LECTURE NOTES IN ELECTRICAL ENGINEERING 4

Xu Huang
Yuh-Shyan Chen
Sio-Iong Ao
Editors

Advances in Communication Systems and Electrical Engineering

Springer

Advances in Communication Systems
and Electrical Engineering

Lecture Notes in Electrical Engineering

Volume 4

Advances in Communication Systems and Electrical Engineering
Xu Huang, Yuh-Shyan Chen, and Sio-Iong Ao
ISBN 978-0-387-74937-2, 2008

Time-Domain Beamforming and Blind Source Separation
Julien Bourgeois, and Wolfgang Minker
ISBN 978-0-387-68835-0, 2007

Digital Noise Monitoring of Defect Origin
Telman Aliev
ISBN 978-0-387-71753-1, 2007

Multi-Carrier Spread Spectrum 2007
Simon Plass, Armin Dammann, Stefan Kaiser, and K. Fazel
ISBN 978-1-4020-6128-8, 2007

Xu Huang • Yuh-Shyan Chen • Sio-Iong Ao
Editors

Advances in Communication Systems and Electrical Engineering

Springer

Edited by

Xu Huang
School of Information Sciences
& Engineering
University of Canberra
Canberra ACT 2601
Australia

Yuh-Shyan Chen
Department of Computer Science
& Information Engineering
National Taipei University
151 University Rd.
San Shia, Taipei 237
Taiwan R.O.C.

Sio-Iong Ao
IAENG Secretariat
37-39 Hung To Road
Unit 1, 1/F
Hong Kong
People's Republic of China

ISBN: 978-0-387-74937-2 e-ISBN: 978-0-387-74938-9
DOI: 10.1007/978-0-387-74938-9

Library of Congress Control Number: 2007935439

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

Preface

A large international conference on Communication Systems and Electrical Engineering was held in Hong Kong on March 21–23, 2007, under the International MultiConference of Engineers and Computer Scientists IMECS 2007. The IMECS 2007 is organized by the International Association of Engineers (IAENG), which is a nonprofit international association for the engineers and computer scientists. The IMECS conferences serve as good platforms for the engineering community to meet with each other and exchange ideas. The conferences also struck a balance between theoretical and application development. The conference committees have been formed with over two hundred committee members who are mainly research center heads, faculty deans, department heads, professors, and research scientists from over 30 countries. The conferences are truly international meetings with a high level of participation from many countries. The response that we have received from the MultiConference is excellent. There have been more than 1,100 manuscript submissions for the IMECS 2007. All submitted papers have gone through the peer review process and the overall acceptance rate is 58.46 percent.

This volume contains revised and extended research articles written by prominent researchers participating in the IAENG International Conference on Communication Systems and Applications, the IAENG International Conference on Internet Computing and Web Services, and the IAENG International Conference on Electrical Engineering of the MultiConference IMECS 2007. Topics covered include communications theory, communications protocols, network management, wireless networks, telecommunication, electronics, power engineering, signal processing, and industrial applications. The papers are representative of those subjects that sit on the top end of high technologies. The book offers state of the art advances in communication systems and electrical engineering and also serves as an excellent reference work for researchers and graduate students working with or in the communication systems and electrical engineering fields.

Xu Huang
Yuh-Shyan Chen
Sio-Iong Ao

Contents

Chapter 1
Cooperative Transmission to Increase Gain by Employing a Precoding Scheme

Ho-Jung An, Jee-Hoon Kim, and Hyoung-Kyu Song

Multiple-input multiple-output (MIMO) systems are composed of source and destination nodes with multiple antennas. But when nodes cannot support multiple antennas due to size or other constraints, especially for mobiles, the MIMO cannot be used to provide diversity. Cooperative communication enables single antenna to realize virtual multiple antenna by sharing their antennas. Therefore, it offers transmit diversity. To achieve high performance by using Space-time block codes (STBC), we apply a precoding scheme to the transmit symbols. Even though precoding scheme has several shortcomings, it is possible to employ STBC with one transmit antenna, thereby achieving high gain without loss of transmission rate. Simulation results reveal that it is possible to obtain better error performance at the low signal-to-noise ratio (SNR) by using proposed scheme.

1.1 Introduction

The wireless channel suffers from multipath fading. The fading can cause a significant fluctuation in signal strength [1]. Therefore, repetitive transmission of signal can effectively mitigate the effect of fading by generating diversity. Spatial diversity, in particular, is generated by transmitting signals from different locations, thereby allowing reception of independently faded versions of the transmitted signal at the receiver. Spatial or multiple-antenna diversity techniques are particularly attractive because they can be easily combined with other forms of diversity—for example, time and frequency diversity. They also offer dramatic performance gains when other forms of diversity are unavailable [2]. From this point of view, MIMO systems provide a number of advantages in fading channel environments. Although MIMO systems have several benefits, they cannot be used to provide diversity when transmitters or receivers cannot support multiple antennas due to size, cost, or hardware limitations.

Ho-Jung An, Jee-Hoon Kim, and Hyoung-Kyu Song
uT Communication Research Institute, Sejong University, Seoul, Korea
e-mail: kgrfz@sdc.sejong.ac.kr,bid4u@sdc.sejong.ac.kr,songhk@sejong.ac.kr

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

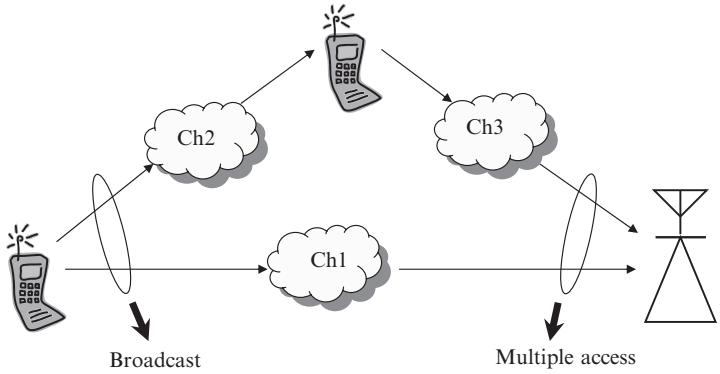


Fig. 1.1 Illustration of a cooperative communication model

As shown in Fig. 1.1, a cooperative communication system is composed of source, relay, and destination nodes. Source information is broadcast to relay and destination nodes. Information signals that were transmitted from source and relay are multiple-accessed to the destination, and then combined. Cooperative communication, therefore, enables single antenna mobiles to generate a virtual multiple antenna transmitter by exploiting relay. Thus, it can achieve transmit or spatial diversity. It is also very useful when the channel environment of direct path is inferior. Each node shares their antennas and other resources, and the source acts as a relay as well as an information source.

To apply (2,2) Alamouti's STBC [3], a mobile needs two transmit antennas. But, this paper applies a precoding scheme [4] to the transmit symbols to employ STBC with a single antenna and to achieve high gain without loss of transmission rate. A mobile with several antennas is better [5], but it is impossible in actuality. A precoding scheme is usually utilized in down-links because transmitters must know one's own transmit channel state information (CSI), and there is slight power consumption. Also, the performance is decreased if the CSI received from the receiver is not perfect. However, it can achieve more performance improvement by obtaining high gain, as well as benefit the transmission rate by transmitting symbols after precoding processing. Moreover, we employ amplify-and-forward for full diversity.

An outline of the remainder of the paper is as follows: Section 1.2 provides our system model. In Sect. 1.3, the scheme of transmitting sequences applying Alamouti's STBC for single antenna is described. Section 1.4 compares the BER performance of the proposed scheme with other methods, and Sect. 1.5 offers conclusions.

1.2 System Model

As mentioned in Sect. 1.1, transmission of symbols in applied (2,2) Alamouti's STBC requires two antennas to transmit two symbols at the same time. By using a relay, it generates the effect of two transmit antennas but transmission rate is

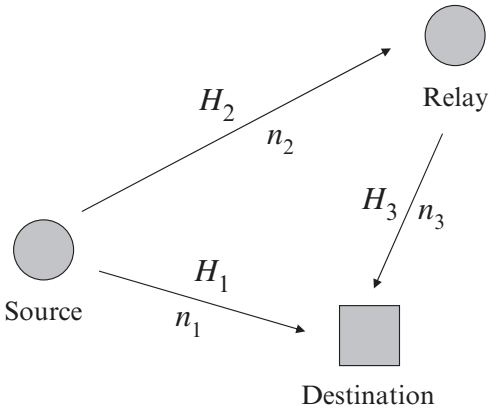


Fig. 1.2 Cooperative communication model in multipath and slow-fading channels

decreased because the source transmits one symbol each time with single antenna. So, we transmit sequences that combine two symbols.

We consider the scenario as shown in Fig. 1.2, where each source, relay, and destination nodes have single antenna. For the proposed scheme, we assume that cooperative system channels, $\mathbf{H_1}$, $\mathbf{H_2}$, and $\mathbf{H_3}$ undergo multipath and slow fading, and sequences are transmitted at narrow-band frequency. Also, the source must know the CSI between source and relay, and between source and destination. However, the source node occupies more wireless resources for instantaneous CSI of source-relay and source-destination. Time delay and error also occur during estimated CSI transmission, and bit error rate (BER) decreases due to channel estimation error. Therefore, we consider the statistical characteristics of channel instead of instantaneous CSI.

To transmit symbols c_1 and c_2, source node broadcasts sequences s_1 and s_2, which are multiplied the Alamouti's STBC form by reciprocal of channels, and are combined. And we use orthogonal frequency division multiplexing (OFDM) form. The (2,2) Alamouti's STBC utilized in this paper are

$$\begin{pmatrix} c_1 & c_2 \\ -c_2^* & c_1^* \end{pmatrix} \tag{1.1}$$

and the transmit sequences of processed precoding are as follows,

$$\begin{aligned} \mathbf{s_1} &= \mathbf{H_1^+} \cdot \mathbf{c_1} + \mathbf{H_2^+} \cdot \mathbf{c_2} \\ \mathbf{s_2} &= \mathbf{H_1^+} \cdot -\mathbf{c_2^*} + \mathbf{H_2^+} \cdot \mathbf{c_1^*} \end{aligned} \tag{1.2}$$

where $\mathbf{H_i^+}$ denotes the reciprocal of $\mathbf{H_i}$ and $(\cdot)^*$ is the complex conjugate.

Because decode-and-forward does not achieve full diversity [2], we employ amplify-and-forward to provide full diversity. Relay in this method receives a noisy

version of the sequences transmitted by source, and then amplifies and retransmits this noisy version. The destination combines the received sequences from source and relay, and makes a decision. Even though the noisy version is amplified by relay, it can make a better decision because the destination receives two independently faded versions of the sequences [1].

1.3 Expansion of Cooperative Transmission with a Single Antenna

1.3.1 Proposed Scheme

For the first sequence transmission, the received sequences are as follows,

$$\begin{aligned} \mathbf{r_{S,D}}[\mathbf{m}] &= \mathbf{H_1} \cdot \mathbf{s_1} + \mathbf{n_1} \\ &= \mathbf{H_1} \cdot (\mathbf{H_1^+} \cdot \mathbf{c_1} + \mathbf{H_2^+} \cdot \mathbf{c_2}) + \mathbf{n_1} \\ &= \mathbf{c_1} + \mathbf{H_1} \cdot \mathbf{H_2^+} \cdot \mathbf{c_2} + \mathbf{n_1} \end{aligned} \tag{1.3}$$

$$\begin{aligned} \mathbf{r_{S,R}}[\mathbf{m}] &= \mathbf{H_2} \cdot \mathbf{s_1} + \mathbf{n_2} \\ &= \mathbf{H_2} \cdot (\mathbf{H_1^+} \cdot \mathbf{c_1} + \mathbf{H_2^+} \cdot \mathbf{c_2}) + \mathbf{n_2} \\ &= \mathbf{H_2} \cdot \mathbf{H_1^+} \cdot \mathbf{c_1} + \mathbf{c_2} + \mathbf{n_2} \end{aligned} \tag{1.4}$$

$$\begin{aligned} \mathbf{r_{R,D}}[\mathbf{m}+\mathbf{N}] &= \mathbf{H_3} \cdot \beta \cdot \mathbf{r_{S,R}}[\mathbf{m}] + \mathbf{n_3} \\ &= \mathbf{H_3} \cdot \beta \cdot (\mathbf{H_2} \cdot \mathbf{H_1^+} \cdot \mathbf{c_1} + \mathbf{c_2} + \mathbf{n_2}) + \mathbf{n_3} \\ &= \beta \cdot \mathbf{H_3} \cdot \mathbf{H_2} \cdot \mathbf{H_1^+} \cdot \mathbf{c_1} + \beta \cdot \mathbf{H_3} \cdot \mathbf{c_2} + \mathbf{n_3'} \end{aligned} \tag{1.5}$$

where $\mathbf{r_{S,D}}$, $\mathbf{r_{S,R}}$, and $\mathbf{r_{R,D}}$ denote the received sequences from source to destination, from source to relay, and from relay to destination, respectively. N is also the duration of one OFDM symbol, and $m = 0, 1, ..., N-1$. The channel impulse response is assumed to be constant for the transmission duration of one OFDM frame. $\mathbf{n_1}$, $\mathbf{n_2}$, and $\mathbf{n_3}$ are zero mean complex additive white gaussian noise (AWGN) containing the effect of interference, and $\mathbf{n_3'}$ is amplified noise. β is the amplifier gain.

For the second sequence transmission, the received sequences are as follows:

$$\begin{aligned} \mathbf{r_{S,D}}[\mathbf{m}+\mathbf{2N}] &= \mathbf{H_1} \cdot \mathbf{s_2} + \mathbf{n_1} \\ &= \mathbf{H_1} \cdot (\mathbf{H_1^+} \cdot -\mathbf{c_2^*} + \mathbf{H_2^+} \cdot \mathbf{c_1^*}) + \mathbf{n_1} \\ &= -\mathbf{c_2^*} + \mathbf{H_1} \cdot \mathbf{H_2^+} \cdot \mathbf{c_1^*} + \mathbf{n_1} \end{aligned} \tag{1.6}$$

$$\begin{aligned} \mathbf{r_{S,R}}[\mathbf{m}+\mathbf{2N}] &= \mathbf{H_2} \cdot \mathbf{s_2} + \mathbf{n_2} \\ &= \mathbf{H_2} \cdot (\mathbf{H_1^+} \cdot -\mathbf{c_2^*} + \mathbf{H_2^+} \cdot \mathbf{c_1^*}) + \mathbf{n_2} \\ &= \mathbf{H_2} \cdot \mathbf{H_1^+} \cdot -\mathbf{c_2^*} + \mathbf{c_1^*} + \mathbf{n_2} \end{aligned} \tag{1.7}$$

$$\begin{aligned} \mathbf{r_{R,D}}[\mathbf{m}+\mathbf{3N}] &= \mathbf{H_3} \cdot \beta \cdot \mathbf{r_{S,R}}[\mathbf{m}+\mathbf{2N}] + \mathbf{n_3} \\ &= \mathbf{H_3} \cdot \beta \cdot (\mathbf{H_2} \cdot \mathbf{H_1^+} \cdot -\mathbf{c_2^*} + \mathbf{c_1^*} + \mathbf{n_2}) + \mathbf{n_3} \\ &= \beta \cdot \mathbf{H_3} \cdot \mathbf{H_2} \cdot \mathbf{H_1^+} \cdot -\mathbf{c_2^*} + \beta \cdot \mathbf{H_3} \cdot \mathbf{c_1^*} + \mathbf{n_3'} \end{aligned} \tag{1.8}$$

The source does not transmit sequences to the destination during a transmission period of relay, because source information is multiple-accessed by the destination node. To simplify the equations, we will use the following notations:

$$\begin{aligned} \mathbf{G} &= \mathbf{H_1} \cdot \mathbf{H_2^+} \\ \mathbf{K} &= \mathbf{H_2} \cdot \mathbf{H_1^+} \\ \mathbf{W} &= \mathbf{H_3} \cdot \mathbf{H_2} \cdot \mathbf{H_1^+} = \mathbf{H_3} \cdot \mathbf{K} \end{aligned} \tag{1.9}$$

Then, the received sequences from direct link and relay link can be separated into two estimated symbols, $\widetilde{c_1}$ and $\widetilde{c_2}$. For the direct-link,

$$\begin{aligned} \widetilde{\mathbf{c_1}} &= \mathbf{r_{S,D}}[\mathbf{m}] + \mathbf{G} \cdot \mathbf{r^*_{S,D}}[\mathbf{m}+\mathbf{2N}] \\ &= \mathbf{c_1} + \mathbf{G} \cdot \mathbf{c_2} + \mathbf{n_1} + \mathbf{G} \cdot (-\mathbf{c_2} + \mathbf{G}^* \cdot \mathbf{c_1} + \mathbf{n_1^*}) \\ &= (1 + \mathbf{G} \cdot \mathbf{G}^*) \cdot \mathbf{c_1} + \mathbf{N_1} \\ \widetilde{\mathbf{c_2}} &= \mathbf{G}^* \cdot \mathbf{r_{S,D}}[\mathbf{m}] - \mathbf{r^*_{S,D}}[\mathbf{m}+\mathbf{2N}] \\ &= \mathbf{G}^* \cdot (\mathbf{c_1} + \mathbf{G} \cdot \mathbf{c_2} + \mathbf{n_1}) - (-\mathbf{c_2} + \mathbf{G}^* \cdot \mathbf{c_1} + \mathbf{n_1^*}) \\ &= (1 + \mathbf{G}^* \cdot \mathbf{G}) \cdot \mathbf{c_2} + \mathbf{N'_1} \end{aligned} \tag{1.10}$$

Similarly, for the relay-link,

$$\begin{aligned} \widetilde{\mathbf{c_1}} &= \mathbf{W}^* \cdot \mathbf{r_{R,D}}[\mathbf{m}+\mathbf{N}] + \mathbf{H_3} \cdot \mathbf{r^*_{R,D}}[\mathbf{m}+\mathbf{3N}] \\ &= \mathbf{W}^* \cdot (\mathbf{W} \cdot \mathbf{c_1} + \mathbf{H_3} \cdot \mathbf{c_2} + \mathbf{n'_3}) + \mathbf{H_3} \cdot (\mathbf{W}^* \cdot -\mathbf{c_2} + \mathbf{H_3^*} \cdot \mathbf{c_1} + (\mathbf{n'_3})^*) \\ &= (\mathbf{W}^* \cdot \mathbf{W} + \mathbf{H_3} \cdot \mathbf{H_3^*}) \cdot \mathbf{c_1} + \mathbf{N_3} \\ \widetilde{\mathbf{c_2}} &= \mathbf{H_3^*} \cdot \mathbf{r_{R,D}}[\mathbf{m}+\mathbf{N}] - \mathbf{W} \cdot \mathbf{r^*_{R,D}}[\mathbf{m}+\mathbf{3N}] \\ &= \mathbf{H_3^*} \cdot (\mathbf{W} \cdot \mathbf{c_1} + \mathbf{H_3} \cdot \mathbf{c_2} + \mathbf{n'_3}) - \mathbf{W} \cdot (\mathbf{W}^* \cdot -\mathbf{c_2} + \mathbf{H_3^*} \cdot \mathbf{c_1} + (\mathbf{n'_3})^*) \\ &= (\mathbf{H_3^*} \cdot \mathbf{H_3} + \mathbf{W} \cdot \mathbf{W}^*) \cdot \mathbf{c_2} + \mathbf{N'_3} \end{aligned} \tag{1.11}$$

where $\mathbf{N_1}$, $\mathbf{N'_1}$, $\mathbf{N_3}$, and $\mathbf{N'_3}$ are combined noises, and we neglect the amplifier gain β to simplify the equations. Note that multiplication used in the equations is the Hadamard product because we use the OFDM form, so the order of multiplication is not important.

