. The standard ABC algorithm has the following neighbor production formula,

$$W_i(c+1) = W_i(c) + \varphi_i(W_i(c) - W_k(c)) \tag{9.5}$$

In Eq. 9.5, φ_i is the random generated number in the interval $[-1, +1]$ and k is a randomly produced index different from i. If a food source position is not improved through a predetermined number of trials which is called the limit; the employed bee associated with that food source becomes a scout bee and tries to find a new source. The main steps of the ABC algorithm are given as follows:

1. Initialize
2. Repeat
3. Move the employed bees onto their food source and evaluate fitness
4. Move the onlookers onto the food source and evaluate their fitness
5. Move the scouts to search for new food sources
6. Memorize the best food source found so far
7. Until (termination criteria are satisfied)

9.3.2 Particle Swarm Optimization Algorithm

In 1995, Kennedy and Eberhart proposed the particle swarm optimization (PSO) algorithm [15]. The algorithm is based on simulated social behavior of bird and fish flocks. The basic steps of PSO are below:

1. Initialize
2. Repeat
3. Compute fitness values of all particles
4. Replace the best particles in the swarm
5. Select the best particle
6. Compute the velocities of all particles
7. Memorize and update the particle positions
8. Until (termination criteria are met)

In PSO, control parameters are cognitive and social components (c_1, c_2) and inertia weight (w). They play an important role in the change of the velocity and position for particles. Each particle represents a single solution of the optimization problem in search space. Populations of particles move in the search space by tracking the current optimum particles and modifying the positions in order to find the best solution. The particles update their velocities and positions according to Eqs. 9.6 and 9.7. The velocity update is defined as in Eq. 9.6

$$v(l+1) = wv(l) + c_1 rand(0,1)(p(l) - x(l)) + c_2 rand(0,1)(g(l) - x(l)) \tag{9.6}$$

where $p(l)$ and $g(l)$ are particle best and global best solutions, respectively. l is time step and the particle is gone to a new position calculated by the velocity updated at each time step. The new position updated using the new velocity and previous position and it is calculated as in Eq. 9.7

$$x(l+1) = x(l) + v(l+1) \tag{9.7}$$

9.3.3 Genetic Algorithm

In 1975, Holland proposed the genetic algorithm (GA) which is the most popular metaheuristic algorithm in the literature [7]. The genetic algorithm is an evolutionary and population based algorithm inspired by biological evolution, which uses operators such as crossover, selection and mutation. The basic GA is presented below.

1. Initialize population
2. Repeat
3. Evaluation
4. Reproduction
5. Crossover
6. Mutation
7. Until (requirements are met)

At the start of the algorithm, the initial population consists of a collection of chromosomes that represent a set of solutions for the problem. The best solution is represented by the chromosome which achieves the minimum error function value. The reproduction operator selects the chromosomes and then the chromosomes are sent to the crossover operator. In the crossover operation, two new chromosomes are produced from two existing chromosomes in the population. The common point in the selected chromosomes is randomly chosen and their corresponding digits are exchanged. Therefore, new chromosomes which represent the new solutions are produced. The crossover operation is controlled by a crossover rate (CR). In the mutation operation, the mutation operator randomly mutates some bits of the chromosomes. The mutation rate (MR) determines the number of mutation operations. The algorithm is realized by applying evaluation, reproduction, crossover and mutation operations. At the end, the best chromosome is obtained for the solution of the problem.

9.3.4 Differential Evolution

The differential evolution (DE) algorithm was presented by Storn and Price in 1997 [26]. The DE is an evolutionary algorithm like genetic algorithms and it uses similar operators, such as crossover and mutation. The main steps of the DE algorithm are summarized as follows:

1. Initialize population
2. Evaluation
3. Repeat
4. Mutation
5. Recombination
6. Evaluation
7. Selection
8. Until (termination criteria are met)

The differential evolution algorithm is controlled by scaling factor and crossover rate. In the mutation process, each of the K parameter vectors is mutated and the search space is expanded with the mutation operation. A mutant solution vector $\hat{X}_t$ is produced by Eq. 9.8

$$\hat{X}_t = X_{r_1} + F(X_{r_3} - X_{r_2}) \tag{9.8}$$

where F is the scaling factor and solution vectors X_{r_1}, X_{r_2} and X_{r_3} are randomly selected and must satisfy the following condition as in Eq. 9.9

$$X_{r_1}, X_{r_2}, X_{r_3}\Big|_{r_1 \neq r_2 \neq r_3 \neq t} \tag{9.9}$$

where t is the index of the current solution. In the crossover operation, the mutated vector is scrambled with the parent vector to generate a trial vector by

$$y_t^v = \begin{cases} \hat{x}_t^v R_v \leq CR \\ x_t^v R_v > CR \end{cases} \tag{9.10}$$

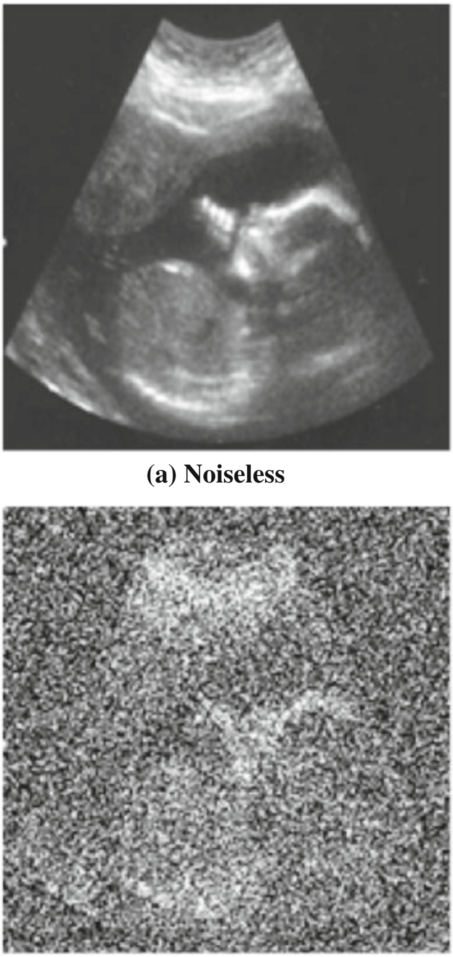
where CR is the crossover constant and R_v is a randomly chosen real number between [0,1] and v represents the v th component of the corresponding array. The performances of the trial vector and its parent are compared and the best one is chosen. If the parent is better, it is kept in the population.

9.4 Results and Discussion

In this section, firstly, a fetal ultrasound image was used as the training image in the proposed noise elimination approach. It has 256 × 256 pixels and was taken by Samsung Medison equipment. Figure 9.2 shows the noiseless and noisy ultrasound training image. The noiseless fetal ultrasound image was corrupted by Gaussian noise with zero mean and 0.75 variance.

In the proposed noise elimination approach, the ABC, PSO, GA and DE algorithms were applied as the parameter adaptation algorithm. The control parameters of the applied parameter adaptation algorithms are presented in Table 9.1 and

Fig. 9.2 Training images



(a) Noiseless

(b) Noisy

Table 9.1 Control parameter values of the ABC, PSO, GA and DE algorithms

ABC		PSO		DE		GA	
Colony size	20	Swarm size	20	Pop. size	20	Pop. size	20
Limit	27	Cognitive and social component (c_1, c_2)	1.8	Scaling factor (F)	0.5	Crossover rate (CR)	0.8
		Inertia weight (w)	0.6	Crossover rate (CR)	0.9	Mutation rate (MR)	0.01

Table 9.2 The comparison of performances of the ABC, PSO, GA and DE algorithms

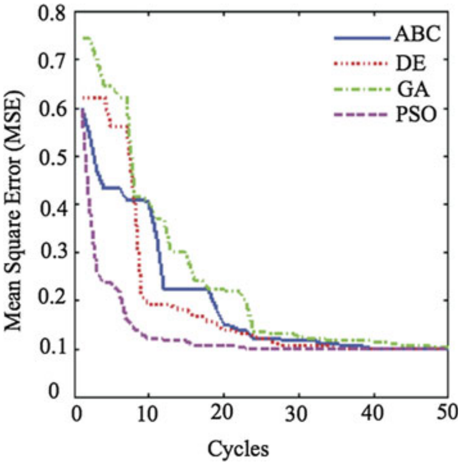
	ABC	PSO	DE	GA
Mean	0.0991	0.0992	0.0991	0.0993
Std	$1.0033e-07$	$1.1620e-04$	$2.5848e-006$	$2.7172e-04$
Best	0.0991	0.0991	0.0991	0.0991
Psnr (dB)	19.47	19.46	19.46	19.46
Time (s)	18.12	17.78	18.65	14.95

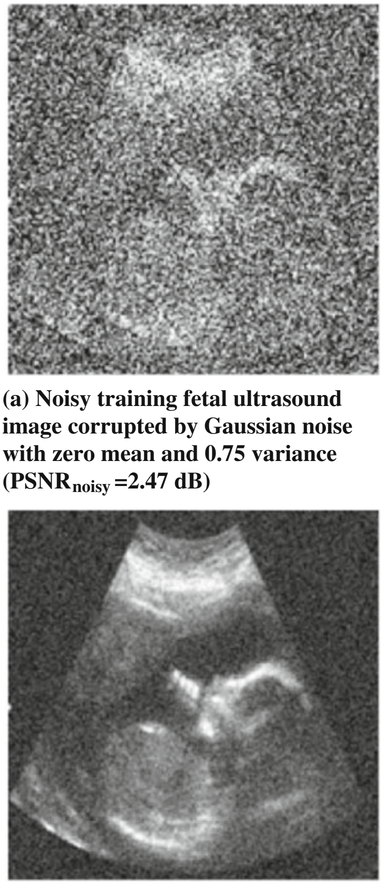
were determined experimentally. The order of the 2D FIR digital filter was set to $M = N = 3$. Each algorithm was run 30 times with different random seeds. The initial values of the parameters were selected randomly from the interval $(-1, 1)$. The colony size and the cycle number were selected as 20 and 150 for each of the parameter adaptation algorithms, respectively. The simulation was carried out in a personal computer with the following specification using MATLAB: Intel Pentium Core2 Duo T7500 2.2 G CPU, 2048 MB RAM and XP OS.

In Table 9.2, the values of the performance indicator parameters of the ABC, PSO, GA and DE algorithms are tabulated. We see that the mean values obtained by all algorithms are very similar to each other. The ABC has the best value of standard deviation. It shows that ABC is the most robust algorithm among four algorithms. The PSNR values are almost the same for all algorithms. The execution times of all the competitor algorithms are less than twenty seconds. GA needs 4.45 s while DE does 18.65 s.

The convergence characteristics of the ABC, PSO, GA and DE algorithms are shown in Fig. 9.3. As seen from the Fig. 9.3, PSO initially shows the best performance among the algorithms. However ABC demonstrates the best performance after 40 cycles.

Fig. 9.3 Convergence characteristics of the ABC, PSO, GA and DE algorithms





(a) Noisy training fetal ultrasound image corrupted by Gaussian noise with zero mean and 0.75 variance ($\mathbf{PSNR_{noisy}}$ =2.47 dB)

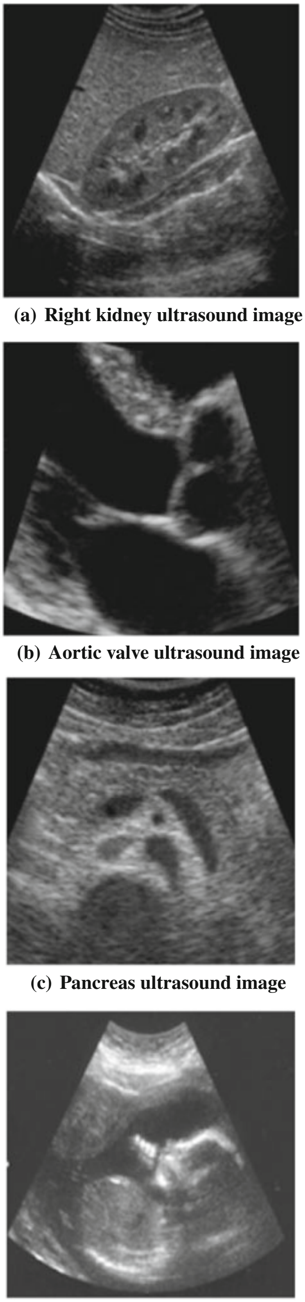
(b) Denoised fetal ultrasound image obtained by ABC algorithm ($\mathbf{PSNR_{denoised}}$ =19.47 dB)

Fig. 9.4 Fetal ultrasound images

In Fig. 9.4, the denoised fetal ultrasound image using the ABC algorithm is shown. The PSNR value obtained by the ABC algorithm is 19.47 dB.

In order to examine the proposed noise elimination process, the ultrasound images of different organs were also used, as seen in Fig. 9.5. The ultrasound images were corrupted by Gaussian noise with zero mean and $0.1 - 1$ variance. For the changing noise variance, the ABC, PSO, GA and DE algorithms were employed for noise elimination on ultrasound images.

Fig. 9.5 Ultrasound images



(a) Right kidney ultrasound image

(b) Aortic valve ultrasound image

(c) Pancreas ultrasound image

(d) Fetal ultrasound image

In Table 9.3, the noise variances versus PSNR values are tabulated for the ABC, PSO, GA and DE algorithms on different ultrasound images. From this table, it is seen that the performances of the competitor algorithms are very similar but the best values were achieved using the ABC and DE algorithms. From Table 9.3, it is clear that, when the noise variance increases, the image quality decreases.

In Fig. 9.6, the denoised ultrasound images which were obtained by the ABC algorithm are shown. The biomedical images were corrupted by Gaussian noise with zero mean and unit variance. In the proposed noise elimination approach, the ABC algorithm was used and the PNSR values of the denoised right kidney, aortic valve, pancreas and fetal ultrasound images are almost 17.00 dB.