1.3.2 STBC

We transmit similar to the proposed scheme sequences, which combine with two symbols, but without precoding. Therefore, the transmit sequences are as follows:

$$\begin{aligned} \mathbf{s_1} &= \mathbf{c_1} + \mathbf{c_2} \\ \mathbf{s_2} &= -\mathbf{c_2^*} + \mathbf{c_1^*} \end{aligned} \tag{1.12}$$

By transmitting combined sequences, it can obtain the same gain in comparison with the one that can be obtained by transmitting one symbol each time.

1.3.3 Amplify-and-Forward

As mentioned above, the relay receives the noisy version of the sequences transmitted by source, and then amplifies and retransmits this noisy version. This scheme can be viewed as repetition coding from two separate transmitters, except that the relay transmitter amplifies its own receiver noise [2].

1.4 Simulation Results

Simulation results compare the proposed scheme with STBC form, amplify-and-forward, and direct transmission. In direct transmission, the source transmits symbols c_1 and c_2 to the destination without relay. The transmission rate is therefore increased at twice the rate in comparison with transmitting symbols employed STBC.

From the system model in Sect. 1.2, we can derive two different scenarios that the direct link between source and destination is obstructed by shadowing effects and is unshadowed. But, the transmit symbol power of the source is the same as 1 for both scenarios. In the simulation, 8-path Rayleigh fading channel and QPSK modulation are used. We also employed a maximum likelihood (ML) decision. We also assume that estimated channels are perfect. To see the difference in performance clearly, the proposed scheme and STBC form are employed in both amplifying and non-amplifying formats.

1.4.1 Scenario: Unshadowed Direct Link

In this line-of-sight (LoS) scenario, the noise power of direct link, N_0, is identical to relay link, N_0. Figure 1.3 shows the BER performance of this scenario. As shown in Fig. 1.3, the proposed method (both amplifying and non-amplifying) can bring a great performance improvement compared with STBC that does not employed precoding. It also shows high BER performance at the low SNR. In the case of the non-amplifying method, a BER of 10^{-4} is accomplished at 8 dB, while it is attained at 9 dB in case of amplifying.

1.4.2 Scenario: Shadowed Direct Link

In this scenario, we consider that direct link suffers from path loss, therefore relative noise power is larger than relay link. Figures 1.4 and 1.5 reveal the performances when path loss is 3 dB and 7 dB, respectively.

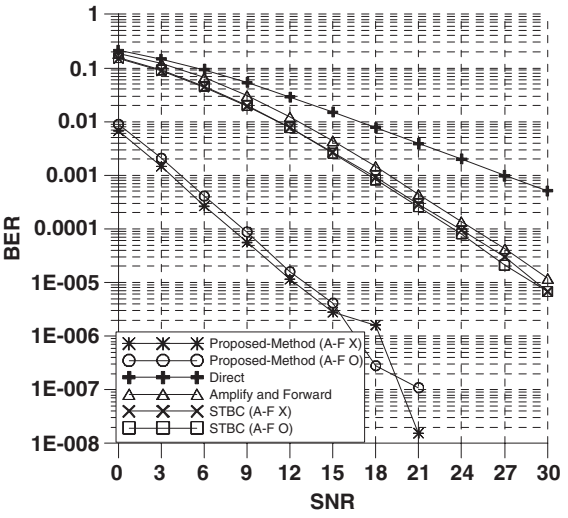


Fig. 1.3 BER performance comparison in case direct link is unshadowed

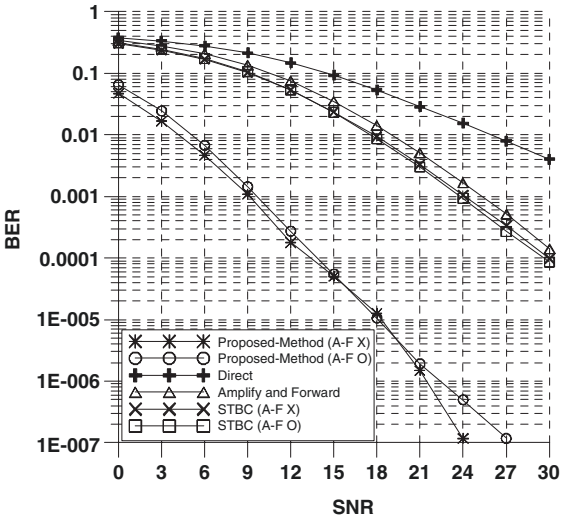


Fig. 1.4 BER performance comparison when path loss of direct link is 3 dB

As we can see in the simulation results, BER performance increases as the path loss of direct link is increased. When the path loss is 7 dB, BER of 10^{-3} is attained at 16 dB, while it is accomplished at 9 dB if the path loss is 3 dB. Nevertheless, the proposed method—both amplifying and non-amplifying—can bring great performance improvement compared with other schemes.

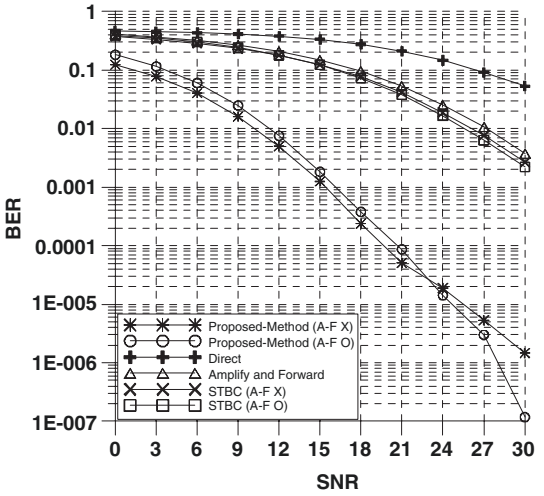


Fig. 1.5 BER performance comparison when path loss of direct link is 7 dB

1.5 Conclusions

To achieve high-performance by using (2,2) Alamouti's STBC, we presented cooperative transmission—which employs precoding scheme—in this paper the proposed method can bring high BER performance even at the low SNR. Since we utilize the statistical characteristics of a not completely known channel, the result of BER performance may be lower than actual provided simulation results. Even if we consider the reduction of performance by using the statistical characteristics of the channel, it can acquire performance benefits. Moreover, as shown in simulation results, the proposed scheme is more useful in the non-NLoS (LoS) environment.

Acknowledgements This research was supported by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC), and the 21st Century Frontier R&D Program in Korea.

References

1. Nosratinia A Hunter TE and Hedayat A (2004) Cooperative communication in wireless networks. *IEEE. Commun. Magazine*, vol. 42, no 10. Oct. 74–80
2. Laneman JN, Tse DNC and Wornell GW (2004) Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Trans. on Inform. Theory*, vol. 50, no. 12. Dec. 3062–3080
3. Alamouti SM (1998) A simple transmit diversity technique for wireless communications. *IEEE J. Select. Areas. Commun.*, vol. 16, no. 8. Oct. 1451–1458

4. Shamai S and Zaidel BM (2001) Enhancing the cellular downlink capacity via co-processing at the transmitting end. *IEEE. VTS. 53rd.*, vol. 3. May. 1745–1749
5. Miyano T, Murata H and Araki K (2004) Cooperative relaying scheme with space time code for multihop communications among single antenna terminals. *IEEE. Global. Telecommu. Conference*, vol. 6, no. 3. Dec. 3763–3767

Chapter 2
High Capacity and Reliable Techniques for Wireless Personal Area Networks

Bon-Wook Koo, Myung-Sun Baek, Jee-Hoon Kim, and Hyoung-Kyu Song

An enormous amount of multimedia data will be transmitted by various devices in the wireless personal area network, and this network environment will require very high transmission capacity. In this paper, we apply multiple antennas to MB-OFDM UWB system for high performance. With an emphasis on a preamble design for multichannel separation, we address the channel estimation in MB-OFDM system with multiple antennas. By properly designing each preamble for multiple antennas to be orthogonal in the time domain, the channel estimation can be applied to the MB-OFDM proposal for IEEE 802.15.3a standard in the case of more than two transmit antennas. By using the multiantenna scheme and proposed channel estimation technique, the reliability and performance of the MB-OFDM system can be improved.

2.1 Introduction

The rapid dissemination of digital electronic devices within the home and personal area, such as digital video disk (DVD) players, MP3 audio players, camcoders, and digital audio and television, caused increments of multimedia communication and the advent of entertainment networking. For a low-cost and high-performance wireless entertainment network to be able to support streaming multimedia content and full-motion video, ultra-wideband (UWB) technology is selected as a solution for the IEEE 802.15.3a standard [1]. UWB has attracted considerable interest in the research and standardization communities due to its promising ability to provide high data rate at a low cost with relatively low power consumption [2]. In the IEEE 802.15.3a standard, the data rate must be high enough (greater than 110 Mb/s) to satisfy a set of multimedia industry needs for wireless personal area network

Bon-Wook Koo, Myung-Sun Baek, Jee-Hoon Kim, and Hyoung-Kyu Song
uT Communication Research Institute, Sejong University, Seoul, Korea
e-mail: bebop@sdc.sejong.ac.kr, sabman@sju.ac.kr, bid4u@sdc.sejong.ac.kr, songhk@sejong.ac.kr

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

(WPAN) communication. The standard also address the quality of service (QoS) capabilities required to support multimedia data types [1]. The standard is focused on short distance wireless applications, connecting multimedia devices such as cable/satellite set-top boxes, DVDs, digital cameras, digital video recorders, TVs, displays (LCD, DLP, etc.), and PCs over distances under 10 meters. Therefore, higher rates and reliable transmission are required to satisfy the condition.

Conventionally, more bandwidth is required for higher data rate transmission. However, due to spectral limitations, it is often impractical or sometimes very expensive to increase bandwidth. In this case, the scheme of using the multiple transmit and receive antennas for spectrally efficient transmission is an alternative solution [3], and multi-input multi-output (MIMO) systems provide significant capacity gain in proportion to the number of antennas. Therefore, we apply MIMO architectures using space-time block code (STBC) to MB-OFDM UWB system. As an application of the MIMO architecture, a preamble structure for employing STBC with more than two transmit antennas is designed to be orthogonal in the time domain, and the channel estimation performance based on an investigated preamble structure is highlighted. The preamble architecture provides a feasible solution for the channel estimation without restoring channel samples corresponding to the number of substantial subcarriers used in data transmission by interpolation. The proposed preamble can be applied to the preamble of the MB-OFDM proposal for the IEEE 802.15.3a standard.

The outline of the paper is as follows: Section 2.2 describes the MB-OFDM system and the multi-antenna MB-OFDM system model. Section 2.3 describes a preamble design for multichannel estimation. The performance of the multiantenna MB-OFDM receiver based on numerical and simulation results is discussed in Sect. 2.4. Lastly, the concluding remarks are given in Sect. 2.5.

2.2 System Model

2.2.1 MB-OFDM UWB System

In the WPAN system based on MB-OFDM, the whole available UWB spectrum between 3.1-10.6GHz is divided into 14 sub-bands with 528MHz bandwidth [4]. The transmission rate of the MB-OFDM is between 53.3-480 Mbps. In data rate 53.3 and 80 Mbps, time and frequency spreading are used, and only time spreading is used between 106.7-200 Mbps. In each sub-band, a normal OFDM modulated signal with 128 subcarriers is used. For rates up to 200 Mbps, the QPSK scheme is employed in the constellation mapping, and rates over 200 Mbps use dual carrier modulation (DCM). The main difference between the MB-OFDM system and other narrowband OFDM systems is the way that different sub-bands are used in the transmission with several hopping patterns.

2.2.2 Space-Time Coded MB-OFDM UWB System

The STBC is a representative diversity transmission technique. Space-time block codes from complex orthogonal design obtain gain of transmitting diversity by transmission of the same data for multiple transmit antennas. Therefore, STBC technique brings the improvement in error performance of the communication systems.

We adopt two and four transmitting antennas and one receiving antenna. Figure 2.1 shows the block diagram of the STBC MB-OFDM system in the case of two transmitting antennas. For two transmit antennas, 2×2 code, as proposed by Alamouti, can be used as follows [5]

$$C_2 = \begin{pmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{pmatrix} \tag{2.1}$$

where s_1 and s_2 are transmitted signals, and the superscript $(\cdot)^*$ denotes the conjugate of signal. Modulated data through constellation mappers are coded by Eq. (2.1), and transmitted through two transmit antennas and time slots. The receiver performs the channel estimation and combines data to obtain diversity gain. The combined data can be obtained as follows

$$\begin{aligned} \widetilde{s}_1 &= \widehat{h}_1^* r_1 + \widehat{h}_2 r_2^* \\ \widetilde{s}_2 &= \widehat{h}_2^* r_1 - \widehat{h}_1 r_2^* \end{aligned} \tag{2.2}$$

where $\widetilde{s}_1$ and $\widetilde{s}_2$ are combined signals, and $\widehat{h}_1$ and $\widehat{h}_2$ are estimated channel state information (CSI). r_1 and r_2 are received signals at time t and time $t+T$. In the case of four transmit antennas, C_4 of 4×4 code is used as follows:

$$C_4 = \begin{pmatrix} s_1 & s_2 & s_3 & 0 \\ -s_2^* & s_1^* & 0 & -s_3 \\ -s_3^* & 0 & s_1^* & s_2 \\ 0 & s_3^* & -s_2^* & s_1 \end{pmatrix}. \tag{2.3}$$

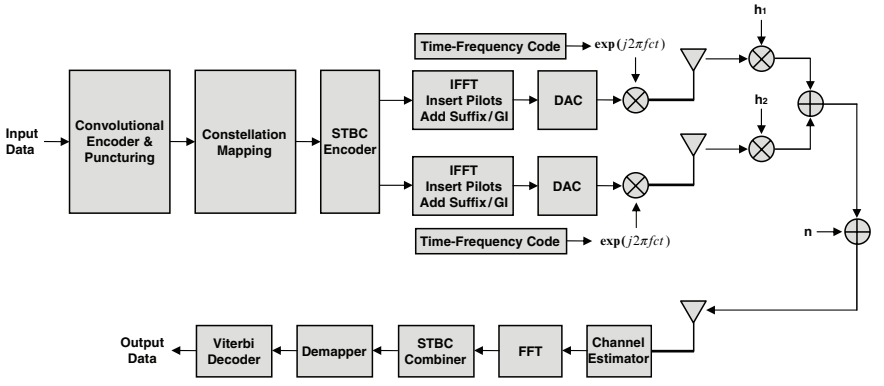


Fig. 2.1 The space-time coded MB-OFDM system with two transmitting antennas and one receiving antenna

C_4 has a code rate of 3/4, because three symbols are transmitted through four time slots. The combining process is similar to two transmit antennas, and combined signals are as follows:

$$\begin{aligned}\widetilde{s_1} &= \widehat{h}_1^* r_1 + \widehat{h}_2 r_2^* + \widehat{h}_3 r_3^* + \widehat{h}_4^* r_4 \\ \widetilde{s_2} &= \widehat{h}_2^* r_1 - \widehat{h}_1 r_2^* + \widehat{h}_4^* r_3 - \widehat{h}_3 r_4^* \\ \widetilde{s_3} &= \widehat{h}_3^* r_1 - \widehat{h}_4^* r_2 - \widehat{h}_1 r_3^* + \widehat{h}_2 r_4^*.\end{aligned} \tag{2.4}$$

2.3 Preamble Design for Multiple Antennas of MB-OFDM

In the MB-OFDM system, the channel estimation is executed at the all sub-band, and in the STBC MB-OFDM system, the channel estimation is carried out for every transmit antenna of the all sub-band. For the channel estimation of each antenna, we design preambles using the Zadoff-Chu sequence—one of the constant-amplitude zero-autocorrelation (CAZAC) sequences [6]. The sequence C_k is as follows,

$$C_k = \exp[j\frac{M\pi k^2}{N}] \tag{2.5}$$

where N is a length of preamble, and $k = [0, 1, \ldots, N-1]$. M is an integer relatively prime to N, and we consider that the case of M is 1.

C_k has the property of a periodic autocorrelation function that is zero everywhere except at a single maximum per period. In the MB-OFDM system, the channel estimation sequence is transmitted twice at each band, and the length of sequence is 128 except the zero-padded suffix and guard interval. By using C_k of 64-symbol length ($N = 64$), we design the extended 128-length Zadoff-Chu sequence with zero padding. Therefore, the proposed sequence can be adopted to the MB-OFDM specification. The extended Zadoff-Chu sequence is as follows

$$S_{8m+n+1} = \begin{cases} C_{4m+n} & for\ \ m : even\ \ including\ \ zero \\ 0 & for\ \ m : odd \end{cases} \tag{2.6}$$

where $m \in \{0, 1, 2, \ldots, 15\}$, $n \in \{0, 1, 2, \ldots, 7\}$.

Figure 2.2 shows the structure of preambles. To apply four transmitting antennas, four preambles can be designed by cyclic shift. For two transmitting antennas, P1 and P3 preambles are used, and for four transmitting antennas, all preambles are used.

From Chu (1972), we deduce generalized relation to determine the number of distinguishable paths D as follows

$$1 \leq D \leq \frac{L}{N_t} \tag{2.7}$$

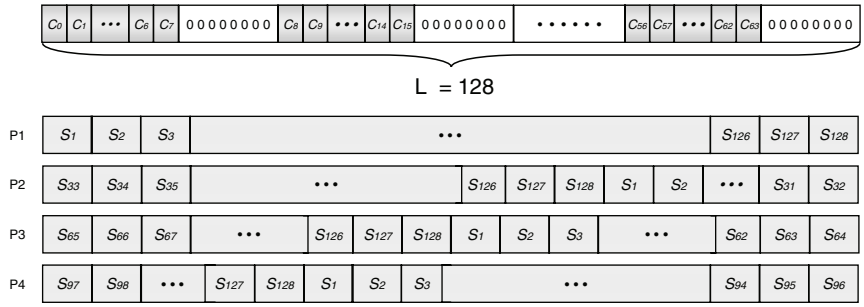


Fig. 2.2 The structure of each 128 length preamble for four transmit antennas system

where L indicates the symbol-length of sequence, and N_t is the number of transmit antenna.

The orthogonality of preambles is broken and the preamble is not suitable for MB-OFDM specification, when the system just uses 64-symbol length sequences. However, extended sequences will keep the property of the orthogonality at channel model (CM) 1 and 2 when the system uses two and four transmitting antennas. It is noted that D is 64 for $N_t = 2$ and is 32 for $N_t = 4$ in Chu (1972). The orthogonality of extended sequences is broken at CM3 and CM4, because the number of channel paths exceeds D. However, because CM3 and CM4 almost never come into existence, the proposed preamble can be used in the WPAN. Using the orthogonality, the receiver can execute the channel estimation and separate channel impulse response (CIR) of each transmitting antenna. There are two main methods in estimating the CSI—the LS and MMSE method. First, we look at the LS method that can be achieved by the following equation

$$\widehat{h}_{LS}^{(p)}[i] = \frac{1}{L}\sum_{l=0}^{L-1}\frac{r[i+l]}{s^{(p)}[l]} \tag{2.8}$$

where $\widehat{h}_{LS}^{(p)}[i]$ is the estimated $i-th$ CIR at the $p-th$ channel branch, and r and s are the received signal and transmitted preamble, respectively.

The LS method only uses the information of training sequences to estimate the CSI. However, the MMSE method utilizes the information of channel characteristics and SNR. To have minimum information quantity, the estimator assumes *a priori* knowledge of noise variance and channel covariance, and is expressed as

$$\widehat{H}_{MMSE} = R_H(R_H + \frac{1}{SNR}I)^{-1}\widehat{H}_{LS} \tag{2.9}$$

where $R_H = E[HH^H]$ is the auto-covariance matrix of H, the superscript $(\cdot)^H$ denotes Hermitian transpose, and I is the identity matrix.

2.4 Performance Evaluation and Discussion

The performance of the proposed preamble is evaluated in terms of MSE in this section. The MSE that shows the error-variance is one of the most common measures for evaluating the quality of the estimator. It can be calculated as follows:

$$MSE = E[(h-\widehat{h})(h-\widehat{h})^H]. \tag{2.10}$$

Figure 2.3 shows the MSE performance of two and four transmitting antennas at CM 1-4 [8]. In the case of two and four transmitting antennas of CM 1 and 2 transmitting antennas of CM2, the system can keep the orthogonality of preambles. However, in the other cases, MSE performances are very poor because the orthogonality of the preamble is broken because of the reasons mentioned through Chu (1972) in Sect. 2.3.

To evaluate the performance of the STBC MB-OFDM system, we plot the BER performance of MMSE and perfect channel estimations of two transmit antennas at data rate 106.7, 160, and 200 Mbps in Figs. 2.2–2.4. The channel model of the BER performance simulation is 1 (CM1). Simulation results show that there is a gap between perfect channel estimation and MMSE estimation. However, the system has the better BER performance than single antenna.

Figure 2.5 shows the effect of the number of transmitting antennas on the BER performance. Simulations are executed in conditions that are one, two and four transmitting antennas at data rate 320, 400, and 480 Mbps. In the case of four

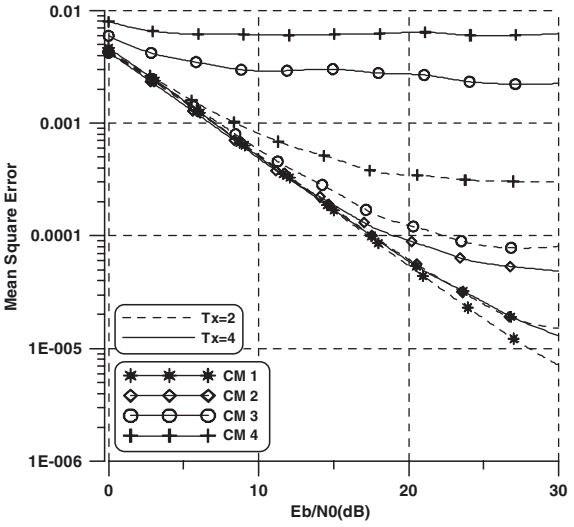


Fig. 2.3 The MSE performance of proposed preambles applied to two and four transmitting antennas with MMSE estimator

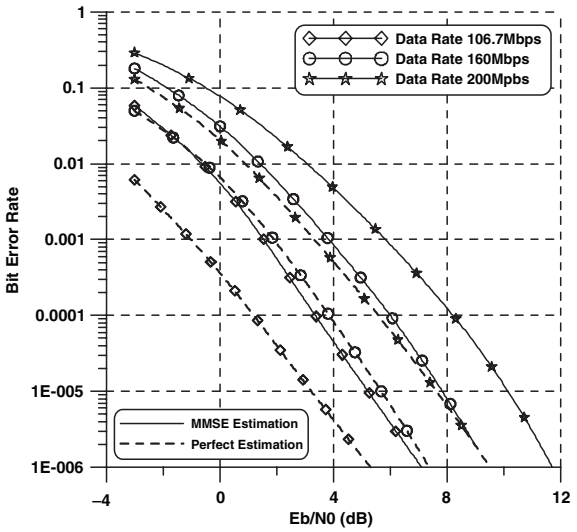


Fig. 2.4 The BER performance of perfect channel estimation and MMSE method at the two-transmitting antenna system

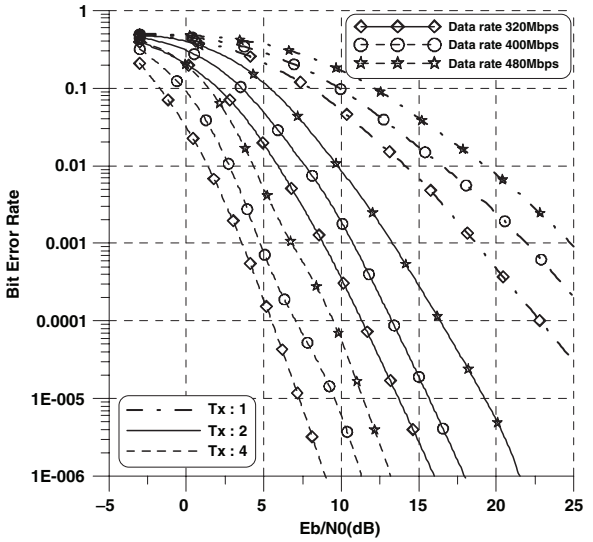


Fig. 2.5 The BER performance of one, two and four transmitting antennas at data rate 320, 400, and 480 Mbps

transmitting antenna, the data rate is 3/4, which is confirmed from Ye, Winters, Sollenberger (2001). Simulation results show that multi-antenna systems have the better performance than single antenna systems. Also, as the number of antennas increases, the system has a better BER performance.

2.5 Conclusions

In this paper, we apply space-time architecture to MB-OFDM systems based on WPAN for high-capacity transmission, and propose the new preamble structure for channel estimation that is required in MIMO architecture. Through the MSE performance, simulation results have shown that the proposed sequence can be adopted to multi-antenna MB-OFDM systems. The BER performance shows that the reliability of STBC MB-OFDM systems is improved efficiently by increasing the number of antennas. As the new preamble is applied, it has been shown that the MB-OFDM system with multipleantennas can achieve high-transmission capacity.

Acknowledgements This research is supported by the Ubiquitous Computing and Network (UCN) Project, the Ministry of Information and Communication (MIC), and the 21st Century Frontier R&D Program in Korea.

References

1. IEEE 802.15 WPAN high rate alternative PHY Task Group 3a(TG3a)[Online], Available: http://www.ieee802.org/15/pub/TG3a.html
2. Qunhua L and Rusch LA (2002) Multiuser Detection for DS-CDMA UWB in the Home Environment. *IEEE Trans. Wireless Communications*, vol. 20, no. 9. December 701–1711
3. Ye L, Winters JH, and Sollenberger NR (2001) Signal detection for MIMO-OFDM wireless communications. *IEEE International Conference Communications*, vol. 10. June 3077–3081
4. MultiBand OFDM Alliance (MBOA) Special Interest Group (SIG), WiMedia Alliance, Inc. (WiMedia) (2005) MultiBand OFDM Physical Layer Specification. Release 1.1, July
5. Alamouti SM (1998) A Simple Transmit Diversity Technique for Wireless Communications, *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8. October 1451–1458
6. Chu DC (1972) Polyphase Codes With Good Periodic Correlation Properties. *IEEE Transactions on information theory*, vol. 18, no. 4. July 531–532
7. Cho D-J, You Y-H, and Song H-K (2004) Channel Estimation with Transmitter Diversity for High Rate WPAN Systems. *IEICE Trans. Commun.*, vol. E87-B, no. 11. Nov.
8. IEEE P802.15-02/490rl-SG2a. (2003) Channel Modeling Sub-committee Report Final. February

Chapter 3
Including Adaptive Behavior in a Wireless Network System

Ulrich Dümichen and Uwe Baumgarten

3.1 Introduction

Interest in wireless networks has increased dramatically in the last few years. In contrast to wired networks, however, wireless networks need to be managed more elaborately. In many cases, it is very useful to have a middleware that takes control over the connection management, fault management, and communication management. Most current approaches concentrate on a subset of topics to be determined in this context. This chapter will take a closer look at the *BlueSpot System* [5], which is a project that develops an adaptive middleware to handle a complete wireless network management system. The middleware is used to control wireless MeshNet infrastructures. There are mobile services on top of the middleware, which are implemented in either native C code or as Java MIDlets. The middleware itself is available for different operating systems, like Linux, Symbian OS, or Windows Mobile, and thus for various devices such as Smartphones or PDAs. Constructed in this manner, the BlueSpot System is used to demonstrate today's existing wireless network management mechanisms and to compare these mechanisms with one another. Additionally, because of the well-defined interfaces the middleware provides, it is very easy to integrate new concepts and to gain benchmarking results that are comparable to existing ones before applying the BlueSpot System. The middleware is built up modularly, so that parts of it can be extended or exchanged very easily.

In contrast to other projects, the BlueSpot System can be adapted to a very wide field of problem areas without losing exiting results from other use cases. The underlying network can be divided into communication nodes. Each communication node is classified as either an infrastructure node, which helps to establish the network; or as a client node, which is a user of the infrastructure. The hardware used for

Ulrich Dümichen and Uwe Baumgarten
Technische Universität München, Fakultät für Informatik, Boltzmannstr.3, 85748 Garching bei München, Germany
e-mail: {duemiche,baumgaru}@in.tum.de

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

infrastructure nodes are Gumstix.[1] While developing the system, the main aspects were self-configuration, self-healing and self-organization of wireless networks with the ability to support nearly every kind of mobile service. Services are equipped with a meta information base. This describes the demands the service has of the middleware. The middleware itself adapts to these demands by changing its behavior dynamically.

In the following sections, this chapter will take a closer look at the architecture of middleware and the contained parameters of the meta information base of a service. Afterwards, it will describe the application range divided into different abstraction layers. These are needed to consider different adaption behaviors. In the next section, this application range will be used to classify different existing approaches, as well as new approaches. It will describe how the network and the middleware adapt to an example parameter—a mobile service demands. After that, a network management approach will be presented that is used to bring the described approaches in line. Therefore, two different strategies will be presented. Finally, the chapter will give a comparison of other approaches to adaptive middleware and will point out the main differences, ending with the description of work still to be done in the future.

3.2 Middleware Description

The middleware consists of different layers. Figure 3.1 shows an overview of its structure. To integrate different wireless network technologies, the BlueSpot System provides an interface that enables connection to underlying network drivers. This interface is located at the bottom of the stack and is named the *Network Adapter Interface*. The underlying network adapters inherit from this interface. Through this method, these network adapters are integrated into the middleware. The network

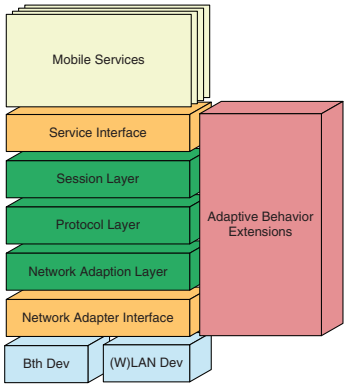


Fig. 3.1 Middleware structure

[1] Gumstix are XScale-based embedded computers that are the size of a chewing gum stick. For additional information, see http://www.gumstix.com/.

adapter interface represents the consistent abstraction of any used network technology, but the popular technology at the moment is Bluetooth. On top of the network adapter interface, sits the *Network Abstraction Layer*, which abstracts the used communication technology from the rest of the middleware stack. This way, the top adjacent layer does not need to know anything about the nature of the underlying network. Additionally, all relevant network attributes are collected in a defined set of parameters and are passed to the upper layers of the stack.

The *Protocol Layer* sits on top of the network abstraction layer. It handles the forming of message headers for communication. Additionally, it is responsible for the controlling of the available communication partners detected by the network adapters. To abstract communication from connections is the job of the next layer—the *Session Layer*. As the name implies, it is responsible for session handling and end-to-end communication. Using Bluetooth as underlying network technology, the session layer is very important. In this case, the communication is often disturbed due to lost connections. After the reconnection, accomplished by the protocol layer, the connection is immediately reassigned to the still existing session and communication can be continued.

This happens completely transparently to the running mobile services on top. To dock mobile services to the middleware, there is the *Service Interface*. It defines the entry point for the execution of the services. It also enables the management of the mobile services with functions, such as those needed to start or stop a service.

As mentioned before, services can be implemented either in native C code or as Java MIDlets. Services implemented in native C are integrated via dynamic library linking, whereas Java MIDlets are started in a new process. In this case, the data exchange between service and middleware is done via socket communication. This is necessary because Java Native Interfaces (JNI) are not supported by Java MIDlets yet.

In general, there are two ways of adding adaptive behavior to software. One way is to directly insert the corresponding code into its source code. This leads to a mishmash of the software's functionalities and adaptive functionalities within the same source code, which affects its readability enormously. The second approach is to define interfaces within each software module. The adaptive behavior is gained by implementing the interface into a separate module and by connecting this module to the corresponding interfaces. This approach has a big advantage—that the software's functionalities and its adaptive behavior are kept strictly separated. It is also easy to later add further adaptive behavior without the need to edit the original code.

A disadvantage of this approach is that, in some cases, it is almost not possible to separate adaptive functionalities from the original ones. This is why both approaches are used within this middleware. While our implementation works, the second was strictly preferred wherever it was possible. The adaptive extensions for this middleware are implemented in the adaption module alongside the main stack. This includes the defined interfaces for each layer. For example, there is the *Protocol Interface*, which enables the binding of different replaceable wireless routing protocols to the protocol layer. The classification of different adaptive extensions assigned to the different middleware layers will be discussed in detail in Sect. 3.5.