Table 9.3 Comparison of PSNR values produced by the ABC, PSO, GA and DE algorithms

Image	Parameter adaptation algorithm	Noise variance					
		0.1	0.3	0.5	0.7	0.9	1
Fetal	ABC	36.83	27.43	23.01	20.09	17.90	16.99
	PSO	36.76	27.42	23.00	20.07	17.89	16.98
	DE	36.83	27.43	23.01	20.09	17.90	16.99
	GA	36.73	27.40	22.99	20.06	17.78	16.98
Pancreas	ABC	36.98	27.45	23.02	20.10	17.91	17.00
	PSO	36.97	27.43	22.97	20.04	17.88	16.98
	DE	36.98	27.45	23.02	20.10	17.91	17.00
	GA	36.96	27.37	22.96	20.04	17.87	16.98
Aortic valve	ABC	37.00	27.46	23.02	20.10	17.92	17.00
	PSO	37.00	27.45	23.01	20.09	17.90	17.00
	DE	37.00	27.46	23.02	20.10	17.92	17.00
	GA	36.63	27.45	22.89	20.08	17.91	16.99
Right kidney	ABC	37.00	27.46	23.02	20.10	17.91	17.00
	PSO	36.96	27.43	23.01	20.03	17.90	16.92
	DE	37.00	27.46	23.02	20.10	17.91	17.00
	GA	36.89	27.42	23.01	19.97	17.88	16.82

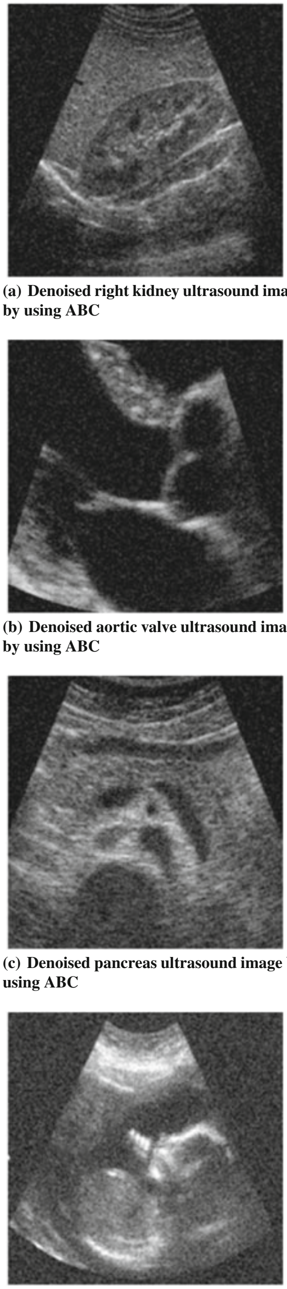


Fig. 9.6 Denoised ultrasound images obtained by ABC

(a) Denoised right kidney ultrasound image by using ABC

(b) Denoised aortic valve ultrasound image by using ABC

(c) Denoised pancreas ultrasound image by using ABC

(d) Denoised fetal ultrasound image by using ABC

9.5 Conclusion

Recently, metaheuristic techniques have been effectively used in many research areas and they have become the state of the art method for solving optimization problems. In this chapter, metaheuristic algorithms were applied to optimize the 2D FIR filter coefficient matrix for medical image denoising by using the proposed noise elimination approach. The ABC, PSO, GA and DE algorithms were the metaheuristic algorithms employed as the parameter adaptation algorithm in the proposed approach. The proposed noise elimination approach was successively used to remove Gaussian noise on noisy ultrasound images. However, this approach can be applied only if the reference noise from the detector or sensor is achieved. For this proposed approach based on metaheuristic techniques, the good result obtained is an advantage, but generating the reference noise is also a disadvantage. However, the results show that the ABC, PSO, GA and DE algorithms are very efficient and advantageous in the optimization problem. At the same time, the metaheuristic techniques were efficiently used in the design of a 2D FIR digital filter like 1D filter design.

Acknowledgements The authors are indebted to the reviewers for their constructive suggestions which significantly helped in improving the quality of this paper. This work was supported by Research Fund of Erciyes University. Project Number: FDK-2012-4156.

References

1. H.S. Bhadauria, Medical image denoising using adaptive fusion of curvelet transform and total variation. Comput. Electr. Eng. **39**, 1451–1460 (2013)
2. I. Boussaad, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics. Inf. Sci. **237**, 82–117 (2013)
3. S. Das, A. Konar, A swarm intelligence approach to the synthesis of two-dimensional IIR filters. Eng. Appl. Artif. Intell. **20**, 1086–1109 (2007)
4. D.L. Donoho, Denoising by soft thresholding. IEEE Trans. Inf. Theory **41**, 613–627 (1995)
5. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Studies in Computational Intelligence (Addison-Wesley Longman Publishing Co., Boston, 1989)
6. P. Gravel, G. Beaudoin, J.A. De Guise, A method for modeling noise in medical images. IEEE Trans. Med. Imaging **23**(10), 1221–1232 (2004)
7. J. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975)
8. D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
9. N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters. J. Franklin Inst. **346**, 328–348 (2009)
10. D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif. Intell. Rev. **42**(1), 21–57 (2012)
11. N. Karaboga, S. Kockanat, H. Dogan, The parameter extraction of the thermally annealed schottky barrier diode using the modified artificial bee colony. Appl. Intell. **38**(3), 279–288 (2013)
12. D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. **39**, 459–471 (2007)

13. N. Karaboga, F. Latifoglu, Elimination of noise on transcranial Doppler signal using IIR filters designed with artificial bee colony - ABC-algorithm. Digit. Sig. Proc. **23**(3), 1051–1058 (2013)
14. N. Karaboga, F. Latifoglu, Adaptive filtering noisy transcranial Doppler signal by using artificial bee colony algorithm. Eng. Appl. Artif. Intell. **26**(2), 677–684 (2013)
15. J. Kennedy, R. Eberhart, Particle swarm optimization, in *IEEE International Conference on Neural Networks* (1995), pp. 1942–1948
16. S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing. Science **220**, 671–680 (1983)
17. S. Kockanat, N. Karaboga, Parameter tuning of artificial bee colony algorithm for Gaussian noise elimination on digital images, In *2013 IEEE International Symposium on Innovations in Intelligent Systems and Applications* (2013), pp. 1–4
18. S. Kockanat, N. Karaboga, T. Koza, Image denoising with 2-D FIR filter by using artificial bee colony algorithm, In *2012 International Symposium on Innovations in Intelligent Systems and Applications* (2012), pp. 1–4
19. F. Latifoglu, A novel approach to speckle noise filtering based on artificial bee colony algorithm: an ultrasound image application. Comput. Methods Programs Biomed. **111**, 561–569 (2013)
20. J.S. Lee, Digital image enhancement and noise filtering by using local statistics. IEEE Trans. Pattern Anal. Mach. Intell. **2**(2), 165–168 (1980)
21. T. Loupas, W. Mc Dicken, An adaptive weighted median filter for speckle suppression in medical ultrasound images. IEEE Trans. Circuits Syst. **36**(1), 129–135 (1989)
22. W.S. Lu, A. Antoniou, *Two-Dimensional Digital Filters* (Marcel Dekker, New York, 1992)
23. N.E. Mastorakis, F. Gonos, Design of two-dimensional recursive filters using genetic algorithms. IEEE Trans. Circuits Syst. I: Fundam. Theory Appl. **50**(5), 634–639 (2003)
24. P. Perona, J. Malik, Scale-space and edge eetection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
25. H. Rabbani, R. Nezafat, S. Gazor, Wavelet-domain medical image denoising using bivariate laplacian mixture model. IEEE Trans. Biomed. Eng. **56**(12), 2826–2837 (2009)
26. R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
27. S.T. Tzeng, Design of 2-D fir digital filters with specified magnitude and group delay responses by GA approach. Sig. Proc. **87**(9), 2036–2044 (2007)
28. L. Yin, R. Yang, M. Gabbouj, Y. Neuvo, Weighted median filters: a tutorial. IEEE Trans. Circuits Syst. II, Analog Digit. Sig. Proc. **43**(3), 157–192 (1996)

Chapter 10
Medical Image Registration Based on Metaheuristics: A Comparative Study

A. Nakib, E.-G. Talbi and S. Corniglion

10.1 Introduction

Image registration is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. It is a critical step in all image analysis tasks in which the final information is gained from the combination of various data sources like in image fusion or change detection.

It geometrically aligns two images: the source and the target images. It is done by determining a transformation that maps the target image to the source one. Thus, registering a sequence of images consists in determining, for each couple of successive images, the transformation that makes the first image of the couple match the following image.

Comprehensive surveys of the registration approaches are available in the literature, we can cite [5, 12]. Registration approaches can be roughly based on:

- geometric image features (geometric registration), such as points, edges and surfaces;
- measures computed from the image grey values (intensity based registration), such as mutual information.

In many cases, a satisfactory solution can be found by using a rigid or an affine transformation (deformation model applied to the target image), i.e. the target image

A. Nakib (✉)
Laboratoire LISSI, Université Paris Est, 122 Rue P. Armangot,
94400 Vitry-sur-Seine, France
e-mail: nakib@u-pec.fr

E.-G. Talbi
Laboratoire d'Informatique Fondamentale de Lille (UMR CNRS 8022),
Villeneuve d'Ascq, France

S. Corniglion
DataTech Institute (DSTI), Paris, France

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_10

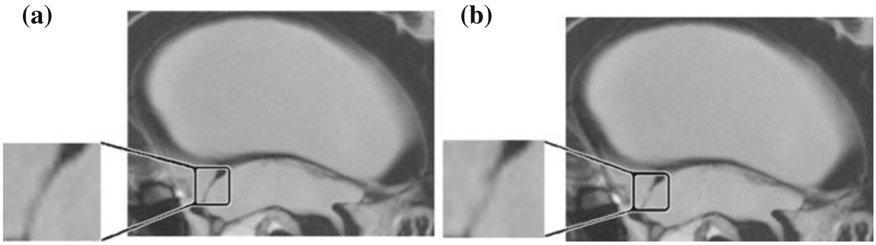


Fig. 10.1 Two images from a brain cine-MRI sequence: **a** first image of the sequence, **b** sixth image of the sequence

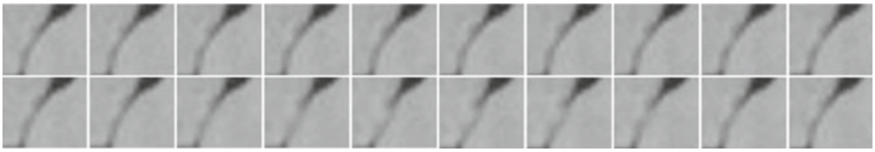


Fig. 10.2 A sequence of cine-MR images of the region of interest

is only translated, rotated and scaled to match the source image [6]. Elastic registration is required to register inter-patient images or regions containing non-rigid objects. The goal is to remove structural variation between the two images to be registered. As stated in [5], most applications represent elastic transformations in terms of a local vector displacement (disparity) field, or as polynomial transformations in terms of the old coordinates.

In the problem at hand, each image of the region of interest (i.e. lamina terminalis) is extracted from a brain cine-MRI sequence. An example of two images extracted from a brain cine-MRI sequence is presented in Fig. 10.1. Hence, each sequence is composed of 20 MR images. An example of sequence is illustrated in Fig. 10.2. The goal is to register each couple of successive images of the sequence. Hence, for a sequence of 20 images, 19 couples of successive images have to be registered. Then, the transformations that result from this matching operation can be used to assess the deformation movements of the third cerebral ventricle.

10.2 The Registration Process

A method inspired from [6] is proposed in this chapter to evaluate the movement in sequences of cine-MR images. This operation is required in order to assess the movements in the ROI over time. In [6], a segmentation process is performed on each image of the sequence, to determine the contours (as a set of points) of the walls of the third cerebral ventricle. Then, a geometric registration of each successive contours is performed, based on an affine deformation model. In the present work, we propose to use an intensity based registration instead of a geometric registration process. This way, we do not have to use a segmentation process anymore. Moreover, to evaluate the pulsatile movements of the third cerebral ventricle more precisely, an elastic deformation model is used in this chapter.

Let Im_1 and Im_2 be two successive images of the sequence. Let the transpose of a matrix A be denoted by A^{T}. Then, we assume that a transformation T_{Φ} allows to match Im_1 with $Im'_1 = T_{\Phi}(Im_2)$ and, for every pixel $o_2 = (x_2 \;\; y_2)^{\mathrm{T}}$ of Im_2, it is defined by:

$$\begin{aligned} x'_1 &= c_1\, x_2^2 \;+\; c_2\, y_2^2 \;+\; c_3\, x_2\, y_2 \;+\; (c_4\; |c_4| + 1)\, x_2 \;+\; c_5\; |c_5|\; y_2 \;+\; (c_6)^3 \\ y'_1 &= c_7\, x_2^2 \;+\; c_8\, y_2^2 \;+\; c_9\, x_2\, y_2 \;+\; c_{10}\; |c_{10}|\; x_2 \;+\; (c_{11}\; |c_{11}| + 1)\, y_2 \;+\; (c_{12})^3 \end{aligned} \tag{10.1}$$

where $o'_1 = \left(x'_1 \;\; y'_1\right)^{\mathrm{T}} = T_{\Phi}(o_2)$. The set of parameters $\Phi = \{c_1, c_2, \ldots, c_{12}\}$ is estimated through the maximization of the following criterion:

$$C(\Phi) = \frac{NMI(\Phi)}{P(\Phi) + 1} \tag{10.2}$$

where $NMI(\Phi)$ computes the normalized mutual information [11] of Im_1 and Im'_1; $P(\Phi)$ is part of a regularization term that penalizes large deformations of Im_2, as we are dealing with slight movements in the ROI. Besides, as the size of the ROI is not constant, we have to normalize the coordinates of the pixels. Then, we make the pixels in the ROI range in the interval $[-0.5, 0.5]$. The use of this interval transforms discrete coordinates of the pixels into continuous ones. This interval was determined empirically, and it is well fitted to the regularization term, and to the transformation model used. $NMI(\Phi)$ and $P(\Phi)$ are defined as follows:

$$NMI(\Phi) = \frac{H(Im_1) + H(Im'_1)}{H(Im_1,\; Im'_1)} \tag{10.3}$$

$$P(\Phi) = \max_{o_2 \,\in\, Im_1 \,\cap\, Im'_1} \left(o_2 - o'_1\right)^{\mathrm{T}} \left(o_2 - o'_1\right) \tag{10.4}$$

where $Im_1 \cap Im'_1$ is the overlapping area of Im_1 and Im'_1 (see Fig. 10.3); $H(Im_1)$ and $H(Im'_1)$ compute the Shannon entropy of Im_1 and Im'_1, respectively, in their overlapping area; $H(Im_1,\; Im'_1)$ computes the joint Shannon entropy of Im_1 and Im'_1, in their overlapping area. They are defined as follows:

$$H(Im_1) = -\sum_{i=0}^{L-1} p(i)\; \log_2\left(p(i)\right) \tag{10.5}$$

$$H(Im'_1) = -\sum_{j=0}^{L-1} p'(j)\; \log_2\left(p'(j)\right) \tag{10.6}$$

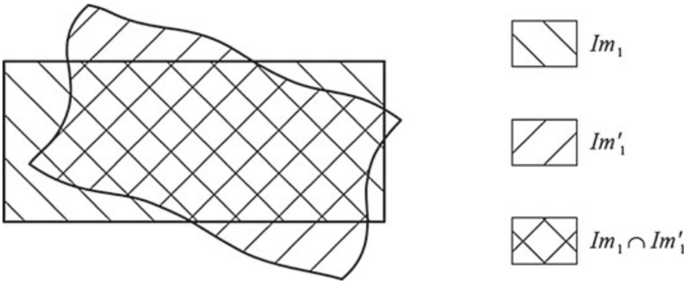


Fig. 10.3 Overlapping area ($Im_1 \cap Im'_1$) of the source image (Im_1) and the transformed target image (Im'_1) in the registration of a couple of successive images of a sequence

$$H(Im_1,\ Im'_1) = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} p(i,\ j)\ \log_2\left(p(i,\ j)\right) \tag{10.7}$$

where L is the number of possible grey values that a pixel can take; $p(i)$, $p'(j)$ and $p(i,\ j)$ are the probability of the pixel intensity i in Im_1, the probability of the pixel intensity j in Im'_1 and the joint probability of having a pixel intensity i in Im_1 and j in Im'_1, respectively. They are defined as follows:

$$p(i) = \frac{g\ (i)}{\sum_{k=0}^{L-1} g\ (k)} \tag{10.8}$$

$$p'(j) = \frac{g'\ (j)}{\sum_{l=0}^{L-1} g'\ (l)} \tag{10.9}$$

$$p(i,\ j) = \frac{g\ (i,\ j)}{\sum_{k=0}^{L-1}\sum_{l=0}^{L-1} g\ (k,\ l)} \tag{10.10}$$

where $g(i)$ is the histogram of the overlapping area of Im_1 (occurrence of gray level i in Im_1); $g'(j)$ is the histogram of the overlapping area of Im'_1 (occurrence of gray level j in Im'_1); $g(i,\ j)$ is the joint histogram of the overlapping area of Im_1 and Im'_1 (occurrence of having a grey value equal to i in Im_1 and to j in Im'_1, see Eq. (10.11)). However, in this work, we apply a low-pass filter to these histograms, using a convolution with a Gaussian function, in order to accelerate the convergence of the optimization process. Applying this filter reduces indeed the number of local optima in the objective function, by smoothing it. An illustration of the histogram of an MR image from a sequence, and of its corresponding smoothed histogram, are illustrated in Fig. 10.4.