Each node of the network is equipped with the middleware. As part of the self-configuration, every node automatically tries to connect to its neighbors to build up a network at startup. At the moment, the topology of the network is defined by a connection list that is given to every node. This was necessary to handle the complexity of network construction, but the replacement of this list, and thus the automization of network forming, will be part of future work. When a connection drops, the concerned nodes automatically try to reconnect to each other. After a defined time out without successful reconnection, the currently used routing protocol is notified. This must build up a different communication path as part of the self-healing behavior.

Available services are announced and distributed by the service discovery service (SDS), which is also used to find a communication partner equipped with the same mobile service within the entire network. If a node does not provide a required service, it is sent by the SDS to this node automatically. Afterward, the service is started immediately. Its binaries are transmitted in the form of a zip archive containing the several different binaries for the different possible hardware architectures and the Java *jar* file.

To be able to run the service on each node in the entire network, it is necessary to have the service in different versions—on the one hand as a Java version, and on the other as precompiled binaries. It is necessary to have these binaries compiled for each hardware platform.

Within the zip archive, there is a folder for each supported hardware platform containing corresponding binaries. Within the BlueSpot System, this procedure is called *service mobility*. The service mobility leads to an overhead of transferred data when the service is delivered to another node, but this is the only way to support all different devices. After the delivery, the SDS searches the received zip archive and extracts the preferred binaries. Subsequently, it is able to start the new service. With these efforts of self-management, the usability of the complete system is simplified enormously.

3.3 Meta Information Base

Because each mobile service has different demands in order to run steadily on top of the middleware, a mobile service has to provide information about its requirements. These requirements are summarized in the *meta information base* (MIB). The MIB contains a set of parameters that are adjustable according to the service's requirements. An example parameter described in the MIB is the minimum bandwidth the service needs to run properly. Besides the minimum bandwidth, there are several additional supported parameters that will be described in the following. Here are the currently considered parameters:

- minimum bandwidth
- maximum bandwidth
- maximum allowed latency time
- priority of The service

- connection-orientated or connectionless service
- streaming or message-based communication
- preferred routing algorithm

In addition to the minimum bandwidth requirement, the maximum bandwidth a service will use is considered as well. This simplifies the arrangement of the available throughput within a domain (see Sect. 3.5.2). A routing algorithm is able to plan the allocation of network resources more easily if it has this information.

Another main parameter is the maximum allowed latency time. In the case of a telephone service, a long latency time would cause reduced speech quality due to long pauses while waiting for an answer from the communication partner.

The forth parameter is the predefined priority for a service. This priority, which is handled as an initial value, is used to determine which of the competitive services will be preferred. There are two approaches available for handling services priority. The simple one uses the parameter defined by the MIB as a constant value. The second approach changes this parameter by considering the bandwidth the service has used within a defined time span. After transmitting a lot of data, the service is downgraded and thus its priority is decreased. In contrast to this, when the service has used less bandwidth, its priority is increased again. This mechanism is useful in achieving a fair allocation of available bandwidth for services, where many services must share a connection with very little total bandwidth.

The parameters described next are boolean values. The first one determines whether the communication occurs in connectionless or connection-orientated manner. This is used to disable the sending of acknowledgment packets to save bandwidth and calculation time in case an indication of successful transmission of data is not needed. If the service is set to streaming mode, the middleware provides memory space for buffering received packets.

The last parameter is a string value and names the routing algorithm that fits best to the properties of the service. If there is more than one service running, the routing algorithm used is selected by the middleware. Before starting a new service it sends a request to all already running services to receive their preferred algorithm and their priority value. These two values are used to calculate a mean value for each routing algorithm spanning the entire network. The one with the highest mean value is chosen as the next algorithm.

3.4 Application Range

To describe the adaptive behavior of the system, it is necessary to have a look at its application range. Figure 3.2 shows the different abstraction layers that are to be considered. On the right hand side we have the mobile service with its MIB. The MIB includes the requirements the service needs to run properly. This brings the service to the point of origin to observe the adaptive behavior of the middleware. Before starting a service, the middleware must analyze the requirements of that service. If the demands cannot be met by the middleware, it must free resources or to

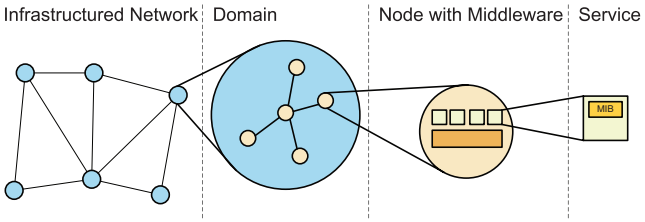


Fig. 3.2 Application range

induce a change of network behavior. If so, neighbor nodes will become involved in the adaption handling. All nodes of the network are grouped into different domains to simplify network management. In the BlueSpot System, the domain is equal to a *Bluetooth Piconet* [2], while in other cases it may make sense to group the nodes into an hierarchical or a geographical order. Most of the time, a domain is built up either in a star layout or as a tree. As a result, there is only one path to a designated node. The connection to other domains is done by bridging nodes. In the case of the BlueSpot System, these nodes belong to two different piconets, hence they form a *scatternet* [3]. These nodes are either called *slave-slave nodes* because they are slaves in both piconets, or *master-slave nodes* because they are master in one and slave in another. As part of the adaption handling, the middleware first examines the situation within the domain. If the demands cannot be met, it must expand its activities over the boundaries of the domain into the entire network. Especially using the Bluetooth technology, it is necessary to examine different possible scatternet topologies. The selection of the best fitting topology depends on the main requirements of the Mobile services. For a detailed discussion of topology selection for Bluetooth-based networks, see Dümichen (2007) [4].

3.5 Adaptive Behavior

The following subsections will describe the adaptive extensions of the middleware. They will explain how the system is able to react to demands described in the MIBs—i.e., required bandwidth for communication. For this reason, it will consider the behavior of the system by means of the abstraction layers described in the section above, but in the opposite direction. It will also point out the special requirements of the middleware within its architecture.

3.5.1 Network

The most common way to handle (QoS) parameters is to change the behavior of the network. This is achieved by adjusting the parameters of the routing algorithm—e.g., by replacing a reactive routing algorithm with a proactive one, fewer control

messages are caused and thus less bandwidth is used.[2] In the case of the BlueSpot System, there are three different algorithms implemented at the moment. The first is a simple flooding routing protocol with duplicate detection. The second is a modified DSR[3] protocol [6], which is an example of a reactive protocol. The last one is a modified DSDV[4] routing protocol [10] as a proactive protocol. Both protocols — the DSR and the DSDV—had to be slightly modified to meet with the requirements of the BlueSpot System.

If a node asks for an exchange of the routing protocol, and the rest of the BlueSpot System agrees, a broadcast message is sent by this node to all other nodes of the entire network. Every node stops its activities and performs the exchange. After the new protocol has been successfully started, every node builds up its former connections to its neighbors and continues to communicate. The messages that were to be sent while the exchange was buffered are sent after the successful reconnection.

If the exchange of the routing protocol does not free enough bandwidth, the routing algorithm itself must react. Therefore, it initiates a route search to find a new communication path, as shown in Fig. 3.3. In this example, the service S_1 on node *Src* is communicating with node *Dest* via the nodes A_1 and A_2. After the additional service S_2 has started its communication via the same route, the available bandwidth falls under the required threshold stated in the MIB of service S_1. Now the routing algorithm reacts by changing the route to node *Dest* via nodes B_1 and B_2. As a result, the network is more load-balanced and service S_1 obtains its needed bandwidth. If the bandwidth of the new path is still not sufficient, the routing algorithm bundles two different paths. This way, the evolving bandwidth is calculated by adding the available bandwidth of the two paths and substracting the bandwidth used for the required controlling messages.

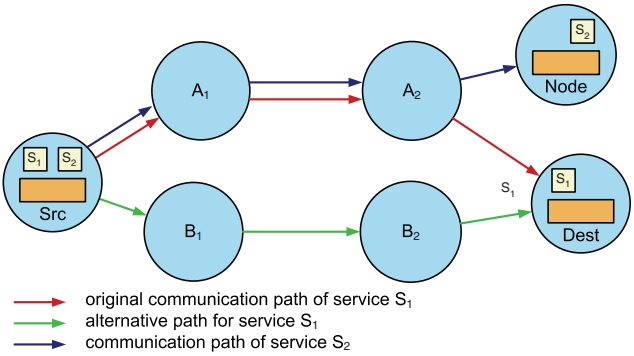


Fig. 3.3 Alternative path finding

[2] Reactive algorithms fit better for scenarios in which clients move at higher speeds through the entire network. In this case, it is necessary that routing updates are made more frequently.

[3] Dynamic Source Routing.

[4] Destination-Sequenced Distance-Vector.

The BlueSpot system provides three different ways for different path routing. The first approach is known as the *best effort* proceeding first introduced by *GÉANT*.[5] Best effort is not configurable and tries to find the best load-balancing on its own. A second approach is the use of priorities. Therefore, we have implemented a modified *DiffServ* protocol [9] that supports four different measures of quality—called *quality classes*. Each connection is associated with such a quality class and will be preferred by a node if its class is higher than the class of another inbounding connection. The third approach is based on resource reservation. This approach picks up the idea of the *IntServ* proceeding [8]. It reserves bandwidth for every connection. Unlike DiffServ, a connection is not associated with a quality class but gets an absolute priority. This is the big advantage of resource reservation: once a connection is established, the system is able to guarantee the requested resources. As a disadvantage, the administration overhead for the resource reservation is very high. Each node contained in a path has to be informed to reserve the requested resources. Using the DiffServ protocol, however every node just has to handle its own connection requests and there is no need for it to know the other involved nodes of a communication path.

When a service indicates in its MIB that it is running in connectionless mode, it does not need the acknowledgments of arrived packets on the receiver side.[6] Thus, bandwidth can again be saved by omitting the *ack* packets.

Considering the middleware architecture, the involved layers are the protocol layer and the session layer. The latter has to handle the different path routing while the protocol layer is responsible for the routing algorithm exchange.

3.5.2 Domain

The considerations above are only valid if the communication takes place between different domains. Within a domain, the adaption occurs in a different way. Here it is not possible to change communication paths because of the described properties of a domain, so the middleware reacts in two different ways.

The first way is to move a node from one domain into another. This is only possible if the concerned node is a client node and it is within range of a second domain. In this case, the middleware on the client node initiates a handover to the new infrastructure node in the second domain. Now the service can use the available bandwidth in the new domain. The handover is implemented in the session layer of the middleware.

The second way is to throttle the bandwidth used by other services. Because of the underlying Bluetooth technology a domain represents a piconet that is built in a star topology with the master in the center. For this reason, the master is the point of origin for bandwidth throttling within a domain.

---

[5] See http://www.geant.net.

[6] This may happen within a sensor network where a sensor does not need to know if its *sent* information receives the listening destination node.

The BlueSpot middleware provides two different approaches to support bandwidth throttling. The first approach is the usage of a token. Each master of a piconet commands a token that it allots to the slaves in a predefined order. This approach is similar to standard token routing protocols that known from the network routing sector. The second approach is a credit system. Each slave obtains an predefined amount of credits that it can employ for communication. If its credits are used up, the slave is not able to send data anymore. After a defined round time, or after all other slaves have used up their credits, every slave obtains a new amount of credits and is able to continue communication. The advantage of the credit system is that all nodes are allowed to communicate simultaneously. While using the token approach, only one node at a time is able to send data. The token approach, however, needs less overhead for managing the communication. Fairness is also easier to handle. The credit system and the token approach are implemented on the height of the protocol layer within the adaption module.

3.5.3 Node with Middleware

Looking at a node, it is not possible to directly manage bandwidth issues. Thus, all approaches need to be indirect. A main starting point is the observation of the CPU load. If the load crosses a predefined threshold, predictably the throughput of the network adapter will drop. This high CPU load is avoided by the middleware in two ways. At first it disables a running service with low priority or it prevents the start of a new one. If this approach does not succeed, the middleware tries to swap services to other nodes. Therefore, it is necessary to determine if a service can be moved to another node at all.

In most cases, this is only possible if the service is a worker and does not need direct user interaction. In the BlueSpot System, this functionality is implemented by using the *service mobility* described in Sect. 3.2. A service is divided into two separated services—one that handles the interaction with the user and another that is responsible for the arithmetic work.

An implemented example within the BlueSpot system is a service that steers a radio-controlled (RC) miniature car. As described before, the service is divided into two different services. One performs the user interaction and enables the user to choose a predefined route for the car to drive. The other calculates each track section. If the car (itself a client of the network, but with very low resources) comes within range of a node, the service running on it already awaits the connection. It immediately takes over the steering of the car until the moment when the car leaves the range of its radio. While steering, the service duplicates itself onto the node, which will control the next track section. To simplify the positioning of the car, every node knows its own coordinates and the maximum velocity of the car. By knowing the start position of the car, and interpreting the steering commands, the steering service is able to calculate the position of the car at any time. Of course, this is not (and never will be) exact driving, but it works fine for demonstration issues of service mobility.

A second parameter to monitor is the available memory. If the node runs out of memory, the middleware falters. In this case, it is not possible to guarantee any bandwidth value. As a solution, the middleware is able to stop a service with a low priority to free some memory. This applies especially to Java services, as they are very resource consuming and run in a separate process. To end a Java service, it is easy to kill its process and to obtain its freed memory. All the described approaches are implemented in the session layer and the service interface.

3.5.4 Service

Considering the situation on the service level, the application of a service should be reviewed by using a simulator. Before the service is allowed to start to run on the real system, its processing is simulated within a test environment. To handle service testing and verification, the BlueSpot System is supplemented with an additional network adapter. This adapter implements the binding to a NS2 simulator,[7] which is able to simulate a complex network topology. The simulator runs through different programmed network situations that cause the middleware to react to fulfill the demands of the service. If these tests show positive results, the service is verified to run properly on the system. Otherwise, the implementation of the service must be reviewed or extra functionality must be added to the middleware to meet the service's requirements.

The simulator is also used to calculate the MIB parameters. If a service provides different classes of quality, the best possible can be selected after evaluation of the service within the simulated environment.

3.6 Network Management

All adaptive extensions described in the previous section can be applied to change the behavior of the entire system to provide additional resources to the on-top running mobile services. Because there are often several different approaches for handling the same situation, however it is necessary to have an elaborate network management system. This system must be able to select the best extensions in any given situation. Within the BlueSpot system, we decided to include our own management system that is responsible for this task. This system is divided into the following modules:

- Network forming
- Modification
- Routing

[7] For additional information about NS2, see http://www.isi.edu/nsnam/ns/.

- Adaptive behavior
- Mobility

Each module encapsulates functions that are accessible by the middleware via defined interfaces. The first module in the list above is the *Network Forming* module. After the startup of the BlueSpot system, it provides the algorithms for the topology forming of the underlying network. This module is not directly part of the adaptive behavior but it is responsible for the initial system configuration that is the point of origin for all other upcoming events.

The *Modification* module contains all functions needed to manage network links during runtime. This way, new infrastructure nodes can be added, malfunctioned nodes can be sorted out, and existing links between nodes can be reallocated. This module enables the BlueSpot system to be flexible concerning its network topology and any changes within it. The BlueSpot system also gains self-healing abilities. All in all, the approaches found in this module are placed with reference to the abstraction layers within the network and the domain layer.

The third module is the *Routing* module. It is responsible for all issues in regard to the several available routing protocols. It includes the functionalities needed for a routing protocol exchange during runtime, as well as the initialization, configuration, or execution of routing protocols. Within the BlueSpot system, the module also supports dynamic routing as described in Sect. 3.5.1, which includes alternative path finding for data transmission.

The most complex module is the *Adaptive Behavior* module, which includes the strategies of adaptiveness in the BlueSpot system. It is responsible for the selection of the right approach of adaption at the right time. One pursued strategy—the network based strategy—is to start with simple adjustments, such as performing resource reservations with the approaches described in Sect. 3.5.1. If the requirements of a running service cannot be met by these adjustments, the Routing module comes in to use. With its help, the BlueSpot System tries to reconfigure the routing tables of the static routing mechanism. Alternatively, if dynamic routing is in use, the module is asked to provide an alternative path between the involved nodes that fits better. If the demands still cannot be fulfilled, the BlueSpot System continues its procedure by dint of the Modification module. Now, as a last attempt, the BlueSpot System tries to reallocate existing communication links within the network and thus changes the network topology.

In addition to this network-based strategy, there is the node-based strategy that contains adaptive approaches described in the domain and the node with middleware abstraction layers. This way, this strategy tries to locally change the behavior of the BlueSpot system to fulfill the requirements of a mobile service. One advantage, is that it is not necessary to know the entire network topology. The usage of the adaptive mechanism is easier. One disadvantage is that the selected approaches cannot be as target-oriented as the approaches done by the network-based strategy.

The last module is the *Mobility* module. It provides all functionalities needed for client mobility. Examples are soft and hard handover, or client inquiring and mobile service exchange mechanisms.

3.7 Related Work

Currently, a considerable amount of research is being done in the field of adaptable middleware. Most approaches aim at extending existing middleware technologies by own attempts. Blair et al. describe how to extend CORBA to gain better network adaptivity for multimedia applications. The approach is to obtain a look into black-box systems to add special algorithms for different network bindings.

Yau and Karim extend CORBA for context-sensitive communication in ubiquitous environments. This approach aims at the special requirements within ad hoc networks, combined with the perception of context-sensitive sensors. Both approaches use an existing CORBA implementation, however, and thus are not able to directly change the behavior of more than one node in a network. Their point of view is to examine only one node and to make the best efforts to optimize its situation. The approach in this paper is to move beyond the boundaries of one single node. The complete system is involved in adaption behavior, and hence there are many more possibilities to meet the demands of a single service.

Concerning the classification of the BlueSpot middleware, McKinley et al. gives a detailed overview of different proceedings in composing adaptive software. To classify the kind of adaption, it describes two different proceedings—the *parameter adaption* and the *compositional adaption*. The first one focuses on an advancement of performance by changing predefined parameters. As described in Sect. 3.5.2, bandwidth can be gained by increasing the priority of a service. If the middleware initiates a routing protocol exchange, a new code is loaded and thus new algorithms are included in the middleware. This is what McKinley et al. call compositional adaption. Further, McKinley et al. examine the different constituents of an adaptive middleware. This idea goes back to Schmidt, who divides a middleware into four layers: *host-infrastructure middleware*, *distribution middleware*, *common middleware* and *domain-specific middleware*. McKinley et al. say, that these four layers bridge the gap between an application program and the underlying operating systems, network protocols, and hardware devices.

The BlueSpot middleware stack described in Sect. 3.2 also fits into this decomposition. Then network adapter interface and the network adaption layer represent the host-interface middleware. They cloak the heterogeneity of the underlying network devices. The protocol layer and the session layer fit in both the Distribution middleware and the Common middleware. They handle fault tolerance as well as high-level programming abstraction to enable developers to write distributed applications in a way similar to stand alone applications. The Service interface implements parts of the Distribution middleware as well as of the Domain-specific middleware. Services are able to connect to the middleware by implementing the interface that the Service interface provides. With the possibility of implementing services in both native C code and as Java MIDlets, each service can be modified according to the service's requirements.

3.8 Future Work

Upcoming activities within the BlueSpot System project will include the development of a monitoring tool. This tool will be used to illustrate the behavior of the complete system by dint of a graphical user interface (GUI). At the moment, the entire code is added by hooks that send user datagram protocol (UDP) massages via a local area network (LAN) device to a predefined destination address. The monitor running on the destination device will record all incoming messages and present them in the GUI. It is necessary to use an extra LAN device so monitoring influences the system's performance as little as possible. With the monitoring tool, the adaptive behavior of the middleware will be obvious. Different approaches—i.e., the use of different routing algorithms—can also be easily compared to each other, and new approaches can be verified quickly. It is also necessary to do more benchmarking on the entire system, especially to evaluate the different adaptive behavior efforts. These benchmarks will be presented in future publications. The only benchmarks that exist at the moment are values concerning the throughput and the latency times of the underlying Bluetooth scatternets with the use of the BlueSpot Bluetooth device driver. However, these values are still not comparable to other implementations, so they will also be published after an exact verification occurs in the future. As described in Sect. 3.2, further work will be done to automate network forming. A main task to achieve is that all nodes manage their connections independantly. The difficulty in this task is to build up the preferred domain-based network with the slave-slave nodes as bridges—e.g., a node must realize that it is a bridge between two domains and therefore must turn itself into a special bridging node. In this node, no messages are reviewed but are only forwarded to the other domain to save resources and gain latency time.

Acknowledgements We thank Michael Steinle and Matthias Langhammer for their suggestions and valuable contributions they provided to our project. Further, we thank *Jörg* Preißinger for his suggestions and valuable comments in discussions of our work, and Tina-Marianne Mensch for her ir-replaceable help in proofreading.

References

1. Blair G, Coulson G, Davies N, Robin P, and Fitzpatrick T (1997) Adaptive middleware for mobile multimedia applications. In: Proceedings of the IEEE 7th International Workshop on Network and Operating System Support for Digital Audio and Video. IEEE, IEEE, 245–254
2. Bluetooth SIG (2004) Bluetooth Core Specification v2.0 + EDR-Piconets. Chap. 4, Bluetooth SIG, 51–58
3. Bluetooth SIG (2004) Bluetooth Core Specification v2.0 + EDR-Scatternet Chap. 8, Bluetooth SIG, 177–178
4. Dümichen U and Baumgarten U (2007) Aspects for forming structured bluetooth based scatternets. In: Proceedings on PDPTA '07. CSREA Press

5. Dümichen U and Baumgarten U (2004) BlueSpot System–Infrastructure for Mobile Services. In: Proceedings of 49. Internationales Wissenschaftliches Kolloquium (IWK) Ilmenau, Vol. 1. Shaker Verlag, pp. 375–380
6. IETF Draft MANET Working Group (2004) The dynamic source routing protocol for mobile ad hoc networks (dsr).
7. McKinley PK, Sadjadi SM, Kasten EP, and Cheng BHC (2004) Composing adaptive software. *Computer* 37(7):56–64
8. Network Working Group (1997) Rfc 2210: The use of rsvp with ietf integrated services.
9. Network Working Group (1998) Rfc2474: Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers
10. Perkins CE and Bhagwat P (1994) Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In: *SIGCOMM 94*. ACM
11. Schmidt DC (2002) Middleware for real-time and embedded systems. *Commun. ACM* 45(6):43–48
12. Yau SS, and Karim F (2004) An adaptive middleware for context-sensitive communications for real-time applications in ubiquitous computing environments. *Real-Time Systems* 26(1):29–61

Chapter 4
Classification Techniques Using Neural Networks and Clustering Architecture for Wireless Sensor Networks

Sudhir G. Akojwar and Rajendra M. Patrikar

4.1 Introduction

Wireless sensor networks are becoming more popular because advances in sensor technology, low-power electronics, and low-power radio frequency design have enabled the development of small, relatively inexpensive and low-power sensors—called microsensors. These wireless microsensor networks represents a new paradigm for extracting data from the environment, and enable the reliable monitoring of a variety of environments for applications that include surveillance, machine failure diagnosis, chemical/biological detection, habitat monitoring, and environmental monitoring, etc. An important challenge in the design of these networks is that two key resources—communication bandwidth and energy—are significantly more limited than in a tethered network environment. These constraints require innovative design techniques to use the available bandwidth and energy efficiently [1]. Communication consumes the largest part of the energy budget in wireless sensor networks. Hence, attempts must be made to implement techniques to save energy on communications. This chapter discusses classification techniques using the ART1 (Adaptive Resonance Theory) and Fuzzy ART (FA) neural networks models.

Real time classifiers classify the sensor readings and then only its class ID needs to be communicated further. This brings a saving of sufficient amount of energy.

Sudhir G. Akojwar
Member IEEE, Assistant Professor, Department of Electronics Engineering,
Rajiv Gandhi College of Engineering Research and Technology, Chandrapur, India
e-mail: sudhirakojwar@rediffmail.com,sgakojwar@ieee.org

Rajendra M. Patrikar
Senior Member IEEE, Professor, Department of Electronics and Computer Science,
Visvesvaraya National Institute of Technology, Nagpur, India
e-mail: rajendra@computer.org,rmpatrikar@vnitnagpur.ac.in

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

4.2 Wireless Sensor Networks (WSN)

Wireless sensor networks are an emerging technology that has a wide range of potential applications including environment monitoring, smart spaces, medical systems, and robotics exploration. Such networks will consist of a large number of distributed nodes that organize themselves into a multihop wireless networks. Each sensor node has one or more sensors, embedded processors, and low-power radios, and is normally battery-operated. Typically, these nodes coordinate and communicate to perform a common task. These sensor nodes remain largely inactive for a long time, but become suddenly active when change in the parameter is detected by sensors.

4.3 Artificial Neural Networks Paradigm in Sensor Networks

Data processing happens to be the most important activity in wireless sensor networks. Because it is a network of dense population of nodes, it becomes a data-driven system. Hence efficient data clustering techniques must be used to reduce the data redundancy and, in turn, reduce overhead on communication. This can be very well accomplished by using some of the algorithms developed within the artificial neural networks paradigm, which can be easily adapted to Wireless sensor networks (WSN). Artificial neural network (ANN) consists of small computing units called neurons, arranged in different layers and interconnected with each other. Simple mathematical computations are performed at each neuron. There are lots of advantages to using ANN in WSN, and will meet the requirements for WSN, like a simple parallel-distributed computations, distributed storage, and data robustness. Thus it can reduce memory requirements to the minimum. The WSN nodes can be clustered in different groups, thereby reducing the dimension of the network. This lowers communication and memory cost.

Sensor networks place several requirements on a distributed storage infrastructure. These systems are highly data driven and are deployed to observe, analyze, and understand the physical phenomenon. A fully centralized data collection strategy is not feasible, given the energy constraints on sensor node communication, and is also inefficient given that sensor data has significant redundancy both in time and in space. Using ANN, it is easy to get compressed summaries of large spatio–temporal sensor data and similarity queries, such as detecting correlations and finding similar patterns. Several algorithms for data processing are available that are modified regression techniques from the field of multidimensional data series analysis, in other scientific fields, principal component analysis, and multidimensional scaling, as discussed in Guestrin et al. Some of the algorithms (ART and Fuzzy ART) that are well developed in the paradigm of ANN are well suited to fit into the requirements imposed on WSN for simple parallel distributed computation, distributed storage, data robustness, and classification of sensor readings.

Auto classification of the sensor readings is important in sensor networks because the data obtained with them has high dimensionality and very large volume, which could easily overwhelm the processing and storage capacity of a centralized database system.

Neural network algorithms use simple computations and do not represent big burdens to memory. Unsupervised learning ANN typically performs dimensionality reduction or pattern clustering. They are able to discover both regularities and irregularities in the redundant input data by an iterative process of adjusting weights of interconnections between large numbers of simple computational units. As a result of dimensionality reduction obtained easily from the outputs of these algorithms, lower communication costs and thus bigger energy savings can also be obtained. ART and Fuzzy ART neural network models can be successfully implemented for WSN. These are the unsupervised models. A neural network algorithm can be implemented in the computational unit of the sensor node. Thus, instead of reporting the raw data, each node can send only the cluster number where the current sensory input pattern has been classified. In that way, a huge dimensionality reduction can be achieved depending on the number of sensor inputs in each unit. At the same time, communication savings will benefit from the fact that the cluster number is a small binary number unlike sensory readings, which can be several bytes long. Thus, specific ANN algorithms such as ART and Fuzzy ART models can prove their merits in a WSN environment.

4.4 Clustering and Classification Techniques

Wireless sensor networks are highly data centric. Data communication in WSN must be efficient one and must consume optimum power. Every sensor node consists of multiple sensors embedded in the same node. Thus, every sensor node is a source of data. These raw data streams [3] cannot be immediately communicated further to the neighboring node or the base station. These sensor data streams need to be classified. A group of sensor nodes forms a cluster. Each node transfers data to a cluster head and then the cluster head aggregates the data and sends it to a base station. Hence, clustering and classification techniques are important and can give new dimension to the WSN paradigm. Basically, classification systems are either supervised or unsupervised [4], depending on whether they assign new inputs to one of an infinite number of discrete supervised classes or unsupervised categories, respectively. In supervised classification, the mapping from a set of input data vectors ($X \in R^d$, where d is the input space dimensionality) to a finite set of discrete class labels ($y \in 1, \ldots C$, where C is the total number of class types) is modeled in terms of some mathematical function y = y(x,w), where w is a vector of adjustable parameters. The values of these parameters are determined (optimized) by an inductive learning algorithm (also termed inducer), whose aim is to minimize an empirical risk functional (related to an inductive principles) to a finite data set of input-output examples ($(x_i, y_i), i = 1, \ldots, N$), where N is the finite cardinality of the available

representative data set. When the inducer reaches convergence or terminates, an induced classifier is generated.

In unsupervised classification, called clustering, or exploratory data analysis, no labeled data are available. The goal of clustering is to separate a finite unlabeled data set into a finite and discrete set of "natural," hidden data structures, rather than provide an accurate characterization of unobserved samples generated from the same probability distribution. This can make the task of clustering fall outside of the framework of unsupervised predictive learning.

There is no universally agreed upon definition for clustering. Most researchers describe a cluster by considering the internal homogeneity and the external separation; patterns in the same cluster should be similar to each other, while patterns in different clusters should not. Both the similarity and the dissimilarity should be examinable in a clear and meaningful way.

ART was developed by Carpenter and Grossberg, as a solution to the plasticity and stability dilemma [5]. ART can learn arbitrary input patterns in a stable, fast, and self-organizing way, thus overcoming the effect of learning instability that plagues many other competitive networks. ART is not, as is popularly imagined, a neural network architecture. It is a learning theory that resonance in neural circuits can trigger fast learning. As such, it subsumes a large family of current and future neural network architectures, with many variants. ART1 is the first member, which only deals with binary input pattern, although it can be extended to arbitrary input patterns by a variety of coding mechanisms. ART2 and Fuzzy ART extend the applications to analog input patterns [7]. Fuzzy ART (FA) benefits the incorporation of fuzzy set theory and ART [5]. FA maintains similar operations to ART1 and uses the fuzzy set operators so that it can work for all real data sets. FA exhibits many desirable characteristics, such as fast and stable learning and typical pattern detection.

4.5 Basics of the ART1 Algorithm

The ART1 model is described in Fig. 4.1, which is discussed in detail in Freeman et al. It consists of three layers (basically only two layers). Layer F0 is the input layer that copies the inputs to the F1 layer and has N nodes (one for each binary bit of input pattern). Layer F1 is the comparison layer, and layer F2 is the recognition or category layer. Layers F0, F1, and F2 are constituted of N, N, and M neurons respectively. Each node in the F2 layer represents a *cluster* or *category*. In this layer, only one node will become active after presentation of an input pattern $I \equiv (I_1, I_2, \ldots\ldots I_N)$. The F2 layer category that will become active would more closely represent the input pattern I. If the network detects novel inputs for which there is no preexisting category, a new category will be formed. Each F1 node $-x_i$ is connected to all F2 nodes $-y_i$ through bottom-up connection weights z_{ij}^{bu}, so that the input received by each F2 node y_i is given by

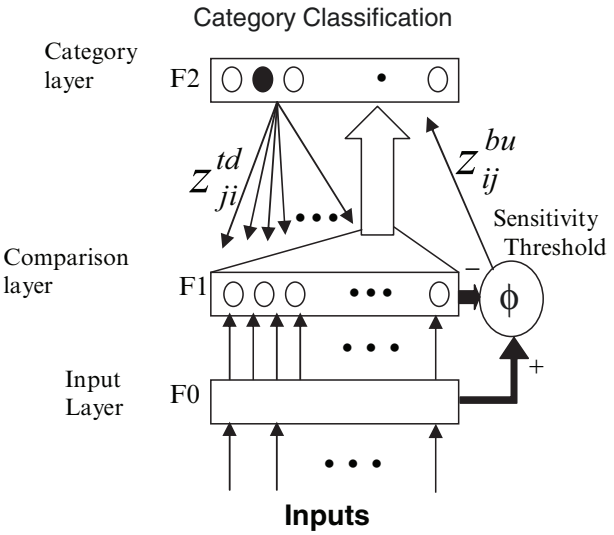


Fig. 4.1 Architecture of ART1 model

$$T_j = \sum_{i=1} z_{ij}^{bu} I_i \tag{4.1}$$

$$y_J = 1, \quad \text{if,} \quad T_j = \max_k\{T_k\} \quad \text{otherwise} \quad y_J = 0$$

Bottom-up weights z_{ij}^{bu} take any real value in the interval [0,K], where

$$K = \frac{L}{L-1+N}, \text{and} L > 1$$

Layer F2 acts as a winner-take-all network—i.e., a competitive layer for the outputs—so that all nodes y_i will stay inactive except the one that receives the largest bottom up input T_j. Once an F2 winning node arises, a top-down template is activated through the top-down weights z_{ji}^{td}. In the fast learning Type 3 model, top-down weights z_{ji}^{td} takes values 0 or 1. Let us call this top-down template $X = (X_1, X_2, \ldots\ldots X_N)$. The resulting vector X is given by the equation

$$X_i = I_i \sum_j z_{ji}^{td} y_i \tag{4.2}$$

Because only one y_i is active, let us call this winning F2 node y_J, so that $y_J = 0$ if $j \neq 0$ and $y_J = 1$. In this case, we can state

$$X_i = I_i z_{Ji}^{td} \quad \text{or} \quad X = I \cap z_J^{td} \tag{4.3}$$

where $z_J^{td} \equiv (z_{1J}, z_{2J}, \ldots . z_{NJ})$. This top-down template will be compared against the original input pattern I according to a predetermined vigilance criteria, tuned by the vigilance parameter $0 < \rho \leq 1$ so that two alternatives may occur:

1. If $\rho|I| \leq |I \cap z_J^{td}|$, the active category J is accepted and the system weights will be uploaded to incorporate this new knowledge.
2. If $\rho|I| > |I \cap z_J^{td}|$, the active category J is not valid for the actual value of the vigilance parameter ρ. In this case y_J will be deactivated (reset) making $T_j = 0$, so that another y_j node will become active through the winner-take-all action of the F2 layer.

Once an active F2 node is accepted by the vigilance criterion, learning takes place. The weights will be updated according to the following algebraic equations:

$$\begin{aligned} z_J^{bu} \mid_{new} &= \frac{L(I \cap (z_J^{td})_{old})}{L-1+|(I \cap (z_J^{td})_{old}|} \\ z_J^{td}|_{new} &= I \cap (z_J^{td})_{old} \end{aligned} \tag{4.4}$$

Only the weights of the connections touching the F2 winning node y_J are updated.