In (10.11), the cardinal function is denoted by $card$, and the functions $Im_1(o)$ and $Im'_1(o)$ return the grey values of a given pixel o in Im_1 and Im'_1, respectively.

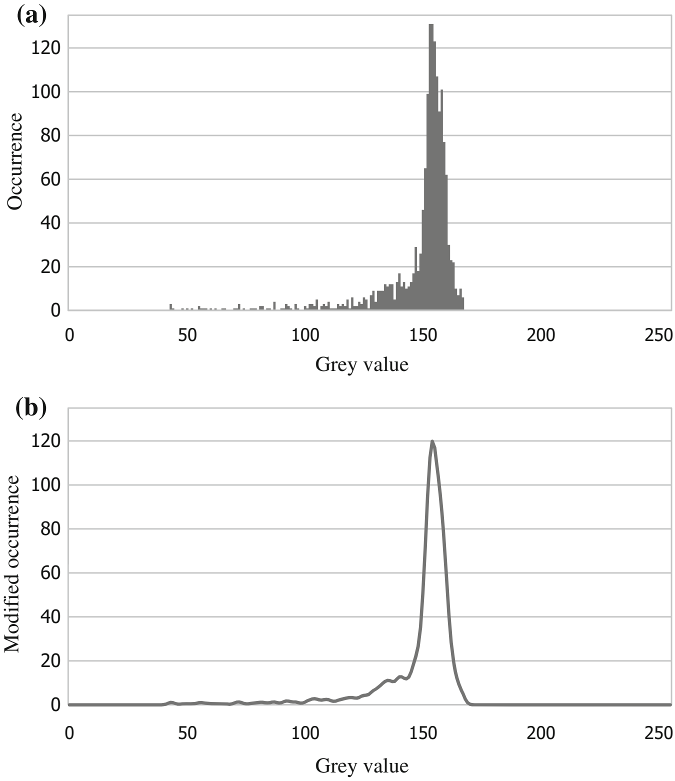


Fig. 10.4 Illustration of the histogram of an MR image: **a** original histogram, **b** smoothed histogram used to accelerate the optimization process

$$g(i, j) = \text{card}\left\{o \in Im_1 \cap Im'_1,\ \ Im_1(o) = i\ \wedge\ Im'_1(o) = j\right\} \tag{10.11}$$

The registration problem can be formulated as an optimization problem defined by:

$$\max\ C(\Phi) \tag{10.12}$$

10.3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a metaheuristic proposed in 1995 by Kennedy and Eberhart [8]. The original PSO algorithm is inspired by the social behavior of biological organisms. PSO is a population-based stochastic approach for solving continuous and discrete optimization problems. In PSO, compared to EA, an individuals is are called particles, move in the search space. The position of a particle represents its fitness. The particles fly over the search space, keeping in memory the best solution encountered. At each iteration, each particle adjusts its velocity vector, based on its momentum, influences of its best solution and of the best solution of its neighbors. Then, the new point is evaluated. Consequently, the displacement of a particle is influenced by three components:

1. Physical component: the particle tends to keep its current direction of displacement;
2. Cognitive component: the particle tends to move towards the best site that it has explored until now;
3. Social component: the particle tends to rely on the experience of the swarm, then moves towards the best site already explored by its neighbors.

In this work, the swarm size is denoted by s, and the search space is n-dimensional. In general, the particles have three attributes: the current position $X_i = (x_{i,1}, \dots, x_{i,n})$, the current velocity vector $V_i = (v_{i,1}, x_{i,2}, \dots, v_{i,n})$ and the past best position $Pbest_i = (p_{i,1}, p_{i,1}, \dots, p_{i,n})$. These attributes are used with the global best position $Gbest = (g_1, g_2, \dots, g_n)$ of the swarm, to update iteratively the state of each particle in the swarm. The objective function to be minimized is denoted by f. The new velocity vector V_i of each particle is updated as follows:

$$\begin{aligned} v_{i,j}(t+1) = v_{i,j}(t) + c_1 r_{1_{i,j}}(t) \left[Pbest_{i,j}(t) - x_{i,j}(t)\right] + \\ c_2 r_{2_{i,j}}(t) \left[gbest_j(t) - x_{i,j}(t)\right] \end{aligned} \tag{10.13}$$

vi, j is the velocity of the i^{th} particle ($i \in 1, \dots, s$) of the j^{th} dimension ($j \in 1, 2, \dots, n$) where: c_1, c_2 are the learning factors that will be fixed throughout the whole process, called acceleration coefficients, r_1 and r_2 are two random numbers in the range [0, 1] selected uniformly for each dimension at each iteration, $v_{i,j}(t)$ is the physical component, $c_1 r 1_{i,j}(t) \left[Pbest_{i,j}(t) - x_{i,j}(t)\right]$ is the cognitive component, where c_1 controls the cognitive behavior of the particle, and $c_2 r_{2_{i,j}}(t) \left[gbest_j(t) - x_{i,j}(t)\right]$ is the social component, where c_2 controls the social behavior of the particle.

The new position X_i of each particle is calculated as follows:

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \tag{10.14}$$

$x_{i,j}$ is the position of the i^{th} particle ($i \in 1, 2, \dots, s$) of the jth dimension ($j \in \{1, 2, \dots, n\}$.

In case of minimization of f, the past best position $Pbest_i$ of each particle is updated as follows:

$$Pbest_i(t+1) = \begin{cases} Pbest_i(t+1) \text{ if} f(X_i(t+1) \geq Pbest_i(t) \\ X_i(t+1) \qquad\qquad \text{Otherwise} \end{cases} \tag{10.15}$$

The global best position $Gbest$, found by the evaluations of the particles during each generation, is defined as:

$$Gbest(t+1) = \min_{Pbest_i} f(Pbest_i(t+1)), \;\; 1 \leq i \leq s \tag{10.16}$$

In the global version of PSO, the best particle $Gbest$ is chosen among the whole population. The information graph is completely connected. The information links between the particles are defined only one time, we call this topology a static information topology. The pseudo code of the original PSO is shown in Algorithm 1. The velocity clamping is problem-dependent. In the original PSO, the particle velocities vi can be clamped in the range $[?vmax, vmax]$ according to the problem at hand. This clamping is used to prevent the particles from moving out of the search space. Changing the values of c_1r_1 and c_2r_2 makes the velocity value increase, until it reaches its maximal bound [3].

Algorithm 1: Classical PSO Algorithm

```
Input: Set the swarm size $s$
Initialize randomly the position x and the velocity $v$ of each particle
set iteration = 1.
for $i = 1$ to $s$ do
    $Pbest_i == X_i$
end
while The stopping criterion is not satisfied do
    for $i = 1$ to $s$ do
        Update position $X_i$ using 10.13 and 10.14
        Evaluate the fitness $f(X_i)$
        update $Pbest_i$, $Gbest$ using 10.15 and 10.16
    end
    $iteration = iteration + 1$
end
Result: Print out $GBest$
```

10.4 Differential Evolution

Differential Evolution (DE) is an optimization algorithm based on Darwinian evolution [2], created by Storn and Price [9, 10]. Several variants to the DE algorithm have been suggested, but the original algorithm is given in Algorithm 2.

Several schemes of DE were proposed in the literature abs most of them are based on different approaches to creating each of the temporary individuals, $\mathbf{v}_i$, and different approaches to the method of creating offspring. One of two crossover schemes is typically used to create offspring. The first, binary crossover. The second common approach is called exponential crossover.

By convention, schemes are labeled in the form $DE/a/b/c$, where a is the method used to select the base vector, b is the number of difference vectors, and c is the method used to create offspring. The scheme used in Algorithm 2 is referred to as DE/rand/1/bin.

Several methods of selecting the base vector have been developed and can be used with either of the crossover methods. Popular base vector selection methods include (in each case the selected vectors are assumed to be unique):

- DE/rand/2: Two pairs of difference vectors are used:

$$\mathbf{v}_i = \mathbf{x}_1 + F \times (\mathbf{x}_2 + \mathbf{x}_3 - \mathbf{x}_4 - \mathbf{x}_5) \tag{10.17}$$

- DE/best/1: The best individual in the population is selected as the base vector:

$$\mathbf{v}_i = \mathbf{x}_{best} + F \times (\mathbf{x}_1 - \mathbf{x}_2) \tag{10.18}$$

The Differential Evolution algorithm has several control parameters that can be set. Ignoring extra parameters introduced by some DE schemes, the main DE control parameters are population size, scale factor (F), and crossover factor (C_r).

The scale factor (F) controls the magnitude of the difference vector and consequently the amount by which the base vector is perturbed. Large values of F encourage large scale exploration of the search space but could lead to premature convergence, while small values result in a more detailed exploration of the local search space while increasing convergence time. The crossover factor (C_r) controls the diversity of the population, since a large value of C_r will result in a higher probability that new genetic material will be incorporated into the population. Large values of C_r result in fast convergence while smaller values improve robustness.

Suitable values for the parameters that allow to obtain a enough good performance on a wide range of problems are known; however, for better results in terms of accuracy, robustness, and speed the parameters must be tuned for each problem.

10.5 The CMA-ES Algorithm

Evolutionary algorithms are stochastic search methods inspired by the principles of biological evolution. Evolution strategies (ES) are a class of evolutionary algorithms typically using a multi-variate normal mutation distribution. CMA-ES is a kind of the effective evolutionary optimization strategy based on the derandomized evolution strategy with covariance matrix adaptation [4]. The covariance matrix adaptation is

Algorithm 2: Classical DE Algorithm

```
Set the population size $S$
Generate a population, $P$, of $S$ individuals by creating vectors of random candidate
solutions, $\mathbf{x}i, i = 1, \ldots, S$ and $|\mathbf{x}_i| = D$
Evaluate the fitness, $f(\mathbf{x}_i)$, of all individuals
while Stoppring criteria not met do
    foreach $i \in \{1, \ldots, S\}$ do
        Select three individuals, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, at random from the current population such
        that $\mathbf{x}_1 \neq \mathbf{x}_2 \neq \mathbf{x}_3$
        Create a new trial vector $\mathbf{v}_i$ using:
        $\mathbf{v}_i = \mathbf{x}_1 + F \times (\mathbf{x}_2 - \mathbf{x}_3)$
        where $F \in [0, \infty[$ is known as the scale factor and $\mathbf{x}_1$ is referred to as the base vector
        Add $\mathbf{v}_i$ to the trial population
    end
    foreach $\mathbf{x}_i$ in the current population referred to as the target vector) do
        Create offspring $\mathbf{u}_i$ as follows:
            $u_{i,j} = \begin{cases} v_{i,j} & \text{if } (U(0,1) \leq C_r \text{or } j = id_{rand}) \\ x_{i,j} & \text{Otherwise} \end{cases}$
        where $C_r \in [0, 1]$ s the crossover probability and $id_{rand}$ is a randomly index.
        if $f(\mathbf{u}_i) < f(\mathbf{x}_i)$ then
            $\mathbf{x}_i \leftarrow \mathbf{u}_i$
        end
    end
end
```

employed, which not only reduces the randomness of evolutionary strategy, but significantly accelerates its convergence rate. In addition,the concept of cumulative evolution path is introduced into CMA-ES, which can improve the overall performance of CMA-ES. In CMA-ES, the behavior of the algorithm is defined by the following vector of parameters: $\Theta_t = (\mathbf{X}_t, \sigma_t, \mathbf{C}_t, \mathbf{p}_t, \mathbf{p}_t^{\sigma}) \in (R^n \times R^+ \times S(n, R) \times R^n \times R^n)$. Then, the evolution of the solution within the search space follows a normal distribution $N = (\mathbf{X}_t, \sigma_t \mathbf{C}_t)$, hence, only $\mathbf{X}_t$, σ_t and $\mathbf{C}_t$ encode the sampling distribution with $\mathbf{X}_t$ representing the average or the spatial position of the distribution and $\sigma_t \mathbf{C}_t$ its covariance matrix, respectively. $\mathbf{p}_t$ and $\mathbf{p}_t^{\sigma}$ are the state variables used to update $\mathbf{C}_t$ and $\mathbf{p}_t$, respectively. The parameter σ_t is the so-called step-size and can be thought of as a global scaling factor for the multivariate normal distribution this view is not fully accurate as $\mathbf{C}_t$ also plays on the scale since it is not normalized in the algorithm and has been introduced such that the global scaling can be adapted faster than the covariance matrix. The vectors $\mathbf{p}_t^{\sigma}$ and $\mathbf{p}_t$ are auxiliary state variables used to update the step-size and the covariance matrix.

Algorithm 3: Classical CMA-ES Algorithm

```
Input:
$D$ dimension of the problem
$\lambda$ Offspring population size $(4.0 + 3.0 \times Log(D))$
$\mu$ or $\mathbf{X}$ Parent population for next generation $\lfloor \lambda/2 \rfloor$
$\sigma_{start}$ Initial standard deviation
$c_{cov}$ Covariance learning raten
while Stoppring criteria not met do
    Update the Covariance Matrix $C^{(t+1)}$ using covariance matrix adaptation principle [4]
    Update the Step Size $\sigma_t$
    Generate Sample Population for generation $t + 1$
    Update the average for generation $t + 1$
    Update the best solution found
end
Output: the best solution $x^*$
```

10.6 Experimental Results and Discussion

The registrations of two couples of slightly different images are illustrated in Figs. 10.5 and 10.6, and the registrations of two couples of significantly different images are illustrated in Figs. 10.7 and 10.8. As we can see in Figs. 10.5e and 10.5f, as well as in Figs. 10.6e and 10.6f, if the movements in the ROI are not significant, then only noise appears in the difference images. Hence, the transformation used to register the couple of images (Figs. 10.5d and 10.6d) does not deform the second image of the couple significantly. On the other hand, significant movements in the ROI leave an important white trail in the difference images, as illustrated in Figs. 10.7e and 10.8e. Then, a significant transformation (Fig. 10.7d and 10.8d) has to be applied in order to eliminate the white trail (see Fig. 10.7f and 10.8f).