4.6 VLSI-Friendly ART1 Algorithm

The implementation complexities can be further reduced by making some sort of normalization of the weight templates [6]. There are two templates of weights that have to be built—the set of bottom-up weights z_{ij}^{bu}, each of which has to store a real value belonging to the interval [0,K]; and the set of top-down weights z_{ji}^{td}, each of which stores either the value 0 or the value 1. Looking at Eq. (4.3), it can be seen that the bottom–up set $\{z_{ij}^{bu}\}$, and the top-down set $\{z_{ji}^{td}\}$ contain the same information: each of these sets can be fully computed by knowing the other set. It can be seen that the bottom–up set z_{ij}^{bu} is a kind of normalized version of the top-down set z_{ij}^{bu}. This way, we can substitute the two sets $\{z_{ij}^{bu}\}$ and $\{z_{ji}^{td}\}$ by a single binary valued set $\{z_{ij}\}$, and modify Eq. (4.1) to take into account the normalization effect of the original bottom-up weights,

$$T_j = \frac{LT_{Aj}}{L-1+T_{Bj}} = \frac{L\sum_{i=1}^{N} z_{ij}I_i}{L-1+\sum_{i=1}^{N} z_{ij}} \tag{4.5}$$

We can show Eq. 4.3, which can be substituted by the following equation, resulting in a system that preserves all the computational properties of the original ART1 architecture:

$$T_j = L_A T_{Aj} - L_B T_{Bj} + L_M \tag{4.6}$$

where $L_A > L_B$ are positive parameters that play the role of the original L and L-1 parameters. $L_M > 0$ is a constant parameter needed to assure that $T_j \geq 0$ for all possible values of T_{Aj} and T_{Bj}. Both the algorithms are implemented and tested on MATLAB with different possible patterns of input.

4.7 Fuzzy Art Neural Network Model

The general structure of the Fuzzy ART neural network is shown in Fig. 4.2 [7].

$$Y_J = 1, \text{ if, } T_j = \max_k \{T_k\} \text{ otherwise } Y_J = 0$$

It consists of two layers of neurons that are fully connected—a 2M-neuron input or comparison layer (F1), and an N-neuron output or competitive layer (F2).

A weight value z_{ji} is associated with each connection, where the indices i and j denote the neurons that belong to the layer F1 and F2, respectively. The set of weights $Z = \{z_{ji} : i = 1,2,\ldots,2M; j = 1,2,\ldots,N\}$ encodes information that defines the categories learned by the network. These can be modified dynamically during network operation. For each neuron j of f2, the vector adaptive weights $z_j = (z_{j1}, z_{j2}, \ldots, z_{j2M})$ correspond to the subset of weights $((z_j \subset Z)$ connected to neuron j. This vector z_j is named prototype vector or template, and it represents the set of characteristics defining category j. Each prototype vector z_j is formed by the characteristics of the input patterns to which category j has previously been assigned through winner-take-all competition.

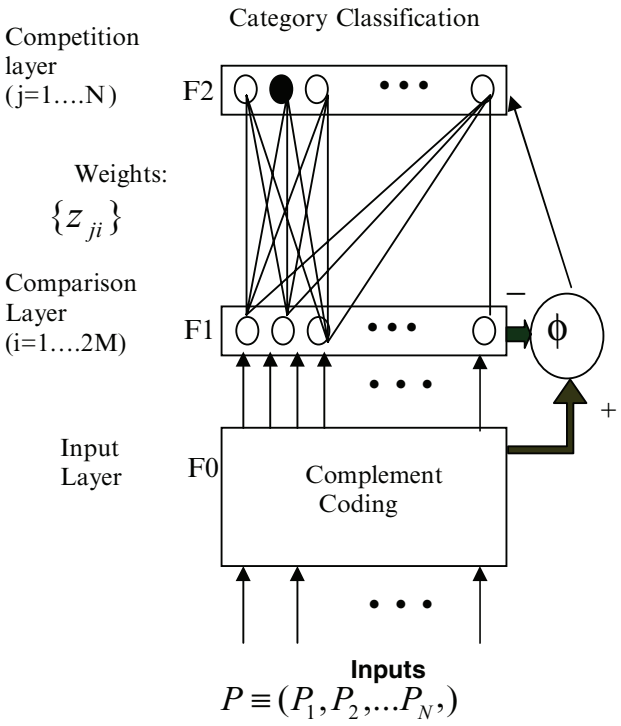


Fig. 4.2 Fuzzy ART neural network model

4.8 Fuzzy ART Algorithm

This algorithm can be described in five execution steps:

1. **Weights and Parameter Initialization:** Initially, all the neurons of F2 are uncommitted, and all weight values z_{ji} are initialized to 1. An F2 neuron becomes committed when it is selected for an input P. Then, the corresponding weights z_{ji} can take values expressed by real numbers in the interval [0,1]. The Fuzzy ART weight vector z_{ji} subsumes both the bottom-up and top-down weight vectors of ART1.
2. **Input Vector Coding:** When a new input vector $P \equiv (P_1, P_2, \ldots, P_N)$ of N elements (where each element is a real number in interval [0,1] is presented to the network, it undergoes a preliminary coding at layer F0. Complement coding of vector P results in a network input vector I of 2N elements such that:

$$I \equiv (P, P^c) = (P_1, P_2, \ldots P_N; P_1^C, P_2^C, \ldots, P_N^C), \text{ where } P_i^c = 1 - P_i.$$

   This coding is used to prevent a category proliferation when the weights erode.
3. **Category Choice:** With each presentation of an input I to F1, the choice function $T_j(I)$ is calculated for each neuron j in F2:

$$T_j = \frac{|I \wedge z_{ji}|}{\alpha + |z_{ji}} \tag{4.7}$$

   Here, the notation $|r|$ represents the cardinality of vector r—i.e., $|r| = \sum_{i=1}^{2N} r_i, \wedge$ is the fuzzy logic AND operator. $I \wedge z_J = (\min(I_1, z_{j1}, \min(I_2, z_{j2}), \ldots,$ $\min(I_{2N}, z_{j2N}))$ and α is a user-defined choice parameter such that $\alpha > 0$. F2 is a winner-take-all competitive layer, where the winner is the neuron $j = J$ with the greatest value of activation T_j for the input I, $T_J = \max\{T_j : j = 1, \ldots, N\}$. If the same T_j value is obtained by two or more neurons, the one with the smallest index j wins. The winning neuron J is retained for steps 4 and 5.
4. **Vigilance Test:** This step serves to compare the similarity between the prototype vector of the winning neuron z_{Ji} and input I, against a user-defined vigilance parameter ρ through the following test

$$\frac{|I \wedge z_J^{\downarrow}}{|I|} \geq \rho \tag{4.8}$$

   where $\rho = [0, 1]$. This comparison is carried out on layer F1—the winning neuron J transmits its learned expectancy, z_J, to F1 for comparison with I. If the vigilance test in Eq. (4.2) is passed, then neuron J becomes selected and is allowed to adapt its prototype vector (Step 5). Otherwise, neuron J is deactivated for the current input I. T_J is set equal to –1 for the duration of the current input presentation. The algorithm searches through the remaining F2 layer neurons (Steps 3 and 4) until some other neuron J passes the vigilance test. If no committed neuron from

the F2 layer can pass this test, an uncommitted neuron is selected and undergoes a prototype vector update (i.e., the new class is assigned for the input).

5. **Prototype Vector Update:** The prototype vector of the winning neuron J is updated according to

$$z_J^{new} = \beta(I \wedge z_J^{old}) + (1-\beta) z_J^{old} \tag{4.9}$$

where β is a user-defined learning rate parameter such that $\beta = [0,1]$. The algorithm can be set to slow learning, with $0 < \beta < 1$; or to fast learning rate, with $\beta = 1$. Once this update step is accomplished, the network can process a new input vector (Step 2).

4.9 Reformulated Fuzzy-ART VLSI Architecture

To realize our approach towards VLSI implementation of real-time classifiers, we present a dedicated VLSI system architecture for the reformulated Fuzzy ART algorithm as shown in Fig. 4.3.

The architecture is divided into three sequential phases—Initialization, Recall, and Learning. It is not economical to implement the whole of this architecture as massively parallel digital networks on a single integrated circuit. Hence, a little sequential approach is used for the given application of an WSN. The reformulated Fuzzy ART algorithm is functionally equivalent to the original one; it is expressed as a stream of operations that can be carried out sequentially. Other modifications will order the sequence of operations. Computations of the choice functions (a division, T_j, for every committed neuron j), and their components ($|I \wedge z_j|$ and $|z_j|$), constitute a substantial proportion of the total processing effort. Fortunately, once the elements $|I \wedge z_j|$ and $|z_j|$ have been computed, their values can be re-used on later occasions. This permits more efficient organization of the algorithm. In addition, there is no need to compute the choice function T_j of a neuron j that would not pass the vigilance test. This modification circumvents the original algorithm's search process, because the neuron j = J with the greatest T_j value that can pass the vigilance test is readily selected. The vigilance test should thus occur as soon as the component $|I \wedge z_j|$ is available for a neuron j. Furthermore, because complement coding is an important normalization such that, by definition, $|I| = |(p; p^c)| = M$, the vigilance test reduces to $|I \wedge z_j| \geq \rho \cdot M$. In parallel with the vigilance test, it is convenient to carry out a subset test, because it depends only on the elements $|I \wedge z_j|$ and $|z_j|$. This test detects when a prototype vector z_j is a subset of I—that is when a prototype vector $|I \wedge z_j| = |z_j|$. If this neuron j is chosen and passes the vigilance test, then its prototype vector z_J remains unchanged during the learning phase because $z_J^{new} = I \wedge z_j = z_J$ in fast learning mode ($\beta = 1$). Therefore, the learning phase can be bypassed, and the network speed increased.

These observations lead to a reformulation of the Fuzzy ART algorithm, as shown in Fig. 4.3 for the fast learning case. The algorithm sequentially processes committed neurons only, and N now represents the number of committed neurons that

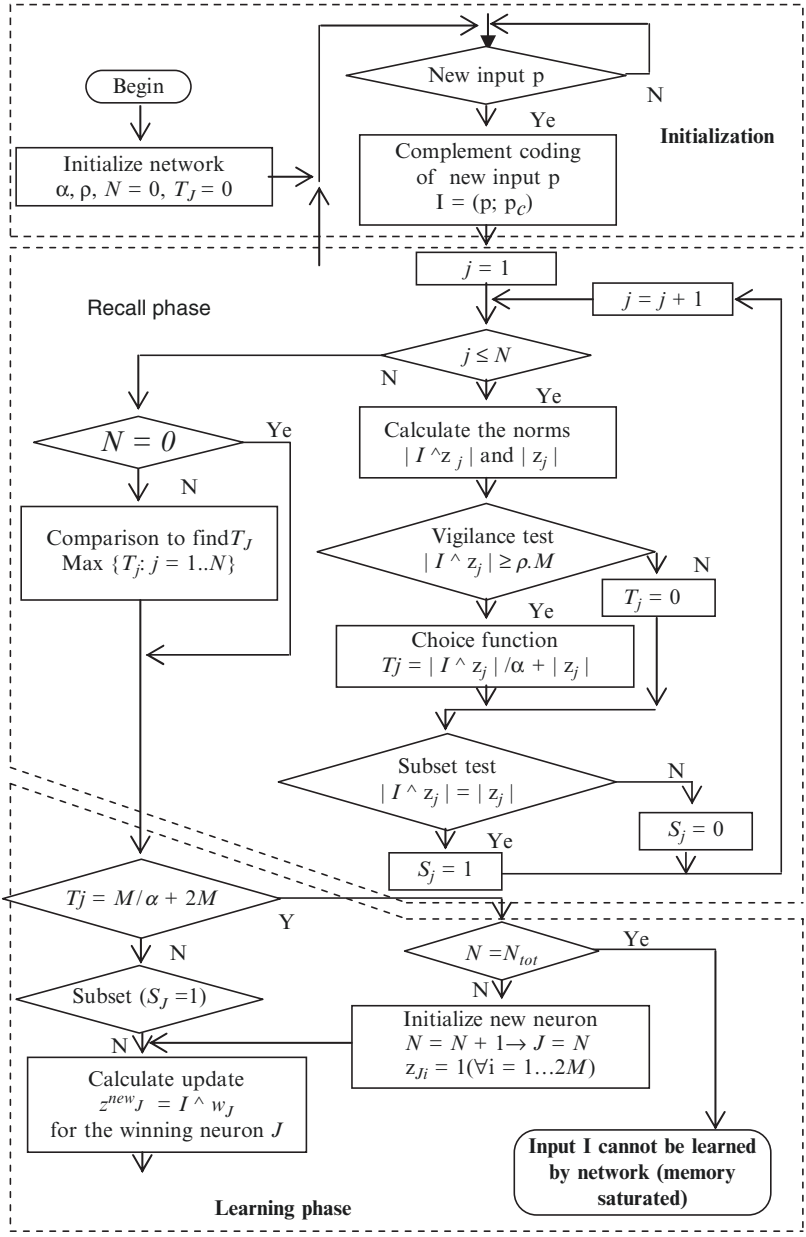


Fig. 4.3 Flowchart—reformulated Fuzzy ART algorithm

start from 1 and increase progressively as learning takes place. The reformulated algorithm is divided into three parts: Initialization, Recall, and Learning. During the Initialization phase, the network parameters are initialized. During the Recall phase, the values T_j and S_j are computed for all committed neurons. If a neuron j is a subset choice of I, then S_j is set equal to 1. If a neuron j passes the vigilance test for I, then the choice function I is computed and $T_j > 0$—otherwise, T_j is set equal to 0. At the end, the winning neuron J is chosen. During the Learning phase, in the case where $T_J \geq \frac{M}{\alpha+2M}$, the winning neuron J corresponds to a committed category that passes the vigilance test. If J is a subset choice for I ($S_J = 1$), then the prototype vector update is bypassed; otherwise, $S_J = 0$, the prototype vector w_J is updated according to Eq. 4.9. In the case where $T_J < \frac{M}{\alpha+2M}$, no committed category has passed the vigilance test. If the maximum number of neurons is not attained ($\mathrm{N} < N_{tot}$), then an uncommitted neuron $\mathrm{J} = \mathrm{N} + 1$ is assigned to I; otherwise ($N = N_{tot}$), the network is unable to categorize I because its memory is saturated. After the Learning phase, the network is ready to accept another input vector.

4.10 Clustering Architectures for Wireless Sensors Network

The strength of the ART1 model is its unique ability to solve a *stability plasticity* dilemma. In fast learning mode, it takes extreamly short training times, it can generate an incrementally growing number of clusters based on the variations in the input data. The network runs entirely autonomously. It does not need any outside control, because it can learn and classify at the same time, provide fast access to match results, and is designed to work with infinite streams of data. All these features make it an excellent choice for applications in wireless sensor networks.

The above modified FuzzyART and ART1 algorithms can be successfully used for sensor networks. The use of ANN—in particular, ART1 and Fuzzy ART—is discussed in detail in Akojwar et al. (2005).

For organizing the distributed data of the sensors, this ART1 neural network can be used in three different clustering schemes for WSNs.

1. **One Clusterhead Collecting all Sensors' Data:** In this architecture, the sensor nodes send the sensory reading to one of them chosen to be a clusterhead, where an Fuzzy ART neuron is implemented. This model, as shown in Fig. 4.3, brings advantages in that we need not to fix the number of clusters (categories) that the network should learn to recognize in advance.
2. **Each Unit is a Clusterhead Clustering Data with Different Levels of Details:** In this architecture, as shown in Fig. 4.4, each unit receives the input data from all sensors' nodes in one cluster by broadcast. Then, each unit classifies the sensor data with different sensitivity thresholds, thus providing a general overall view of the network. Instead of having only one cluster (since the data is broadcast anyway) all sensors nodes in this architecture collect data from units all over and they all have FA implementations.

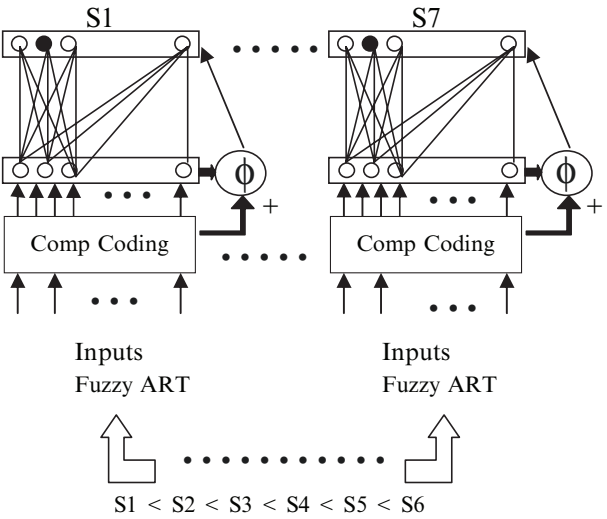
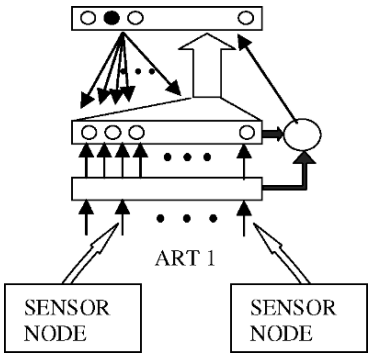


Fig. 4.4 Redundant clusterheads collect data at different levels of detail

We can use different sensitivity thresholds with which we achieve different kinds of views over the same data—coarser with a smaller number of categories, or more detailed with a bigger number of categories.