A comparison between the results obtained by MLSDO and those obtained by several well-known static optimization algorithms is presented in this section. These algorithms, and their parameter setting, empirically fitted to the problem at hand, are defined below (see references for more details on these algorithms and their parameter fitting):

- CMA-ES (*Covariance Matrix Adaptation Evolution Strategy*) [4] using the recommended parameter setting, except for the initial step size σ, set to $\sigma = 0.5$. The population size λ of children and the number of selected individuals μ are set to $\lambda = 11$ and $\mu = 5$;
- SPSO-07 (*Standard Particle Swarm Optimization* in its 2007 version) [1] using the recommended parameter setting, except for the number S of particles ($S = 12$) and for the parameter K used to generate the particles neighborhood ($K = 8$);

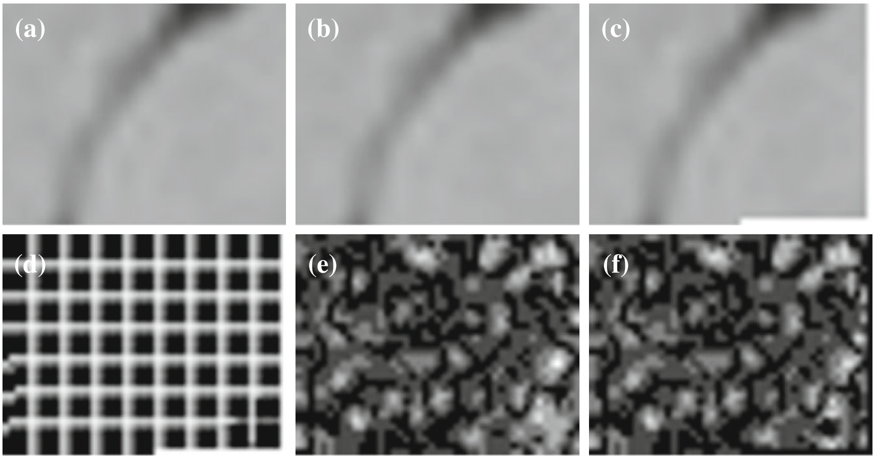


Fig. 10.5 Illustration of the registration of a couple of slightly different images of a sequence: **a** the first image of the couple, **b** the second image of the couple, **c** the second image after applying the found transformation to it, **d** illustration showing the transformation applied on the second image of the couple to register it, **e** illustration showing the difference, in the intensity of the pixels, between the two images of the couple: a *black pixel* indicates that the intensities of the corresponding pixels in the images are the same, and a *white pixel* indicates the highest difference between the images, **f** illustration showing the difference, in the intensity of the pixels, between the first image and the transformed second image

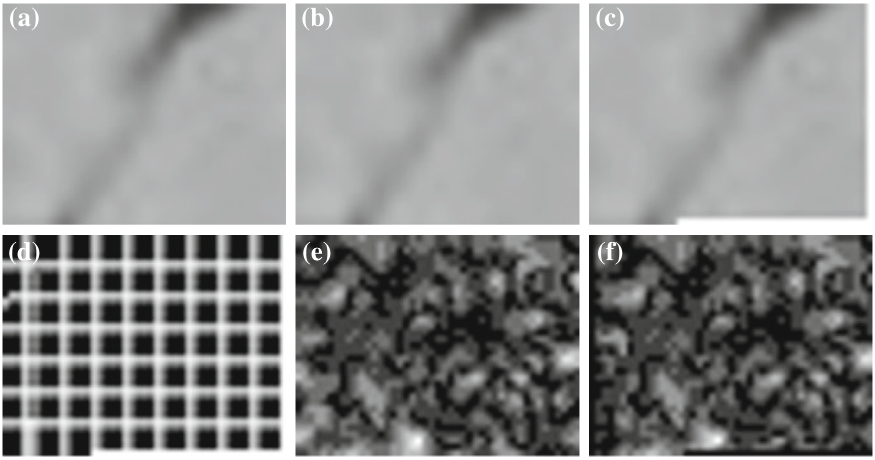


Fig. 10.6 Illustration of the registration of another couple of slightly different images of a sequence, in the same way as in Fig. 10.5

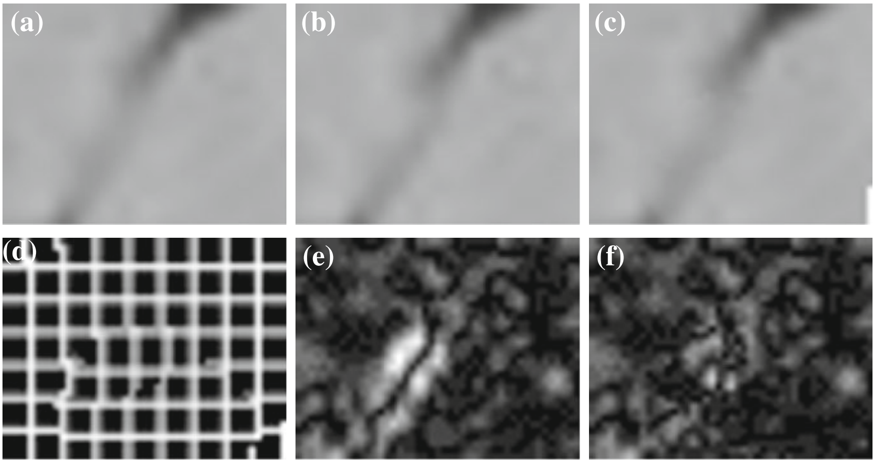


Fig. 10.7 Illustration of the registration of a couple of significantly different images of a sequence: **a** the first image of the couple, **b** the second image of the couple, **c** the second image after applying the found transformation to it, **d** illustration showing the transformation applied on the second image of the couple to register it, **e** illustration showing the difference, in the intensity of the pixels, between the two images of the couple: a *black pixel* indicates that the intensities of the corresponding pixels in the images are the same, and a *white pixel* indicates the highest difference between the images, **f** illustration showing the difference, in the intensity of the pixels, between the first image and the transformed second image

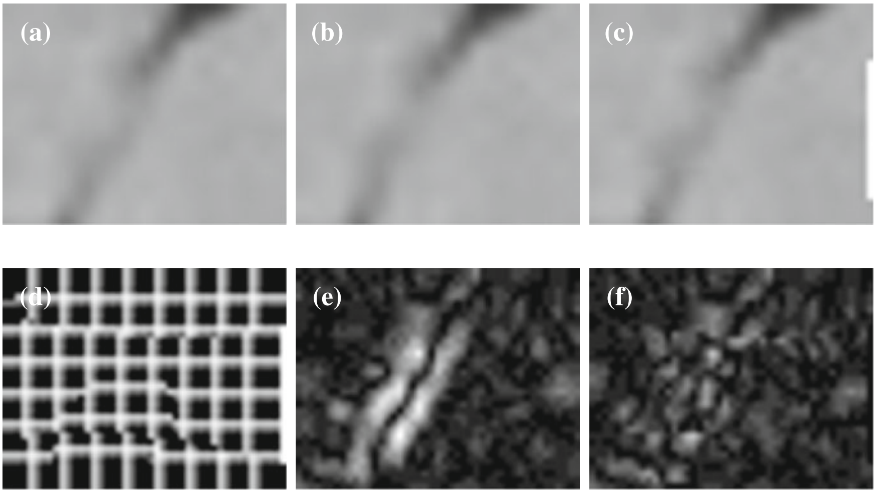


Fig. 10.8 Illustration of the registration of another couple of significantly different images of a sequence, in the same way as in Fig. 10.7

Table 10.1 Transformations found for the registration of each couple of images. The value of the objective function of the best solution found, denoted by $C^*(\Phi(t))$, is also given

t	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	C^* $(\Phi(t))$
1	0.039	−0.022	0.005	0.105	−0.034	0.139	−0.039	0.017	0.025	0.091	0.132	0.090	1.199
2	−0.005	−0.029	0.025	0.085	−0.014	0.203	0.077	0.055	0.051	0.068	−0.077	−0.264	1.201
3	0.055	0.063	0.048	0.094	−0.104	−0.239	0.068	0.000	0.000	−0.074	−0.083	−0.256	1.192
4	0.021	0.031	−0.001	0.095	−0.077	−0.223	0.025	0.013	0.006	0.081	−0.144	−0.246	1.195
5	0.063	0.000	0.003	−0.074	−0.026	−0.089	−0.026	0.041	0.011	0.100	0.145	−0.128	1.218
6	0.002	−0.063	−0.033	−0.115	0.034	0.224	−0.019	−0.027	0.024	0.015	0.087	0.258	1.209
7	0.013	−0.092	0.016	0.036	0.080	0.253	−0.060	−0.045	−0.033	−0.077	0.131	0.247	1.208
8	0.003	−0.068	−0.004	−0.023	0.117	0.238	−0.069	−0.047	−0.032	−0.078	0.131	0.247	1.195
9	0.065	−0.020	−0.007	0.044	0.061	−0.046	−0.064	−0.047	−0.023	−0.081	0.131	0.251	1.201
10	0.050	−0.004	−0.017	0.072	0.056	−0.061	0.051	0.005	0.011	−0.052	0.135	−0.043	1.216
11	0.050	0.000	−0.012	−0.004	0.073	−0.053	−0.059	0.047	0.002	0.099	0.164	−0.178	1.216
12	0.060	0.011	0.003	0.080	−0.033	−0.191	−0.024	0.032	0.036	−0.068	0.108	0.048	1.225
13	0.042	0.000	0.000	0.050	−0.018	−0.060	−0.023	0.016	0.002	−0.085	−0.064	−0.218	1.232
14	0.064	−0.005	0.000	0.094	−0.021	−0.199	−0.016	0.075	0.065	−0.039	0.065	−0.210	1.232
15	0.025	−0.008	0.042	0.049	−0.072	0.172	0.037	0.029	0.000	0.104	0.107	−0.037	1.235
16	0.060	0.007	0.003	0.082	−0.026	−0.191	−0.024	0.032	0.034	−0.063	0.111	−0.049	1.216
17	0.050	−0.005	0.000	0.021	0.010	−0.071	−0.025	0.047	0.052	0.018	0.080	−0.170	1.226
18	0.052	−0.005	−0.017	0.083	0.108	−0.121	−0.018	0.042	−0.001	0.071	0.075	0.149	1.225
19	−0.006	0.056	−0.011	−0.080	0.072	−0.210	−0.025	0.076	0.033	−0.057	0.084	−0.158	1.214

- DE (*Differential Evolution*) [7] using the "DE/target-to-best/1/bin" strategy, a number of parents equal to $NP = 30$, a weighting factor $F = 0.8$, and a crossover constant $CR = 0.9$.

As these algorithms are static, we have to consider the registration of each couple of successive images as a new problem to optimize. Thus, these algorithms are restarted after the registration of each couple of images, using the stagnation criterion defined in Sect. 7.4.3. The results obtained using MLSDO, as a static optimization algorithm, are also given.

The parameters found for the elastic transformation model are given in Table 10.1. In Table 10.2, the average number of evaluations among 20 runs of the algorithms are given. The sum of the best objective function values (see Eq. (7.14)) of each registration of the sequence is also given, averaged on 20 runs of the algorithms. The convergence of MLSDO, and that of the best performing static optimization algorithm on the problem at hand, i.e. CMA-ES, are illustrated by the curves in Fig. 10.9. It shows the evolution of the relative error $\left(\frac{C^*(\Phi(t))-C(\Phi(t))}{C^*(\Phi(t))}\right)$ between the value of the objective function of the best solution found ($C^*(\Phi(t))$) and that of the current solution ($C(\Phi(t))$) for each couple of images (t). The presented curves give an idea about the convergence of the algorithms to an optimal value. It can also be seen as a stagnation metric of the algorithms. In this figure, the number of evaluations per registration of a couple of images is fixed to 5000, in order to enable the comparison of the convergence of the algorithms. For readability, a logarithmic scale is used on the ordinates.

Table 10.2 Average number of evaluations to register all couples of images, and average sum of $C^*(\Phi(t))$, obtained by each algorithm

Algorithm	Evaluations	$\sum_{t=1}^{19} C^*(\Phi(t))$
CMA-ES	7709.14 ± 467.75	1.21 ± 9.1E-4
SPSO-07	8007.21 ± 364.24	1.21 ± 8.8E-4
DE	9131.25 ± 279.20	1.21 ± 9.3E-4
MLSDO	9522.76 ± 648.87	1.21 ± 1.7E-3

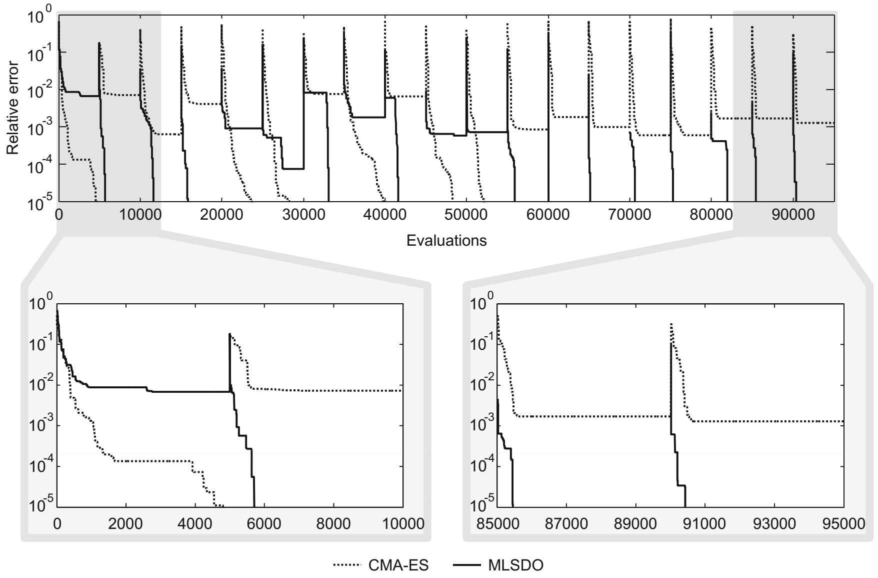


Fig. 10.9 Illustration of the convergence behaviour of CMA-ES and MLSDO

As we can see, the average sum of objective function values given in Table 10.2 shows that the algorithms have a similar average precision. However, we can see in Table 10.2 that the number of evaluations of the objective function performed by MLSDO, used as a dynamic optimization algorithm, is significantly lower than the ones of the static optimization algorithms. A Wilcoxon-Mann-Whitney statistical test has been applied on the numbers of evaluations performed by MLSDO and CMA-ES, the best ranked algorithms in terms of number of evaluations. This test confirms at a 99% confidence level that there is a significant difference between their performances. It can be seen also in Fig. 10.9 that the convergence of MLSDO to an acceptable solution is faster than CMA-ES for the registration of most of the couples of contours, especially for the last ones.

10.7 Conclusion

In this chapter, a registration process based on a dynamic optimization algorithm is proposed to register quickly all the images of a cine-MRI sequence. It takes profit from the effectiveness of the dynamic optimization paradigm. The process is sequentially applied on all the 2D images. The entire procedure is fully automated and provides an accurate assessment of the ROI deformation throughout the entire cardiac cycle. Our work under progress consists in the parallelization of the MLSDO algorithm using Graphics Processing Units.

References

1. M. Clerc et al., The Particle Swarm Central website. http://www.particleswarm.info
2. C. Darwin, *The Origin of Species* (1859)
3. R.C. Eberhart, P.K. Simpson, R.W. Dobbins, Evolutionary computation implementations, in *Computational Intelligence PC Tools* (Academic Press Professional, San Diego, 1996), pp. 212–226
4. N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies. Evolut. Comput. **9**(2), 159–195 (2001)
5. J.B.A. Maintz, M.A. Viergever, A survey of medical image registration. Med. Image Anal. **2**(1), 1–36 (1998)
6. A. Nakib, F. Aiboud, J. Hodel, P. Siarry, P. Decq, Third brain ventricle deformation analysis using fractional differentiation and evolution strategy in brain cine-MRI, in *Medical Imaging 2010: Image Processing*, vol. 7623 (SPIE, San Diego, California, USA, 2010), pp. 76232I–76232I–10
7. K. Price, R. Storn, J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization* (Springer, Berlin, 2005)
8. Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in *Proceedings of the 1999 Congress on Evolutionary Computation* (1999), pp. 1945–1950
9. R. Storn, On the usage of differential evolution for function optimization, in *Biennial Conference of the North American Fuzzy Information Processing Society* (IEEE, 1996), pp. 519–523
10. R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**, 341–359 (1997)
11. C. Studholme, D.L.G. Hill, D.J. Hawkes, An overlap invariant entropy measure of 3D medical image alignment. Pattern Recognit. **32**(1), 71–86 (1999)
12. B. Zitova, J. Flusser, Image registration methods: a survey. Image Vis. Comput. **21**(11), 977–1000 (2003)

Chapter 11
Adaptive ECG Signal Filtering Using Bayesian Based Evolutionary Algorithm

Thibaut Bernard and Amir Nakib

11.1 Introduction

Metaheuristics have been widely used to solve many different optimization problems, however, when the dimension of the problems increases the performance of theses algorithms decreases. This decrease of the performance limited the use of this approach, when the dimension is high, these problems are large scale problems. Many authors proposed several approaches to enhance the performance of the algorithms. The reader can see recent review papers as.

Indeed, find the global optimum of fitness function is very hard to find when all variables have interaction with the decision. Then, this class of problems are called non-separable problems. In large scale problems, the variable interaction increase the total number of the evaluations of the objective function to find a correct solution, this means that in this kind of problem a large number of fitness evaluations is required.

When designing a metaheuristic as an evolutionary algorithm, the choice of the operators is crucial task in order to have the best possible performance. In the literature many evolutionary operators exist, but the rule of choosing is not established. It depends on the expert. Moreover, as we said before, in the case of large-scale problems, the performance decreases after a certain number of generations. This paper, tries to solve these two problems: the choice of operators and the decrease of the performance by using Bayes theory.

In engineering, this problems have a real place and take difficult because the cost of evaluating the objective function involves interaction with other modules such a simulation software. For example obtain a good result for filter parameters is really difficult but is important especially in medical imaging where the result permit to the doctor to perform a diagnostic for the patient. An another example is the image registration who resized and modified image on a reference for localize an anomaly

T. Bernard · A. Nakib (✉)
Laboratoire LISSI, Université Paris Est, 122 Rue P. Armangot, 94400 Vitry-sur-Seine, France
e-mail: nakib@u-pec.fr

A. Nakib and E.-G. Talbi (eds.), *Metaheuristics for Medicine and Biology*,
Studies in Computational Intelligence 704, DOI 10.1007/978-3-662-54428-0_11

and take a decision for an intervention with the surgeon. Precision and rapidity are important for take a good decision faster.

A new representation of population was created in this paper, a image representation of population with color level. This representation show the evolution of the population through the generation. This method is described in the section image colour population display of this article and show the pertinence of this method.

The Purpose of this paper is to combinate bayesian theorem with a genetic algorithm for choose quickly the best strategy use for evolve the next population and take a good result in a little laps of time. We have been inspire by the work of Pelikan et al.

Bayesian theorem used mass function and probability for create a table represented the next evolution of the population. This solution permitted to used the best couple of crossover and mutation for evolve the population. This method need a lot of crossover and mutation to create a "panel" of solution and enable a quickly convergence of the algorithm. With this method we want to see if a best result was obtain using different metrics and combine this for take the mass function.

11.2 Methods

11.2.1 State of the Art

Many existing Population-based meta-heuristics, in this part of this article we display the most important algorithm.

11.2.1.1 Genetic Algorithm

The Genetic Algorithm is probably the most well-know and mostly used evolutionary computation technique. It developed in the early 1970 s at the university of Michigan by John Holland and his students. This type of algorithm is based on the evolution theory of Darwin. The representation of solution, whose named chromosomes, evolve in the course of generation. For evolve, require 3 step, one step of selection, who select chromosomes for the next steps, crossover (stage of intensification), who represent the reproduction of chromosomes and mutation (stage of diversification) who represent the evolution of chromosome.

The step of crossover take two chromosomes "parents" ad produce two offspring, if the fitness value of the offsprings are best, they replace in the population.

The step of mutation take one chromosomes and evolve it, and introduce in population for avoid to get stuck in local optima. If this step was not respected, the algorithm converge in a "bad" solution who is not the best optima.

Many criterion for interrupt the algorithm can take place: if we choose a number of generation, the algorithm prevent at the end of this generation (or number of evaluation) and return the best value and the best chromosome, if we choose a precision, algorithm don't stop before this precision was not achieve.

This algorithm require most variables:

- size of population, type int, define the initial size of population used in algorithm
- crossover rate, type double, define the rate of the crossover after the selection.
- mutation rate, type double, define the rate of the mutation after the selection.
- number of generation or stopping criterion, type int for the number of generation or number of evaluation, type double for the precision.

The motivations behind anticipating a decision: x is an array of variable to be optimized, to choose a strategy for these variables and taking a decision, we evaluate the result at t_{actual}. The problem is that the decisions may have future consequences. We have different issues and the next issues depend on previous decision $x(t)$, $t < t_{actual}$. This decision was the optimal solution and the better choice with the actual situation and data from the past. In the Fig. 11.1 we show the importance of this decision, we learn the evolution on the past and evaluate the best decision in the future.

Lot of theory are available for take a good decision, in this study, we implement most possibility of crossover and mutation and use, the Bayes theorem or Dempster Shaffer theorem for select the best method to evolve a population.

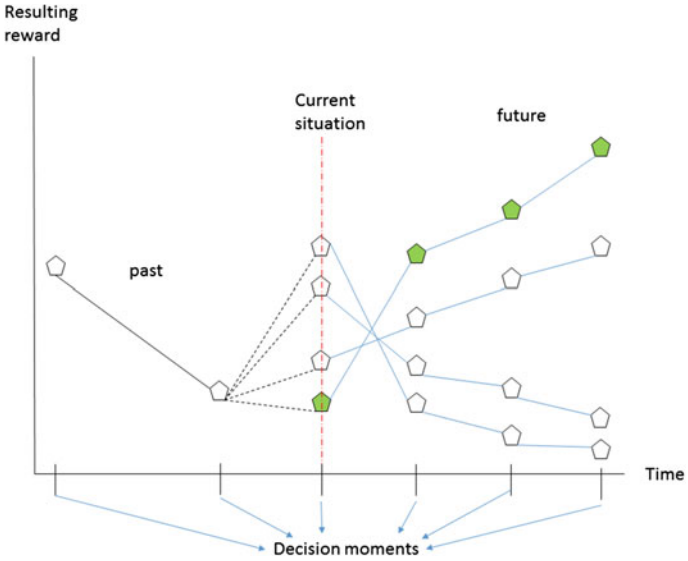


Fig. 11.1 Illustration of anticipation

11.2.2 Choice on Our Algorithm

11.2.2.1 Crossovers

We have five crossovers in this algorithm, for a best use, Bayes probability choice the best strategy of evolution (one strategy represent one crossover and one mutation)

The first crossover which used in this algorithm is the BLX-α. This crossover select two individuals and crosses this with a α parameter who define at the beginning(for this experimentation $\alpha = 0.3$) finally this crossover define for each gene the c_{max} and c_{min} and the difference I between the two genes [2, 4]. With this calcul each childs are randomly draw between the two bound superior and inferior.

Ind_1 and Ind_2 two individuals, $Child_1$ and $Child_2$ the two child whose created, $(G_{1.1}, \ldots, G_{1.N})$ the N genes of the first individual (N = dimension), $(Gc_{1.1}, \ldots, Gc_{1.N})$ the N genes of the first child, b_{inf} and b_{sup} the two bound.

For each dimension:

$$c_{max} = max(G_{1.N}, G_{2.N}) \tag{11.1}$$

$$c_{min} = min(G_{1.N}, G_{2.N}) \tag{11.2}$$

$$I = c_{max} - c_{min} \tag{11.3}$$

$$b_{inf} = c_{min} - I \times \alpha \tag{11.4}$$

$$b_{sup} = c_{max} + I \times \alpha \tag{11.5}$$

$$Gc_{1.N} = rand(b_{inf}, b_{sup}) \tag{11.6}$$

$$Gc_{2.N} = rand(b_{inf}, b_{sup}) \tag{11.7}$$

The second crossover is the discrete crossover, the method is close of the blx-α method, the difference resides in the superior bound (C_{max}) and the inferior bound is (C_{min}) [4, 11].

Ind_1 and Ind_2 two individuals, $Child_1$ and $Child_2$ the two child whose created, $(G_{1.1}, \ldots, G_{1.N})$ the N genes of the first individual (N = dimension), $(Gc_{1.1}, \ldots, Gc_{1.N})$ the N genes of the first child, b_{inf} and b_{sup} the two bound.

For each dimension:

$$b_{inf} = max(G_{1.N}, G_{2.N}) \tag{11.8}$$

$$b_{sup} = min(G_{1.N}, G_{2.N}) \tag{11.9}$$

$$Gc_{1.N} = rand(b_{inf}, b_{sup}) \tag{11.10}$$

$$Gc_{2.N} = rand(b_{inf}, b_{sup}) \tag{11.11}$$

The third crossover is the linear crossover, we generate 3 childs with different operation with the two parents and choose the two best child for the new population.this 3 childs was obtain with 3 different operation, if the two parents are identically, the operation obtain the same individual [16].

Ind_1 and Ind_2 are the two parents, $Child_1$, $Child_2$ and $Child_3$

$$Child_1 = 0.5 \times Ind_1 + 0.5 \times Ind_2 \tag{11.12}$$

$$Child_2 = 1.5 \times Ind_1 - 0.5 \times Ind_2 \tag{11.13}$$

$$Child_3 = -0.5 \times Ind_1 + 1.5 \times Ind_2 \tag{11.14}$$

The next crossover is a simple crossover, this is the one-point crossover describe in [13]. We choose one point randomly in the space $[1, N]$ and exchange the gene between the two parents before this point.

Ind_1 and Ind_2 are the two parents, $(G_{1.1}, \ldots, G_{1.N})$ the N genes of the first individual and $(G_{2.1}, \ldots, G_{2.N})$ the N genes of the second individual, and $Child_1$ and $Child_2$ the two child whose created.

select a point of crossover S in the space $[1, N]$

$$Child_1 = (G_{1.1}, \ldots, G_{1.S}, G_{2.S+1}, \ldots, G_{2.N}) \tag{11.15}$$

$$Child_2 = (G_{2.1}, \ldots, G_{2.S}, G_{1.S+1}, \ldots, G_{2.N}) \tag{11.16}$$

The last crossover is a barycentric crossover. Barycentric crossover as more adapted for continuous problem, with tw parents we build two childs with a "a" parameter which choosen randomly.

Ind_1 and Ind_2 are the two parents,"a" a randomly parameter in [0, 1], and $Child_1$ and $Child_2$ the two child whose created.

$$Child_1 = a \times Ind_1 + (1 - a) \times Ind_2 \tag{11.17}$$

$$Child_2 = (1 - a) \times Ind_1 + a \times Ind_2 \tag{11.18}$$

11.2.2.2 Mutations

Many mutations are available for genetic algorithm, in this study we used 4 mutations whose enumerate before this paragraph. The mutation are very important in your algorithm as the diversification method, the most important in hight dimension is to retain a good diversification.

The first mutation used in this study are the levy mutation [8]. For this mutation, we used a number provided to Levy distribution ($a \sim Levy(\mu, c)$). The individual evolve with this number and he is accepted in the new population.

Ind is a individual select for mutation, $G_1, \dots, G_N$ are the gene of the individual, N the dimension, “a” in number provide to Levy distribution, Mut is the mutant, $Gm_1, \dots, Gm_N$ are the genes of the individual for each dimension:

$$Gm_N = G_N + a \tag{11.19}$$

The second Mutation is the Gaussian mutation [5], this function used a number provided to Gaussian distribution ($a \sim \mathscr{N}(\mu, \sigma^2)$) and works as the Levy mutation.

Ind is a individual select for mutation, $G_1, \dots, G_N$ are the gene of the individual, N the dimension, “b” in number provide to Gaussian distribution, Mut is the mutant, $Gm_1, \dots, Gm_N$ are the genes of the individual for each dimension:

$$Gm_N = G_N + b \tag{11.20}$$

The Third Mutation used in this study, is the scramble mutation. This mutation take all genes of the individual selected for mutation and replace it in new position [7].

For example, for dimension 5:

$Ind = (G_1\ G_2\ G_3\ G_4\ G_5)$

$Ind = (G_1\ G_2\ G_3\ G_4\ G_5)$

The last mutation used is the DE/RAND/1/BIN mutation. This mutation require 3 individuals for generates one offspring. The new individual is composed by the genes of the first parent individual and a part of the difference between two other individuals. Mutation require a parameter F in the area [0, 2] who multiplied the difference [14].

Ind_1, Ind_2 and Ind_3 the three individuals selected, *Off* the offspring, $(G_{1.1}, \dots, G_{1.N})$ the genes of the first individual, $(Go_1, \dots, Go_N)$ the genes of the offspring (N the number of dimension) and F the parameter define at 0.8.

For each dimension:

$$Go_N = G_{1.N} + F \times (G_{2.N} - G_{3.N}) \tag{11.21}$$

11.2.2.3 Metrics

Diversity measures the variety of or difference between objects compared. Diversity can be measured at the gene level, chromosome level and population level and various metric can be used to measure diversity at these levels.

The first metric used in this work is the euclidean distance between all individuals. The distance represent the space who separate individuals, if the value is hight, the population have a good diversification, on the opposite, if the value is small, the population has bad diversification, this is the phase of intensification, all vectors are very close. A study of this metric learn evolution of our algorithm [10].

This is define by the equation:

$$Dist = \sum_{j=1}^{P-1} \sum_{j'=j+1}^{P} (\sum_{i=1}^{D} |G_{ij} - G_{ij'}|) \tag{11.22}$$

With G a gene, P the size if the population, D the dimension, i the rating of the gene and j and j' the rating of the chromosome.

The second metric used is the Inertia, based on extension of the concept of moment of inertia for measurement of mass distribution into arbitrarily high dimensionality spaces [10]. The coordinates of the centroid of P equally weighted points in D-space $C = (C_1, C_2, \ldots, C_D)$ and define like this:

$$C_i = \frac{\sum_{j=1}^{P} G_{ij}}{P} \tag{11.23}$$

and finally, the inertia equal to:

$$I = \sum_{i=1}^{D} \sum_{j=1}^{P} (G_{ij} - C_i)^2 \tag{11.24}$$

The third metric is the entropy. The concept of entropy, as a degree of thermodynamic equilibrium, was used firstly by Clausius in 1865 in order to interpret the irreversibility in some kinds of transformations. In the context of G, entropy represents the amount of disorder of the GP population.

$$E(P) = -\sum_{k} p_k \times log(p_k) \tag{11.25}$$

With p_k is the proportion of the population that have the fitness value in the fitness partition k^{th}.