3. **Clusterhead Collects only Clustering Outputs from Another Unit:** Each sensor node has Fuzzy ART implementations classifying only its sensor readings, as shown in Fig. 4.5. One of these units can be chosen to be a clusterhead, collecting and classifying only the classifications obtained by other units. Because the clusters at each unit can be represented with binary values, the neural network implementation at the clusterhead is ART1 with binary inputs.



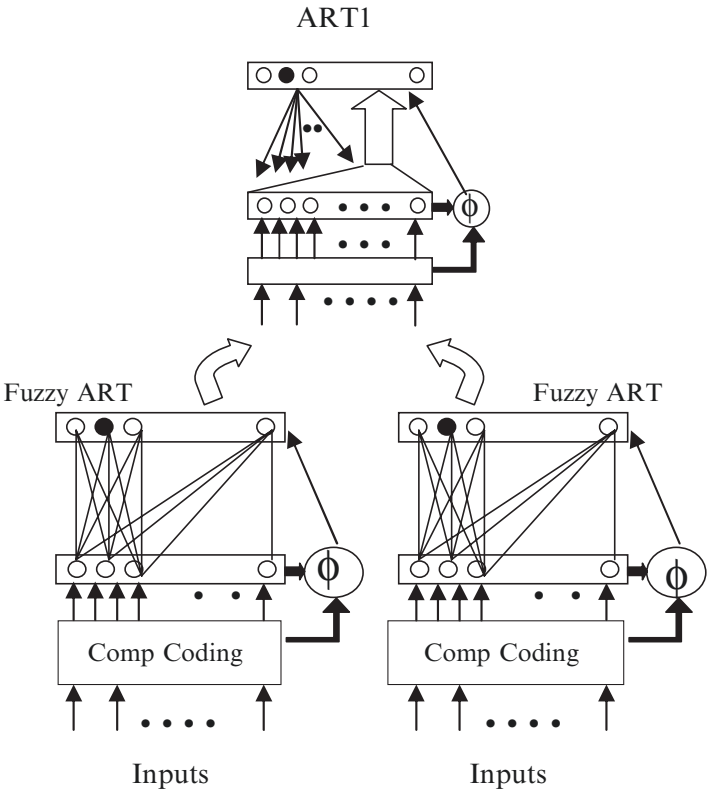


Fig. 4.5 One clusterhead collecting and classifying the data after they are classified at the lower level

4.11 WSN in Industrial Environments

In industrial environments, abundant RF noise is a significant concern. Sources may be explicit, such as 802.11 access points, wireless radios, or radar/radio navigation equipment. Other sources are the result of radiated electrical noise from machinery, such as frequency motor controllers or solid-state switchgear. Such noise sources may adversely impact the reliability and power consumption of sensor networks. Conversely, the impact of interference from the sensor network, particularly on other plant communication channels, must also be taken care of while deploying WSN. The deployment issue for sensor networks is discussed at length in Krishnamurthy etal. The classification and clustering architecture are discussed in Akojwar et al. (2007).

Before deploying WSNs in industrial applications, it should be clear that the coverage areas and reliability of data in the industrial environment may suffer due to noises, co-channel interferences, and multipath and other interferers. The signal strength can be severely affected in factory environments due to reflections of walls

Table 4.1 Inference sources in industrial environments

Broadband interferences	Narrowband interferences
Motors	Cellular telephone
Inverters, SCR circuits	Radio and TV transmitters
Electrical switch contacts	Power-line hum
Computers, ESD	Signal generators
Ignition systems	Local oscillator, UPS system
Voltage regulators	Test equipment
Lighting electromagnetic pulses	Microwave and ultrasonic equipment
Arc/vapor lamps	Electronic ballasts
Pulse generators	Medical equipment
Thermostats	Microprocessor systems

and floors, interferences from ISM bands, and noise generated from equipment or heavy machines. It is important that data integrity is maintained despite the presence of noise and interference for mission-critical data, such as alarms. To ensure reliable delivery of messages from sensor to application host, changes in the signal level and radio connectivity need to be addressed carefully.

The interference signal can be classified as narrowband and broadband. Broadband interferences are signals with a constant energy spectrum over all frequencies that have high energy. They are usually unintentional radiating sources, whereas narrowband interferences are intentional noise sources with lower energy. Table.4.1 shows the narrowband and broadband interference sources that are generally found in the industries. The industrial environment is discussed in detail in Low et al. with case studies. The wireless technology and its deployment issues are elaborated in length in Willig et al.

Hence, filtering the sensor data is critically important. We have used wavelets as a preprocessor for the sensor data. The DWT is used as an entry-level filter.

4.12 Classifier with Wavelet Preprocessor

Wavelet transform is a tool that divides up data, functions, or operators into different frequency components and then studies each component with a resolution matched to its scale. Therefore, the wavelet transform is anticipated to provide economical and informative mathematical representation of many objects of interest. Each of the classifiers in the WSN can use wavelet preprocessors on each sensor signal, which is treated as one dimensional time series. Many wavelet transforms have been more or less complex.

We have chosen Haar 1D wavelet transform, which can be summarized as under if, $c_1(k) = \frac{1}{2}[c_0(k) + c_0(k-1)]$ is the smoothing of the signal, then $w_{1(k)} = c_1(k) - c_0(k)$ is the first wavelet coefficient. This can be generalized as: $w_i(k) = c_i(k) - c_{i-1}(k)$. The first smoothing $c_0(t) = x(t)$ is the signal itself and it is the finest scale.

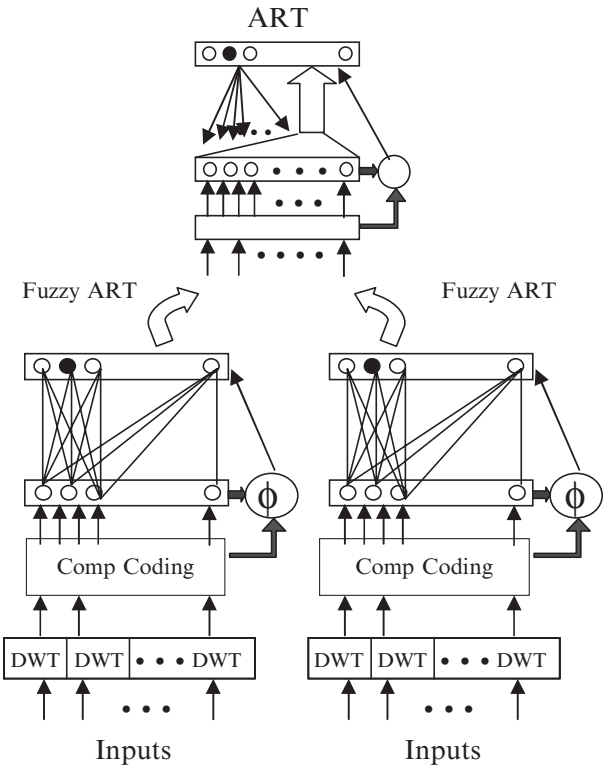


Fig. 4.6 Classifier with wavelet preprocessor

The combination of wavelet coefficients and the residual vector $c_p\{w_1(t), w_2(t), \ldots, w_n(t), c_p(t)\}$ can be taken as a multi resolution wavelet feature at time T of the signal, and it can be processed further as a representation of the signal itself. This discrete wavelet transform (DWT) acts as the entry layer for the classifier as shown in Fig. 4.6, which smoothes the signal for the classifier. The implementation of this classifier and different clustering architectures are discussed in detailed in Akojwar et al. (2006).

4.13 Computation of Energy Savings

With this architecture, a great dimensionality reduction can be achieved, depending on the number of sensor inputs in each unit. At the same time, communication savings benefit from the fact that the cluster number is a small binary number, unlike raw sensory readings that can be several bytes long—real numbers converted from analog inputs. If the number of sensors in each unit is s, the clusterhead collects data from h units, and the number of different categories in each unit can be

represented by b-byte integer, while the sensor readings are real numbers represented with p bytes, then the communication savings can be calculated as:

$$\frac{s.h.p}{h.b} = \frac{s.p}{b} \tag{4.10}$$

Because the communication is the biggest consumer of energy in the sensor node, this leads to bigger energy savings.

4.14 Conclusion

WSNs are highly data-driven and data-centric networks of huge sensor nodes, and have to handle heavy data traffic. The classification technique discussed in this chapter helps to reduce data traffic and, in turn, consumes less energy. Communication is the largest consumer of energy budgets in wireless sensor networks. Clustering sensor nodes in different architectures and then classifying sensor data at each node saves a considerable amount of energy. Energy savings can be of the order of number of classes of sensor data at individual nodes. The proposed WSN clustering architecture provides distributed data storage space for the network. These clustering schemes can be used as data aggregation models for WSN. It also improves the communications bandwidth of the entire network. The popular ART1 and Fuzzy ART algorithms are transformed to suites for VLSI implementation. The modified ART1 and FA are tested on MATLAB with different patterns of inputs. The three different types of clustering architectures will prove their importance in the paradigm of wireless sensor networks. The architectural strategy of ART1 and Fuzzy ART makes them easily suitable for VLSI implementation.

As depicted by Eq. (4.9), the communications savings are directly proportional to the number of classes at each individual node. When the number of classes is larger, the communications saving is better, and the energy consumption is lower. The number of classes should never exceed the value (s* p), and is depend on the application. Thus, classification helps WSN optimize its energy budget.

In industrial environment where noise predominantly exists, this classifier can be used with wavelet preprocessors. Discrete wavelet transform (DWT) is applied to the sensor signals before classifying them. DWT helps to smooth the raw data and also helps in extracting important features in the data like sudden changes in various scales.

References

1. Wendi A, Heinzelman B, Chandrakasan AP, and Balakrishnan H (2002) An Application–Specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transaction on Wireless Communications, vol. 1 no. 4 October
2. Guestrin C, Bodik P, Thibaux R, Paskin M, and Madden S (2004) Distributed Regression: an Efficient Framework for modelling sensor Network Data, in Proceedings of IPSN'04, April 26-27, Berkeley, California, USA

3. Kim CH, Park K, Fu J, and Elmarsi R (2005) Architecture for Streaming Data Processing in Sensor Networks, The 3rd ACS/IEEE International Conference on Computer Systems and Applications
4. Xu R, Student Member, IEEE and Wunsch II D, Fellow (2005) IEEE Survey of Clustering Algorithms, IEEE Transaction on Neural Networks, vol. 16 no. 3, May
5. Carpenter GA and Grossberg S (1992) A Self-Organising Neural Network For Supervised Learning, Recognition, and Prediction, IEEE Communications Magazine, Sep.
6. Serrano-Gotarrendona T and Linares-Barranco B (1997) An ART1 Microchip and its use in Multi-ART1 systems, IEEE Transaction on Neural Networks, vol. 8, no. 5, Sep.
7. Carpenter G, Grossberg S, and Rosen D (1991) "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," Neural Networks, vol. 4, pp. 759–771
8. Freeman JA and Skapura DM (2001) Neural networks Algorithms, applications, and programming techniques, Pearson Education Asia
9. Akojwar SG and Patrikar RM (2005) Neural networks for Wireless Sensor Networks, IMAPS India National Conference on Microelectronics and VLSI (IINC 2005), IIT Bombay, India, Dec. 19–21
10. Krishnamurthy L, Adler R, Buonadonna P, and Chhabra J et al. (2005) Design and Deployment of Industrial Sensor networks: Experiences from a Semiconductor Plant and the North Sea, SenSys'05, November 2–4, ACM
11. Akojwar SG and Patrikar RM (2007) Classification techniques for sensor data and clustering architecture for wireless sensor networks, IAENG International conference on communication systems and applications (ICCSA'07), IMECS2007, pp. 1246–1255, Hong Kong, March 21–23
12. Akojwar SG and Patrikar RM (2007) Real Time Classifier For Industrial Wireless Sensor Network Using Neural Networks with Wavelet Preprocessors, IEEE International conference on Industrial Technology, (ICIT 2006), Mumbai, INDIA. Dec. 15–17
13. Low KS, Win WNN, and Joo Er. (2005) Wireless Sensor Networks for Industrial Environments, IEEE CIMCA-IAWTIC
14. Willig A, Matheus K, and Wolisz A (2005) Wireless Technology in Industrial Networks, proceedings of the IEEE, vol. 93 no. 6, June

Chapter 5
PALMS: A Reliable and Incentive-Based P2P Live Media Streaming System

Poo Kuan Hoong and Hiroshi Matsuo

In recent years, the peer-to-peer (P2P) approach for media streaming has been studied extensively. In comparison with on-demand media streaming, P2P live media streaming faces a very stringent time constraint. To improve the performance metrics, such as startup delay, source-to-end delay, and playback continuity, we present PALMS—a P2P approach for live media streaming where a node employs gossip-based pull and push protocols to receive and forward media data among connected nodes. We present a simple heuristic mechanism for the pull protocol in the selection of media segments and peers. Besides the pull method, a push method is deployed to increase the streaming quality. We know that the presence of free-riders could degrade the delivered streaming quality. In PALMS, a simple score-based incentive mechanism, similar to BitTorrent's tit-for-tat incentive mechanism, is adopted to discourage the existence of free-riders. We conducted simulations and performance comparisons for PALMS. Experimental results demonstrate that PALMS can deliver better streaming quality and more resilience towards the heterogeneity of network bandwidths as compared to some of the existing protocols.

5.1 Introduction

In recent years, due to the popularity of the file-sharing phenomenon that leverages the power of the normal connected computers, peer-to-peer (P2P) networks have been the center of significant research. The P2P approach for media streaming has been studied extensively too in recent years by a number of authors [10,11,13,18,21] For P2P streaming, the basic building block behind P2P streaming system is the

Poo Kuan Hoong, Hiroshi Matsuo
Department of Electrical and Computer Engineering, Nagoya Institute of Technology
e-mail: khpoo@matlab.nitech.ac.jp,matsuo@nitech.ac.jp

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

nodes. Nodes can act both as clients and servers at the same time. Nodes form an overlay and simultaneously forward and receive media stream data to and from other connected nodes.

The focus of our paper is mainly on live streaming under the P2P unstructured overlay network. In live P2P streaming, the media stream is a continuous flow of media data encoded from the streaming server. Generated media content must be delivered to participating nodes under a tight time constraint. Nodes should be able to receive media content before the playout deadline or the media content will be discarded. Inspired by the file-swarming mechanisms, such as *BitTorrent* [7], participating nodes will swarm around newly generated media content. Newly generated media content is retrieved by receiver nodes and, in turn, actively contribute the media contents to other connected nodes. One good example of such application is the distribution of television programs from a single source broadcaster to a large number of dynamic and heterogeneous clients/end-users over the Internet. In this context, we present our approach to large scale P2P streaming for live and non interactive media streaming called PALMS or (acronym for *P2P Unstructured Live Media Streaming*). PALMS is based on data-driven and receiver-based overlay media streaming. It is designed to operate in conditions where nodes have heterogenous bandwidths and resources. Moreover, with PALMS we also address the issue of free-rider nodes. Similar to the *tit-for-tat* incentive policy used by BitTorrent, we introduce a simple score-based incentive mechanism to discourage the existence of free-riders in PALMS.

Although our approach is very similar to BitTorrent, the requirements for live streaming applications differ from normal P2P file sharing application. Live streaming deals with the transmission of practically infinite-sized media streams, which are to be played by the nodes upon reception before the playout deadline. Requirements for streaming to end users for P2P live media streaming applications include a) locate peers with the desired media segments before the playout time deadline, b) choose peers that are likely to provide good performance for playback, c) manage parallel download and upload to connected neighbor nodes, d) manage connected peers in the network due to the dynamicity of peers joining and leaving, and e) discourage the existence of selfish peers that choose only to download media contents without uploading.

5.2 Related Work

The existing P2P streaming systems can be roughly classified into three main families:

Structured: In these systems, participating peers are organized into a hierarchical tree structure to form an application overlay network. Media content is pushed from the source towards all peers. Differences between systems in this family concern the way nodes are organized and algorithms used to build and repair the tree structure. The fundamental limitations of these systems are a) the delivered

quality to individual peers is limited by the minimum bandwidth among the upstream connections from the source, b) a large fraction of outgoing bandwidth of peers that are leaves is not utilized. Some of the systems in this family are NICE [4], *End system Multicast* [6], PeerCast [8], and ZIGZAG [9].

Unstructured: The limitations of the structured family system have motivated a new approach where participating peers form a mesh-based overlay and incorporate a swarm-like content delivery. This approach is inspired by file-sharing protocol with swarm mechanisms such as BitTorrent [7]. Media content is broken by the source into chunks that are then available to the participating peers. Nodes independently request and download the chunks they need to complete the stream. Systems like Chainsaw [14] and DONet [21] have presented a mesh-based P2P streaming mechanism that incorporates swarm-like media content delivery.

Other: These systems do not belong specifically to one of the previous families. Most of these systems have hybrid architectures that combine features of structured control overlay to unstructured media content delivery. Examples of such system are Bullet [12], SplitStream [5].

Recent work in P2P systems concentrates on mitigating non cooperative behavior of peers by adding incentive schemes that discourage the existence of free-riders and encourage cooperation among nodes to share resources in order to increase the total capacity of the P2P system. The common incentive schemes are based on *Barter Economy* [2, 17] where peers offer their bandwidth to other peers according to the amount of bandwidth they get in return. Other approaches are game theoretic approaches, (such as [16]), and have been applied to incentives for parallel downloads—i.e., downloading a large file from several peers.

We extend the work in Zhang et al. (2005) by considering a combination of pull-pull methods, rather than pure pull methods for streaming mechanisms. This reduces the end-to-end delay. In addition, we propose a score-based incentive mechanism to encourage cooperation among participating nodes. Furthermore, we present a framework for the implementation of PALMS.