The final metric is the convergence of the best fitness value. This metric represent the evolution of the algorithm in the course of generation of population. The value was calculated with the fitness function and save at all generation for show the algorithm convergence.

11.2.3 Bayes Theory

In probability theory and statistics, Bayes's theorem is a result of mathematical manipulation of conditional probabilities. He can be derived from basic axioms of probability. The Bayesian probability measures a degree of belief.

We consider two event A and B, and we determinate the probability of A with B known, the initial Bayes's rule is:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{11.26}$$

with P(A) and P(B) the respectively probability of A and B (a priori probabilities), P(A|B) the a posteriori probability of A with B known, she depend of B), P(B|A) the likelihood function of A.

One solution is to create at the beginning a probability data for each metric with two metrics (fusion). With this data we apply a Gaussian fit calculated with a Particle Swarm Optimization.

When the Gaussian fit was calculated, we calculate the area with a Monte-Carlo simulation. In a square we make the Gaussian fit and we generate a number of solution in this square and evaluate the number of "shot" under the Gaussian fit and divide by the total number of "shot". The result is multiplied by the area of the square and take the area of the Gaussian fit and the best result is taken for the next generation.

For example, we make a Gaussian fit in a square size 1 × 1 and we make 1000 simulation of Monte-Carlo.

$$\frac{NbrUg}{ttNbr} \times area \tag{11.27}$$

with $NbrUg$ the number of "shot" under the Gaussian, $ttNbr$ the total number of "shot" and area the area of square.

In this study, we search to know the next strategy used for evolution with the data of next population, the equation used is:

$$m(S_n|G_i) = \frac{m(S_n|G_{i-1}) \times M(S_n)}{\sum_{j=1}^{nbrS}(m(S_j|G_{n-1}) \times m(S_j))} \tag{11.28}$$

with $m(S_n)$ the initial mass function of strategy S_n, $m(S_n|G_{i-1})$ the mass of S_n at the generation G_{i-1}, $nbrS$ the number of strategies, and $m(S_n|G_i)$ the future mass of S_n for the next generation.

Finally, we select the best mass for select the strategy to use.

11.2.4 Algorithm

Algorithm 11.1: Genetic algorithm

```
step 1: initialize the population
step 2: initialize the Probability of each strategy with metric on 50 generation
while stopping criterion is not satisfied do
  step 3: choose the best strategy with Bayes
  step 4: select the chromosomes
  step 5: make crossover of strategy on selected chromosomes for crossover
  step 6: calculate fitness value for new chromosome
  step 7: insert chromosome have a better fitness
  step 8: make mutation of strategy on selected chromosomes for mutation
  step 9: calculate fitness value for new chromosome
  step 10: insert new chromosome
end while
return  the best solution
```

11.2.5 Image Color Population Representation

For see the population in hight dimension, representation with point is not available after the third dimension. In this study, we create new method for see population with a color image.

The size of image was separated in each area of the total number of genes in the population for create a 2D matrix of colors. One line represent a chromosome and one column represent a dimension (Figs. 11.2 and 11.3).

If the image is unified, all chromosomes are the same, if just a line is unified, all genes of this chromosomes are the same and if a column is unified, all genes in this dimension for all chromosomes are the same.

To show the relevance of this method, we have calculated the gradient of this image for each crossover and mutation on 100 generations and compare this with the evolution of inertia.

In this paper, we present two study in the BLX-α crossover and Levy mutation, but for a good display, we work in dimension 30 and show the different results on 8 figures.

In genetic algorithm, the crossover represent the intensification phase and the mutation represent the diversification phase. For the first results gradient and inertia approach zero and in the second study, they do not approach zero (Fig. 11.4).

For the Figs. 11.5 and 11.9, chromosomes are classify in ascending order of fitness value. The couple Figs. 11.6 and 11.7 and Figs. 11.10 and 11.11 show the same evolution on generation (Fig. 11.8).

Fig. 11.2 Flowchart of the of initialization procedure

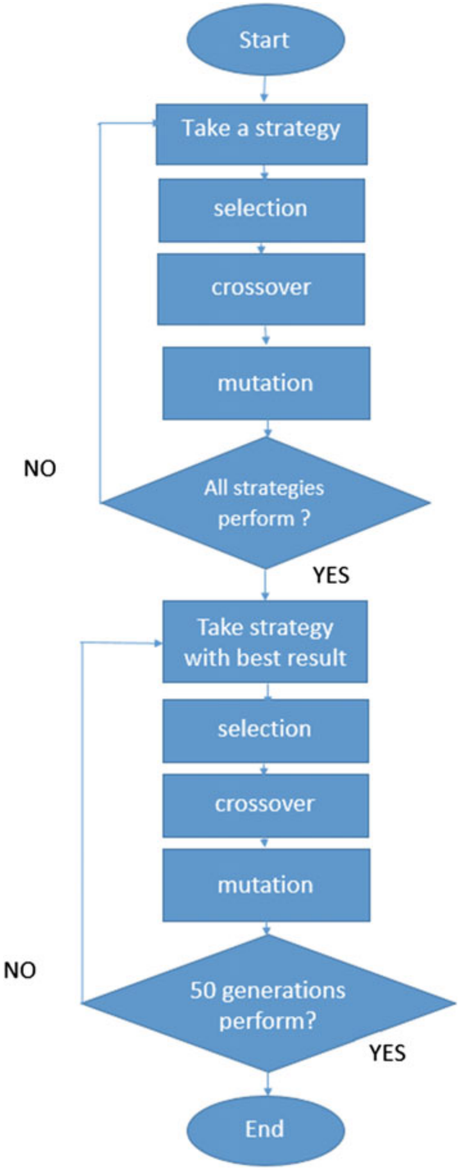
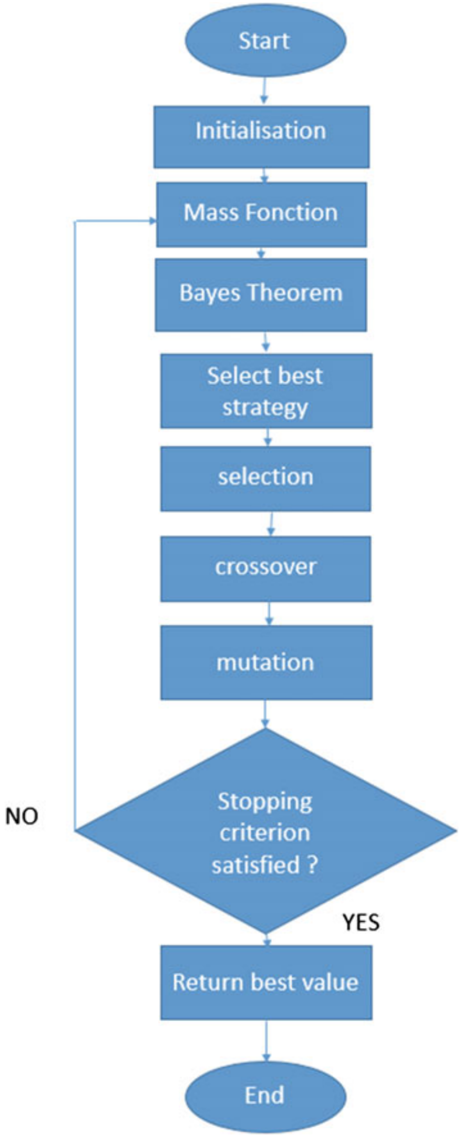


Fig. 11.3 Flowchart of BEA



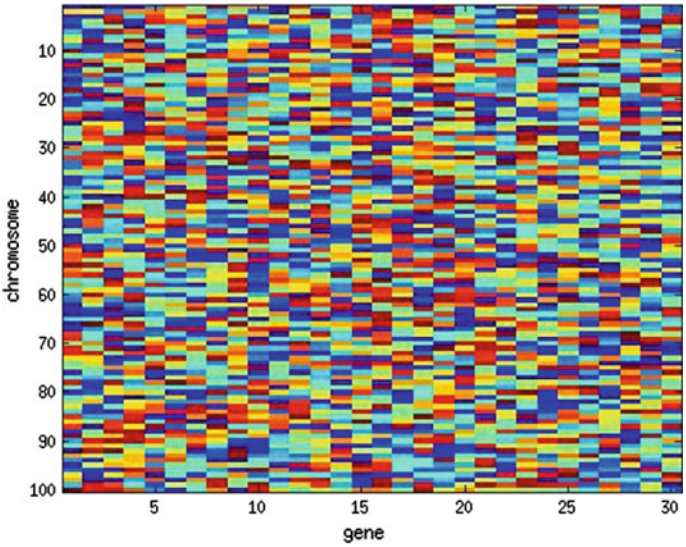


Fig. 11.4 Initial population for BLX-α crossover in dimension 30

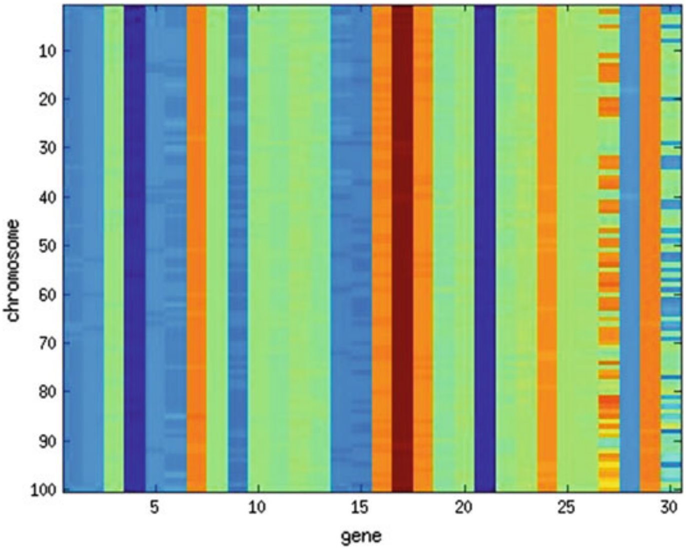


Fig. 11.5 Final population for BLX-α crossover in dimension 30

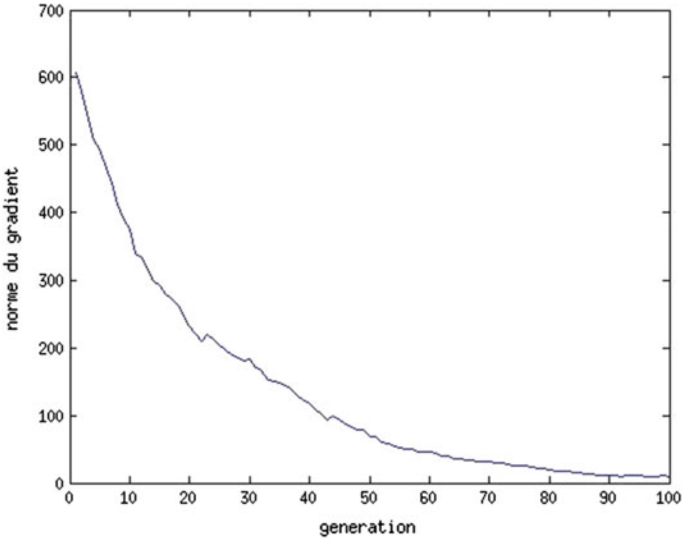


Fig. 11.6 Evolution of gradient norm on 100 generation for BLX-α crossover

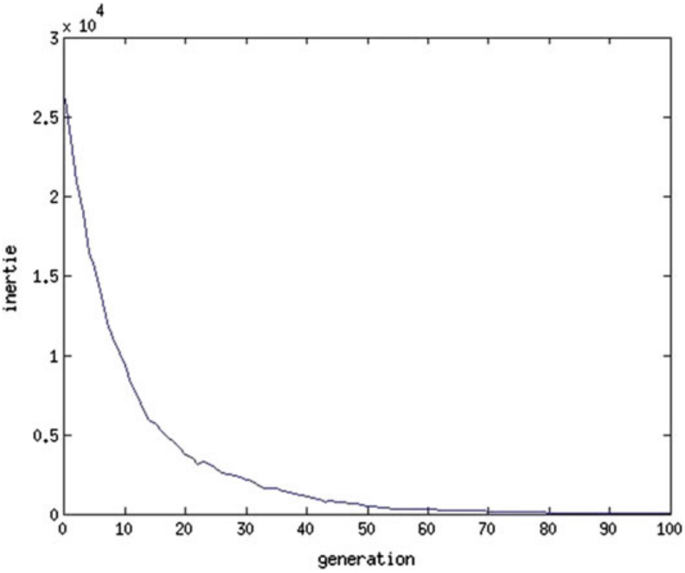


Fig. 11.7 Evolution of inertia on 100 generation for BLX-α crossover

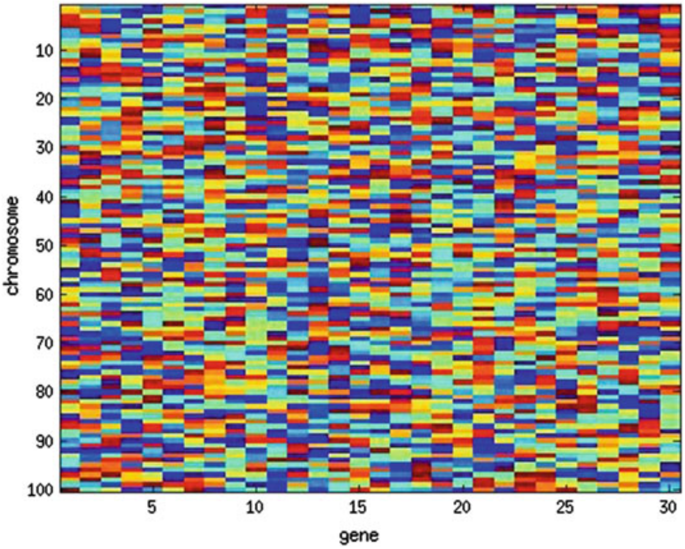


Fig. 11.8 Initial population for Levy mutation in dimension 30

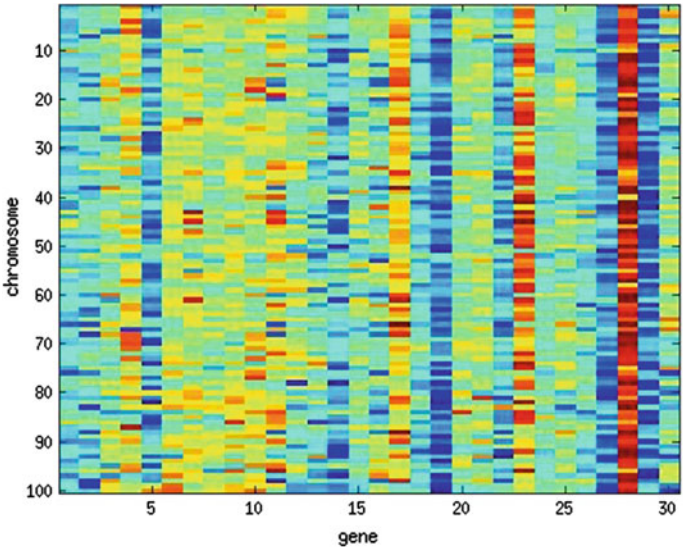


Fig. 11.9 Final population for for Levy mutation in dimension 30

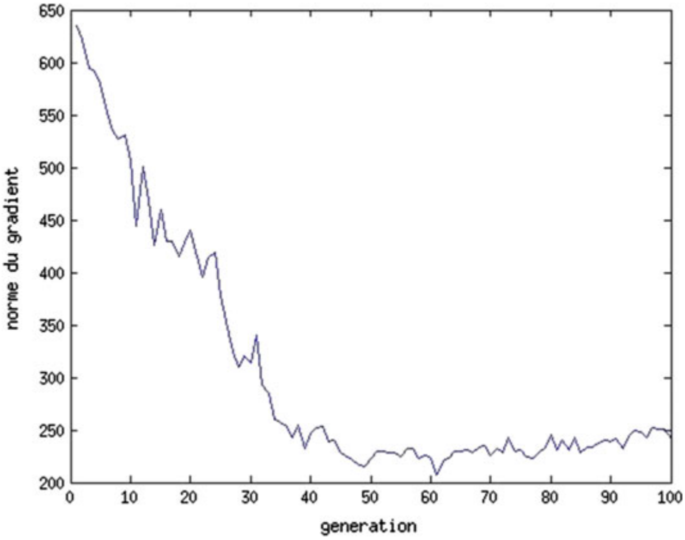


Fig. 11.10 Evolution of gradient norm on 100 generation for Levy mutation

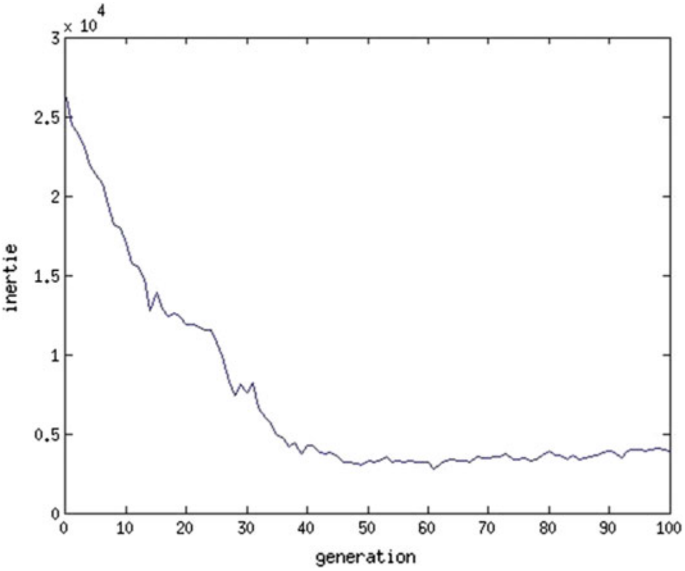


Fig. 11.11 Evolution of inertia on 100 generation for Levy mutation

11.3 Results

11.3.1 Comparison on Test Functions

The algorithms used in the comparison [15] are the “standard EA” (SEA), the self-organized criticality EA (SOCEA), the cellular EA (CEA), and the diversity-guided EA (DGEA). The SEA uses Gaussian mutation with zero means and variance $\sigma^2 = \frac{1}{\sqrt{t+1}}$. The SOCEA is a standard EA with non-fixed and non-decreasing variance $\sigma^2 = POW(10)$, where $POW(\alpha)$ is the power-law distribution. The purpose of SOC-mutation operator is to introduce many small, some mid-sized, and a few large mutations [6]. The DGEA use the Gaussian mutation operator with variance $\sigma^2 = POW(1)$, the diversity boundaries were set to $d_{low} = 5 \times 10^{-6}$ and $d_{hight} = 0.25$.

This table illustrate the average fitness of the SEA, SOCEA, the CEA, the DGEA and our algorithm on 20D problem optimized for 1000 generation, 50D for 2500 generations and 100D for 5000 generation.

problem	SEA	SOCEA	CEA	DGEA	BEA
Ackley 20D	2.49431	0.63380	0.23972	8.05E-4	2.90E-14
Ackley 50D	2.87039	1.52580	0.65169	4.61E-3	8.44E-15
Ackley 100D	2.89336	2.22080	1.14013	0.01329	2.63E-14
Griewank 20D	1.17195	0.93078	0.64280	7.02E-4	0
Griewank 50D	1.61642	1.14741	1.03284	4.40E-3	1.11E-17
Griewank 100D	2.25001	1.62948	1.17907	0.01238	2.22E-16
Rastrigin 20D	11.12678	2.87524	1.25016	2.21E-5	0
Rastrigin 50D	44.67488	22.46045	14.224	0.01664	0
Rastrigin 100D	106.21298	86.36449	58.38013	0.15665	0
Rosenbrock 20D	8392.320	406.490	149.056	96.007	13.09
Rosenbrock 50D	41425.674	4783.246	1160.078	315.395	40.18
Rosenbrock 100D	91250.3	30427.636	6053.870	1161.550	66.95

11.3.2 Interpretation

MLCC (multilevel cooperative co-evolution) [17] is a technique for large scale non-separable function optimization, in the MLCC a set of problem decomposer is constructed based on the random grouping strategy with different group sizes, the evolution process is divided into a number of cycle, and at the start of each cycle MLCC uses a self-adapted mechanism to select a decomposer according to its historical performance. DECC-G [18] is a extend of MLCC by self-adapting the subcomponent sizes. DECC-g relies on random grouping of decision variables into subcomponents in order to increase the probability of grouping interaction variables in non-separable problems. It also evolve a weight vector for co-adaptation of subcomponents for further improving the solutions. A simpler and more intuitive and yet more efficient alternative to self-adaptation of subcomponent sizes which is used in MLCC and is called DECC-ML [12]. DECC-NW [12] is a modification of DECC-G who disabled the adaptive weighting subcomponent and conducted an experiment with the same setting as DECC-G.

For each dimension, Rosenbrock (f_3) problem have a plateau, sometimes the algorithm are in the plateau and it is in the best solution, this act show the importance of diversification in this algorithm. With more diversification this algorithm finds the best solution for most of the time. This function is really difficult to optimize, we see lot of algorithm don't have a good solution for this problem. Diversification permit to have lot of point in the search space do not near each over, with this solution lot of parents create a new point with more chance to have a best result and don't have a similar result with parents. The number of evaluation are $5000 \times dimension$.

Schwefel problem 2.21 are really hard problem too, the result is the most highter chromosome of the solution. The evolution of the solution are really hard, the crossover doesn't have a good result for this function and we have no really difference between the dimension 100, 500 and 1000.

On Sphere(f_1) and Rastrigin(f_4), for all dimension the result are the best result, 0, the precision in C++ for a double is out of load, the convergence a really fast and have a very good precision for all dimension. Some algorithm like DMS-PSO or DECC-ML have 0 on this functions. They a really good result on bayes and dempster shaffer theory.

On Ackley(f_6), the result are better than the set of DECC algorithm or really near. We have a good result for this function in all dimension. All algorithm have a near result for this function and MTS take a zero. We have a good convergence for this problem. In Fig. 11.12 shows the convergence of the algorithm for Ackley problem, this convergence is really fast and take a good result faster.

On Griewank function (f_5), on dimension 100 the limit of C++ are achieving because the result are 0, but on the other dimension, result are near of th other algorithm like DECC-DML or JDEdynNP-F convergence of algorithm on Ackley function.

11.4 BEA for Adaptive Signal ECG Signal Filtering

Filter permit an amelioration of signal, when the signal was degraded by the environment, the electronic components or or the natural problem. The ECG signal (wave form) shall be composed of different signal: wave P is a deflexion corresponding of the depolarization of left and rigth auricle, wave Ta corresponding to the re-polarization of auricles but is not visible, the complex QRS corresponding to a set of deflexion due to depolarization of auricles, the wave T is a deflexion corresponding to the ventricular re-polarization Fig. 11.13.

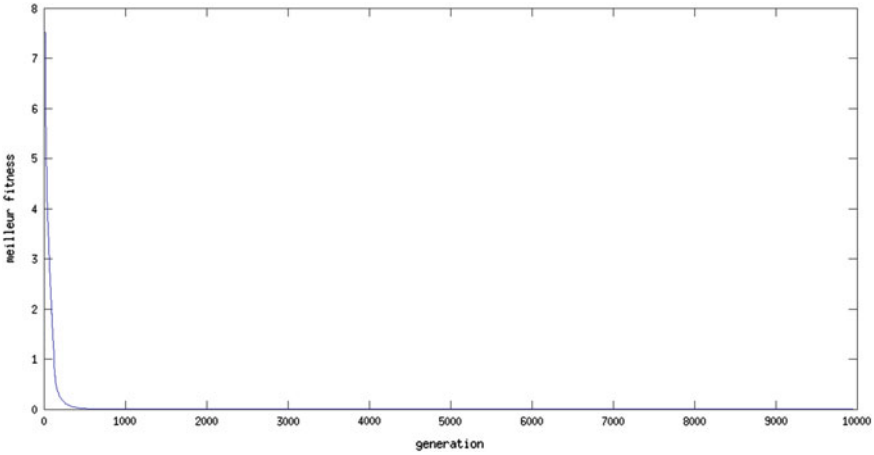


Fig. 11.12 Convergence of algorithm for Bayes method

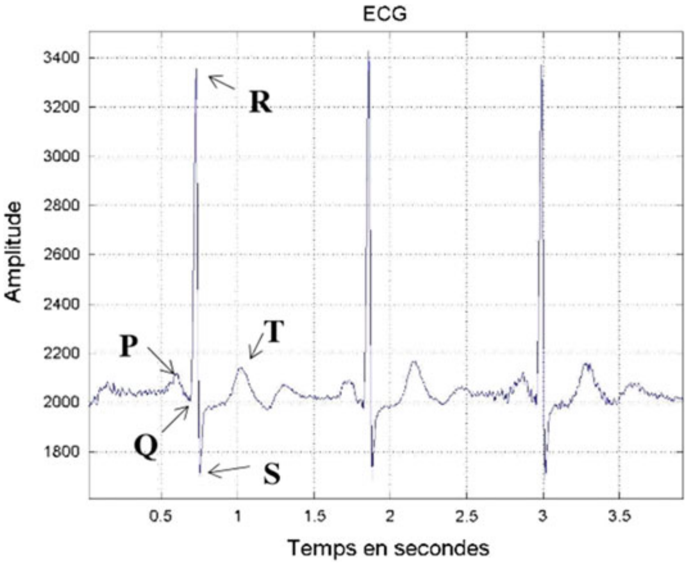


Fig. 11.13 Representation an ECG signal

The algorithm permit to calculate parameter for build analogical filter with a gabarit, we have an initial gabarit we studied the LC electronic circuit of Chebyshev on first order (Figs. 11.14 and 11.15).

Resistance value are fixed, R1 = 50 Ohms and R2 = 99.2 Ohms, the transfer function on electrical power is:

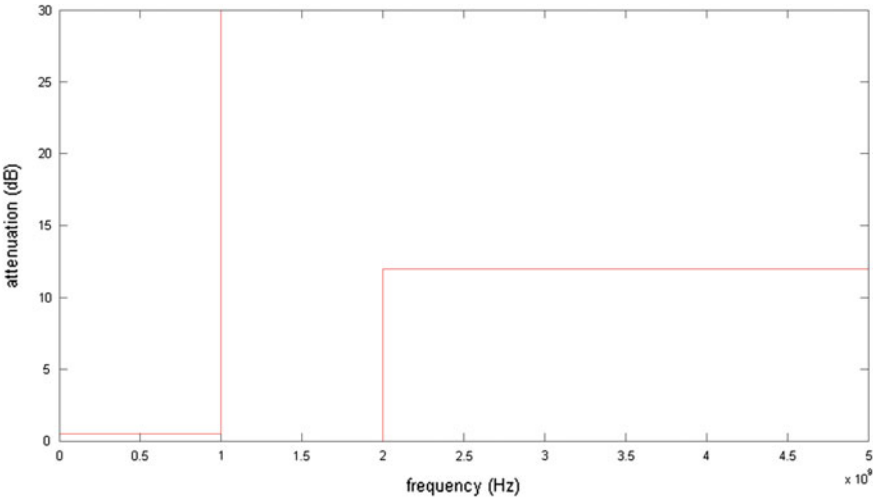
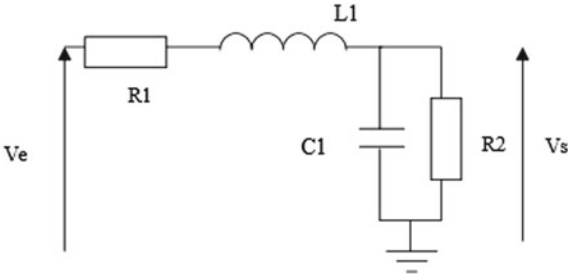


Fig. 11.14 Representation of gabarit

Fig. 11.15 Representation of electronic circuit



$$\frac{P_2}{P_m} = \frac{4R_1}{R_2} \frac{1}{\left|L_1C_1p^2 + \left(\frac{L_1}{R_2} + L_1C_1\right)p + \frac{R_1}{R_2} + 1\right|} \tag{11.29}$$

We search to optimize the module the transfer function on electrical power of LC filter with our algorithm? and we work on frequency inferior of 5 GHz.

$$A(\omega) = \frac{R_2}{4R_1}| - L_1C_1\omega^2 + (\frac{L_1}{R_2} + L_1C_1)j\omega + \frac{R_1}{R_2} + 1| \tag{11.30}$$

$$A_{dB}(\omega) = 10log(\frac{R_2}{4R_1}| - L_1C_1\omega^2 + (\frac{L_1}{R_2} + L_1C_1)j\omega + \frac{R_1}{R_2} + 1|^2) \tag{11.31}$$

example of attenuation function (Fig. 11.16).

For the optimization, we use the mean square error function and we introduce a notion of weight, so the error on each sample doesn't have the same importance, the formula is:

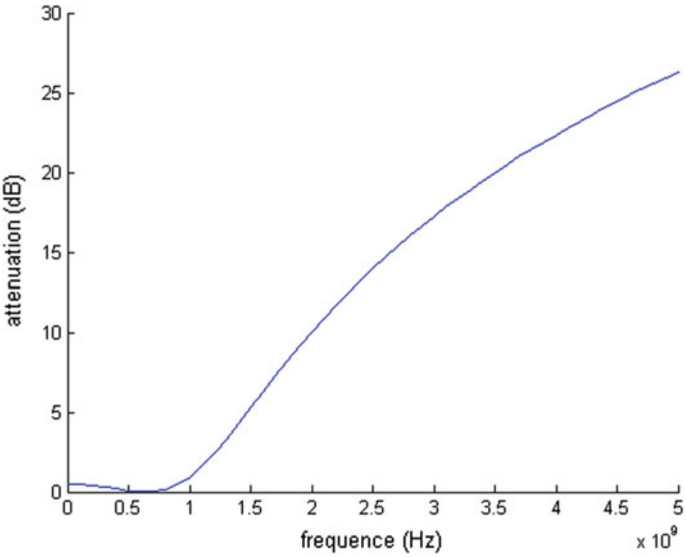


Fig. 11.16 Attenuation function with $L_1 = 14nH$ and $C_1 = 2pF$

Table 11.1 Table of result on Bayes algorithm for 100 generations

run	1	2	3	4	5	means
L_1 $(\times 10^{-9} H)$	12.2343	12.2344	12.2346	12.2335	12.2333	12.2340
C_1 $(\times 10^{-12} F)$	2.46635	2.46635	2.46633	2.46658	2.46661	2.46644

Table 11.2 Table of result on Dempster Shafer algorithm for 100 generations

run	1	2	3	4	5	means
L_1 $(\times 10^{-9} H)$	12.2338	12.2338	12.2338	12.2338	12.2338	12.2338
C_1 $(\times 10^{-12} F)$	2.4665	2.4665	2.4665	2.4665	2.4665	2.4665

$$E(L, C) = \left(\sum_{i=1}^{n} P_i (e(\omega_i))^2 \right)^{\frac{1}{2}} \tag{11.32}$$

with $e(\omega_i)$ the error of the attenuation function on ω_i, P_i the weight of error on ω_i. The number of sample in the bandwidth go to 0 at 1 GHz was less than the attenuate band go to 2 at 5 GHz, the relation increase between Pp (weight in the bandwidth) and Pa (weight in the attenuate band). For the rest of our work, we choose $Pp = 24$ and $Pa = 1$ (Tables 11.1 and 11.2).

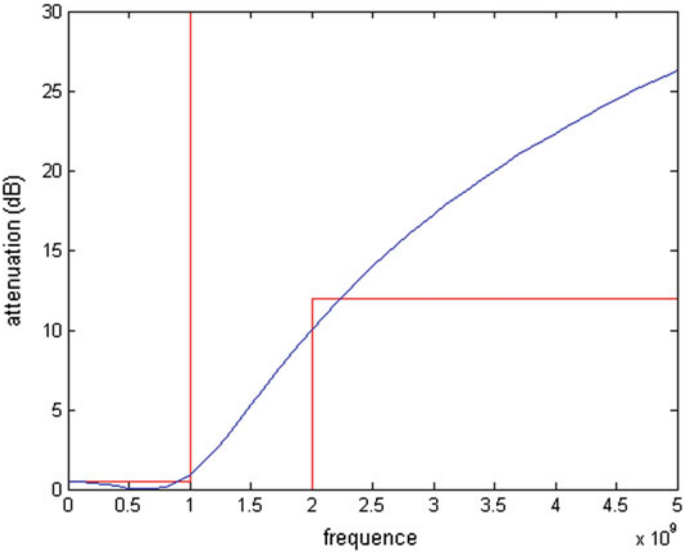


Fig. 11.17 Attenuation function with optimal results

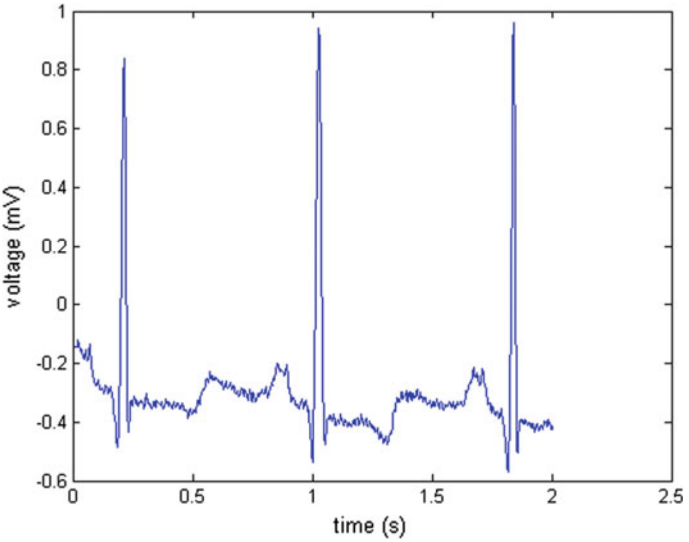


Fig. 11.18 The first ECG on the database on 2 s

With this method, we build an analogical filter and use this on the ECG provided from the MIT-BIH Arrhythmia Database [3] and we applicate a noise in hight frequency for filter this (Figs. 11.17 and 11.18).

11.5 Discussion and Conclusions

Genetic programming transfer the paradigm of evolutionary search to the space of computer program in different evolve language with help of mutation and crossover, many algorithm are established with Matlab or in c/c++ language.

We design an evolutionary algorithm for solving stochastic optimization problems and want to perform anticipation using Bayes theory with past data, change strategy of evolution permit to have a better convergence and perform the future decision.

This algorithm use floating-point representation for search space $\mathbb{R}^D$, real coded GAs do not use techniques of self-adaptation, but might time-decreasing step size [9] or change operator probabilities [1] based on their observed performance. In evolutionary algorithms often aim it finding a precise solution and converging to this solution. The choice of crossover and mutation is really important because permit to have a good diversity in population, the initial choice for building data are make with the criterion when using the metrics. In this paper we have focused on evolutionary algorithms for solving continuous optimization problem. Different solution are available for Bayes Method like use a PSO for calculated a Gaussian fit with probability and take a Monte-Carlo simulation for know the area of fitting, the biggest area are selected for choose a strategy of evolving. Many crossover and mutation exist in the literature, in this article we present many simple method but other may been implemented. The long term goal of evolution consist in the maintenance of "evolvability" of a population, guaranteed by mutation and a preservation of diversity within the population (Figs. 11.19, 11.20, 11.21, 11.22 and 11.23).

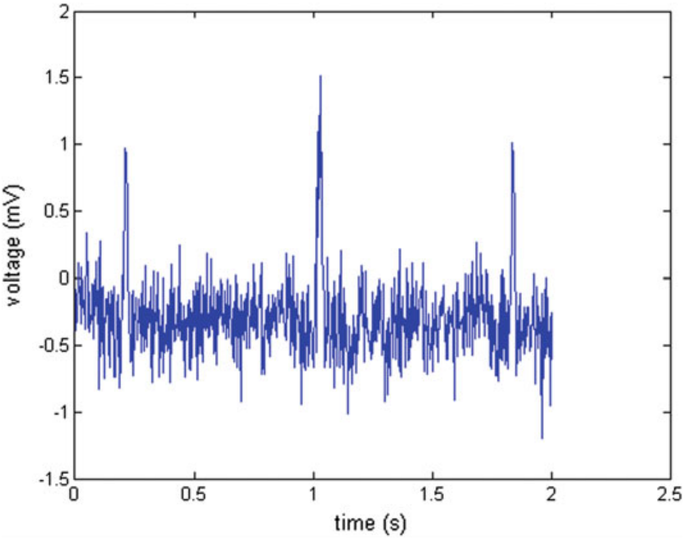


Fig. 11.19 The first ECG with noise

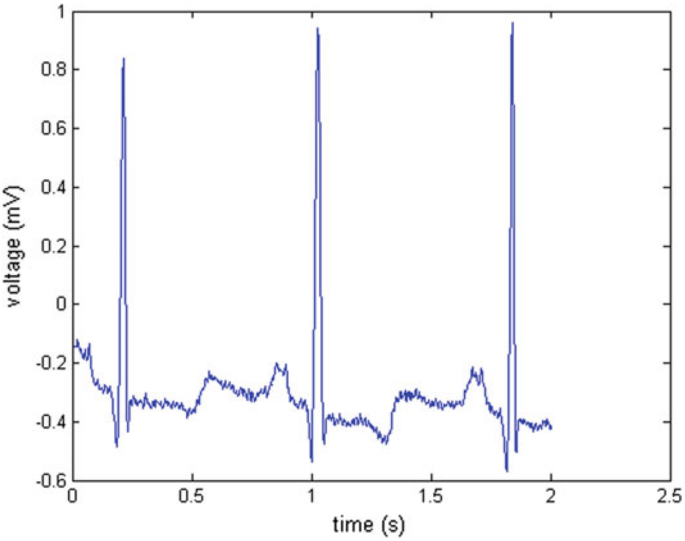


Fig. 11.20 ECG noisy after filtering

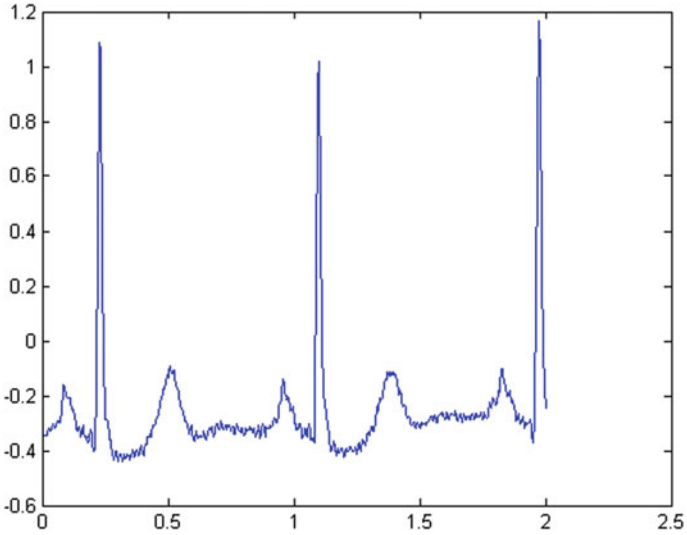


Fig. 11.21 The second ECG on the database on 2 s

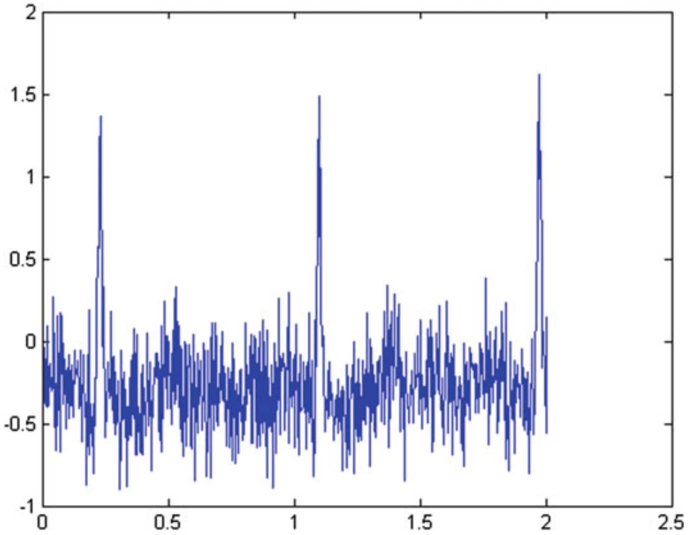


Fig. 11.22 The second ECG with noise

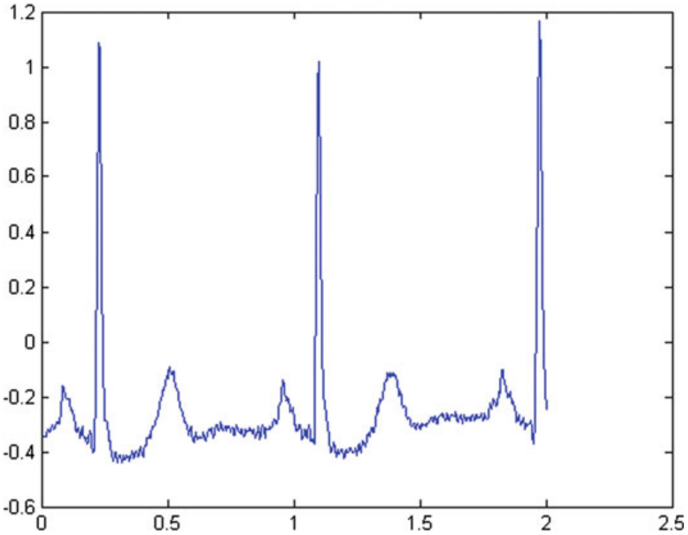


Fig. 11.23 ECG noisy 2 after filtering

Other future work includes another method of selection, other metrics for create first data, more precise or more faster and extended this method to other problem, more complex like an artificial intelligence or image Registration who require a hight precision and multi-objective function or extend to another algorithms.

References

1. L. Davis, Adapting operator probabilities in genetic algorithms, in *Proceeding of the Third International Conference on Genetic Algorithms*, ed. by J.D. Schaffer (Morgan Kaufmann Publishers, San Mateo, CA, 1989), pp. 61–69
2. L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithms and interval-schemata. *Foundation of Genetic Algorithms-2*, ed. By L. Darrell Whitley (Morgan Kaufmann Publishers, San Mateo, 1993), pp. 187–202
3. A.L. Goldberger, L. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mieutus, G.B. Moody, C.-K. Peng, H.E. Stanley, PhysioBank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. Circulation **101**(23), e215–e220 (2000) [Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215]
4. F. Herrera, M. Lozano, J.L. Verdegay, *Fuzzy Connectives Based Crossover Operators to Model Genetic Algorithms Population Diversity*. Technical report #DECSAI-95110 (1995)
5. R. Hinterding, Gaussian mutation and self-adaptation for numeric genetic algorithms. *IEEE International Conference on Evolutionary Computation*, vol. 1 (1995)
6. T. Krink, R. Thomsen, P. Rickers, Applying self-organised criticality to evolutionary programming, in *Parallel Problem Solving From Nature - PPSN VI*, vol. 1, ed. By M. Schoeunauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, H.P. Schwefel (2000), pp. 375–384
7. P. Larra*ñ*aga, C.M.H. Kuijpers, R.H. Murga, I. Inza, S. Dizdarevic, Genetic algorithms for the travelling salesman problem: a review of representations and operators. Artif. Intell. Rev. **13**, 129–170 (1999)
8. C.-Y. Lee, X. Yao, Evolutionary Programming Using Mutations Based on the Levy Probability Distribution. IEEE Trans. Evolut. Comput. **8**(1) (2004)
9. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (Springer, Berlin, 1996)
10. R.W. Morrison, K.A. De Jong, Measurement of population diversity, in *Artificial Evolution, Lecture Notes in Computer Science*, vol. 2310, pp. 31–41
11. H. M*ü*hlenbein, D. Schlierkamp-Voosen, Predictive Models for the Breeder Genetic Algorithm I. Continuons Parameter Optimization. Evolut. Comput. **1**, 25–49 (1993)
12. M. Nabi Omidvar, X. Li, Z. Yang, X. Yao, Cooperative co-evolution for large scale optimization through more frequent random grouping, *WCCI 2010, IEEE World Congress on Computational Intelligence* (2010)
13. R. Poli, W.B. Langdon, *A New Schema Theorem For Genetic Programming With One Point Crossover And Point Mutation*. Technical report: CSPR-97-3 (1997)
14. R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**, 341–359 (1997)
15. R.K. Ursem, Diversity-guided evolutionary algorithms, in *Parallel Problem Solving from Nature, PPSN VII* (Springer, Berlin, 2002)
16. A. Wright, Genetic algorithms for real parameter optimiaztion, in *Foundations of Genetic Algorithms-1*, ed, By G.J.E Rawlin (Morgan Kaufmann, San Mateo, 1991), pp. 205–218
17. Z. Yang, K. Tang, X. Yao, Multilevel cooperative coevolution for large scale optimization, in *IEEE Congress on Evolutionary Computation, CEC 2008 (IEEE World Congress on Computational Intelligence)* (2008)
18. Z. Yang, K. Tang, X. Yao, Multilevel cooperative coevolution for large scale optimization. Inf. Sci. **178**, 2986–2999 (2008)