5.3 PALMS: System Overview

PALMS is based on data-driven and receiver-based unstructured overlay media streaming. It is designed to operate in scenarios where the nodes have heterogenous and variable bandwidth resources. For the ease of exposition, we refer to the media source as the *streaming server* and receivers as *clients*. The term *peers* and *nodes* are interchangeable, and refer to all the clients. Peers in PALMS implement data exchange policies that enforce and reward node cooperation through a score-based incentive mechanism. The incentive mechanism encourages cooperation among participating nodes and allows gradual improvement in the streaming of media content and, in turn, improves the delivered quality of streaming.

PALMS consists of three major components: a) **an overlay construction mechanism**, which organizes participating peers into an overlay; b) **a streaming**

scheduling mechanism, which determines the delivery of content from the streaming source to individual nodes through the overlay; and c) **an incentive mechanism**, which determines service received by a peer as a function of the bandwidth contributed by the peer. In the following subsections, we describe the components of PALMS.

5.3.1 Overlay Construction

In PALMS, nodes are functionally identical. They are free to exchange control information and media content data from the stream. Participating nodes form a randomly connected directed graph overlay network as shown in Fig. 5.1. Each peer maintains a certain number of connected nodes that are known as *neighbors*. Each node receives media content from certain number of parent nodes and relays the content to a certain number of child nodes. *Incoming* and *outgoing* degree denote the number of connected parent and child nodes, respectively.

The basic task of overlay construction mechanism components for each node is to be in charge of finding appropriate neighbors for each node through the gossip method so that the application layer network can be successfully built up. To join the streaming session, a new peer contacts the bootstrapping node, (streaming server in the case of PALMS) to learn about other participating peers upon arrival. This could be regarded as the *login process*. The bootstrapping node returns a list of randomly selected peers that can potentially serve as parent nodes. The new peer contacts these potential parent nodes to determine whether they are able to accommodate a new child node. This is by determining whether the parent node still has enough allocation slots on the outgoing degree. The peer also maintains a target incoming degree. If a new peer can not identify a sufficient number of parents from the reported list, it will make contact with the bootstrapping node again to obtain another

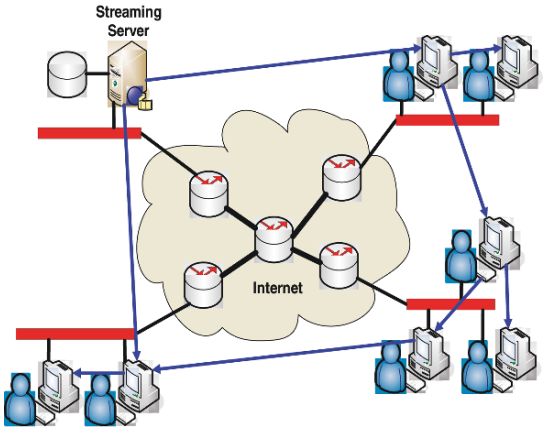


Fig. 5.1 The organized view of PALMS that consists of seven nodes

list of potential parent nodes. The successful new peer is registered with the bootstrapping node and can be selected as the parent node by other peers. Each new joined node synchronizes the local clock with the bootstrapping node during the login process.

All the nodes will self-organize into an unstructured mesh. Each node has a member table that contains a list of neighbor nodes obtained from the bootstrapping node. The information of member tables is encapsulated into a UDP packet and exchanged among neighbors periodically. Each node updates its member table in accordance with the member table sent by its neighbors. If a node does not update its neighbors periodically, it will be removed from the member table. Once a node leaves, it will broadcast a *leave message* to all its neighbors. The nodes that receive this message will delete the respective node from its member table as well. Therefore, the failure of any neighbors can be detected by constantly monitoring periodical messages from neighbors.

5.3.2 Streaming Scheduling

PALMS employs a swarm-like content delivery mechanism that is similar to BitTorrent [7]. The main advantages of the swarming content delivery are its ability to effectively utilize the outgoing bandwidth of participating peers and its robustness against the dynamics of peer arrival and departure, which is also known as *churn*.

The streaming scheduling mechanism of each node is responsible for exchanging packets with all its neighbors. The swarm-like content delivery is incorporated in PALMS. As a parent, each peer periodically generates reports—i.e., *buffer maps* of its newly received packets and sends to its child nodes. As a child, each peer periodically requests a subset of required packets from each parent based on the reports received. The pull mode is deployed to fetch absent packets from its parent nodes and, in turn, tries its best to deliver packets requested by the neighbors. Packets requested by the pull mode are determined by the packet scheduling algorithm, which is very similar to the data-driven approach in DONet [21]. Peer selection for PALMS depends on the rank ordering of score-based incentive mechanisms.

Every node also maintains a *window of interest*, which is the set of sequence packets that the node is interested in acquiring at the current time. Figure 5.2 illustrates the fundamental concept of the sliding window. A sliding *window of availability* contains the list of segments available for each node. This is the information for the *buffer map* shared with other neighbor nodes. The node slides its window of interest forward over time as new packets stream in. If a packet has not been received by the time it *falls off* the trailing edge of the window, the node will consider that packet lost or obsolete and will no longer try to acquire it.

To accommodate the bandwidth heterogeneity among peers, the content is encoded with *multiple description coding* (MDC). Basically, MDC organizes the streaming content into several sub streams where each sub stream can be independently decoded. The use of MDC for video streaming has been widely studied.

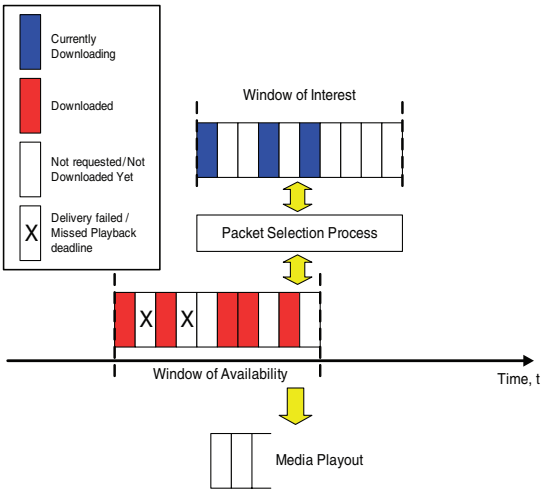


Fig. 5.2 Data buffer for PALMS node

Padmanabhan et al. propose that introducing redundancy can provide robustness in media streaming [13]. The delivered quality to each peer is proportional with the number of independent sub streams it receives. With MDC coding, each peer is able to receive the proper number of sub streams that are delivered through the combination of push-pull streaming mechanisms.

5.3.3 Analysis of the Pure Pull Method

We analyze the detailed process of the pure pull method to provide insight to related issues. Basically, the pull component in PALMS is similar to the data-driven approach in DONet.

In the pull mode of PALMS, when a packet goes from one node to another, three steps are executed as shown in Fig. 5.3. First, the sender receives packets from a connected neighbor and stores in buffer. The sender informs the receiver about the packet stored in buffer by sending a *buffer map* packet. Second, if the receiver needs this packet, a request is sent to the receiver. Third, the sender will deliver all the requested packets to the receiver. As depicted in Fig. 5.3, at least three end-to-end delays are involved in these steps. As a result, the delivery of most packets will have extra delays in one hop. We use $\delta_{buffermap}$ and $\delta_{request}$ to denote the waiting time. The total average latency for a packet transmitted in one hop T_{1hop} can be approximately computed as $\delta_{buffermap} + \delta_{request} + \beta\delta_{request} + 3\delta_{EED}$, where δ_{EED} is the average end-to-end delay between nodes.

The push-pull approach has two significant advantages: to a) reduce the end-to-end delay observed at the end user node, and b) improve the delivered media quality.

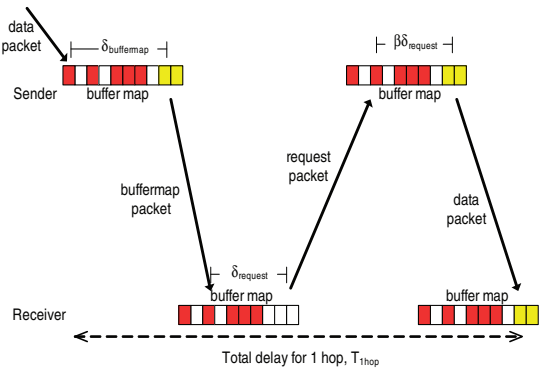


Fig. 5.3 One-hop delay using pure pull method

5.3.4 Incentive Mechanism

PALMS, just like any P2P content delivery system, works on the premise that peers share resources to increase the total capacity of the P2P system. In the case of PALMS, it derives bandwidth from the participating peers who operate independently of each other. A mechanism that creates favorable incentives for all peers—to contribute resources and thus guards against bandwidth starvation in PALMS—is needed to sustain peer interest in sharing bandwidth.

We believe that peer selection for receiving packets at child nodes offers a unique opportunity to tackle both the free-riders and the streaming Quality of Service (QoS) challenges in a synergistic manner. We propose a score-based incentive mechanism that provides service differentiation in peer selection for P2P streaming. Contributors to the system are rewarded with flexibility and choice in peer selection. Free-Riders are given limited options in peer selection, if any, and hence result in low-quality streaming.

We consider that PALMS consists of rational users who choose their contribution level to maximize their individual utility. The contribution level x_i of user i is converted to score S_i, which in turn mapps into a percentile rank R_i. The scoring function used in PALMS is based on the ratio of amount bytes uploaded over amount downloaded. Peer selection depends on the rank ordering of the requestors and candidate suppliers. For example, a peer selection scheme may allow a user to select peers with equal or lower rank to serve as suppliers. The outcome of the peer selection process is the realized quality of the streaming session. User utility U_i is a function of the streaming session quality Q and the contribution cost C:

$$U_i(x_i) = \alpha_i Q(x_i) - \beta_i C(x_i), \tag{5.1}$$

where α_i and β_i define the values of streaming quality and contribution cost to user i.

To quantify the performance of media streaming system, we define quality Q of a streaming session as:

$$Q = \frac{\sum_{i=1}^{T} V_i}{T} \tag{5.2}$$

where T is the number of packets in a streaming session and V_i is a variable that takes a value of 1 if packet i arrives at the receiver before its scheduled play-out time, and 0 otherwise. The quality is different from throughput, because it considers the deadline of each packet. The parameter Q captures other performance parameters such as packet delay, packet loss, and jitter.

Quality can be expressed as a function of contribution, score, or rank. The quality function is system dependent, but should exhibit the following properties: a) it is monotonically non decreasing in user score, b) the quality asymptotically reaches to the value Q_{MAX}, which represents the highest possible quality provided by the system; and c) the function has non negative initial value—i.e., $Q_{BE} = Q(S_i = 0) \geq 0$.

When a peer first joins the system, it begins with a score of zero and receives best-effort service $Q_{BE} = Q(S_i = 0) = 0$. The quality of this service may vary from system to system and vary as a function of system load. For example, a supplier node may choose to serve a node through a push method with a lower score only when it is idle. This best-effort service quality can be highly unpredictable. The only way to receive better quality than best-effort is to contribute to the system and earn a positive score.

The score is a discrete variable and thus the *probability density function* (PDF) is defined only where the score has a meaningful value. To compute the percentile rank, the *cumulative distribution function* (CDF) of the scores is calculated. The CDF is defined as:

$$F(S) = \sum_{i=S_{low}}^{S_{high}} f(i)$$

where f is the PDF of the score. The relationship between the percentile rank and the score is provided by CDF. The percentile is obtained by dividing the CDF with the total number of peers. The scores of all nodes are kept at the streaming server.

We would like to point out that in systems such as PALMS, that are time-sensitive traffic, the free-riders cannot afford to wait more time, because each packet has a certain lifetime. In other words, time-constrained data distribution provides stronger incentives to peers to discourage the existence of free-riders.

5.3.5 Peer Selection: Pull Mechanisms

The main algorithms used for peer selection for pull and push mechanisms are altruistic algorithms.

The algorithm for pull methods is similar to the heuristic used in DONet. Its purpose is to request the rarest packets among those that are locally available, and to distribute the request across different possible suppliers.

Using the information gathered from the buffer map exchanged among neighbor sets, packets that are rarest across the neighborhood are requested with higher priority than more common ones. Packets with the same number of suppliers are randomly requested from one of the neighbors that can provide them. This is to limit the load on any single peer. Figure 5.4 shows the algorithm for pull mechanisms.

Input:
$bw[k]$: bandwidth from neighbor k
num_suppliers : number of suppliers
$bm[i]$: buffer map of connected node i
$rank[i]$: percentile rank of connected node i
$deadline[j]$: deadline of packet j
expected_set : set of packets to be pulled
set_neighbors : number of neighbors of the node

Scheduling:
for packet $j \in expected\_set$ **do**
Make *num_suppliers* = 0
for $l \in \{1..set\_neighbors\}$
• Calculate T_j, Time for Transmitting packet j :
$T_j = deadline[j]$ - *current_time*
$num\_suppliers = num\_suppliers + bm[l,i]$
end for
end for
if *num_supplier*=1
• packets when there is only one potential supplier
for $j \in \{1..$expected_set$\}$
$supplier[j] \leftarrow \arg_r\{bm[r,i]=1\}$
end for j
else
• packets when there are more than one potential suppliers
for $j \in \{1..$expected_set$\}$
for $r \in \{1..$num_suppliers$\}$
• Find r with the highest $bw[r]$ and enough
available time $t[r,j]$
$supplier[j] \leftarrow \arg_r\{$ bw[r] > bw[r'],
t[r',j] > t[r,j],$rank[j] \leq rank[r]$, $r,r' \in set\_suppliers\}$
end for
end for
end if
Output $supplier[j]$
Do Pull packets from $supplier[j]$
Do Update *Buffer Map*

Fig. 5.4 Pull method heuristic algorithm

5.3.6 Peer Selection: Push Mechanisms

A good selection strategy is required to distribute the segments through the push protocol. It should ensure that every node pushes different segments to reduce redundancy. Finally, it should also take into account the requests from neighbor nodes.

Each neighbor node will try to allocate different segments into the *Push Segment Matrix*, *PSm* to be pushed. A simple roulette wheel selection scheme is applied for the next time interval for each node to push the available segments. Segments with

Input:
set_neighbors : number of neighbors of the node
$bm[i]$: buffer map of connected node i
$rank[i]$: percentile rank of connected node k
$deadline[k]$: deadline of packet k
expected_set : set of packets to be pushed

Scheduling:
for packet $k \in$ *expected_set* **do**
for $l \in \{1..set\_neighbors\}$
• Find Packet with the highest time-stamp :
$T_k = deadline[k]$ - *current_time*
end for
end for
• Roulette Wheel Selection for receiver
receiver's with higher $rank[i]$ are given higher probability
Output $receiver[k]$
Do Push packet to $receiver[k]$

Fig. 5.5 Push method algorithm

the highest time-stamp or *least sent* will be given higher priority to be allocated into the *Push Segment Matrix*, *PSm*. Each node keeps a count of how many times each segment is sent. The segment with the least number of times sent will be chosen. The push algorithm is shown in Fig. 5.5.

Let A denote the number of neighbor nodes, and *Seg* be the list of available segments to be pushed that each node possesses. Each node possesses a subset of *Seg*. The *Push Segment Matrix*, *PSm*, is represented by: $A \times Seg$. PSm^t denote the *PSm* at time t.

$$PSm^t = \begin{pmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,1 \\ 0\,0\,1\,0\,0\,0 \end{pmatrix}$$

5.4 Simulation Scenario

To evaluate our approach, we have conducted a simulation-based evaluation of the performance for PALMS. Our simulation implements the proposed pull and push protocol for PALMS that follows the algorithms described in the previous section. To evaluate the quality of media streaming, we implemented a simple MPEG4 video streaming system and run it on a networks simulator *(ns2)* [22] simulation

Table 5.1 Scenarios for comparing different upload bandwidth under PALMS

Upload B/W	SN 1	SN 2	SN 3	SN 4	SN 5
128 kbps	5%	15%	10%	50%	25%
256 kbps	10%	80%	10%	25%	25%
512 kbps	40%	5%	80%	25%	25%
0 kbps	45%	0%	0%	0%	25%

environment. Our streaming system is based on the Microsoft MPEG4 video encoding/decoding source codes. The streaming rate is set as 400 Kbps.

We used GT-ITM [19] to create a flat random graph as our underlying network topology. We vary the number of nodes participate in the streaming networks from 100, 200, 300, and 400, to 500 nodes. The average degree of the graphs is 5. We randomly select a few nodes to depart or join at different periods of time. Each link in the graph has a delay value, which is generated at random. To evaluate the performance of PALMS, we compare the proposed protocols of PALMS against two existing streaming protocols—Chainsaw [14] and PeerCast [20].

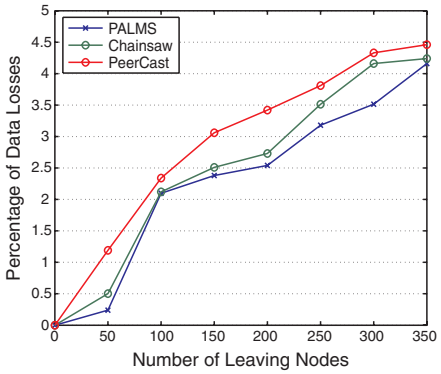
5.4.1 Evaluation Metrics

We evaluate the performances of the PALMS against two other existing P2P media streaming protocols using the metrics in Table 5.1.

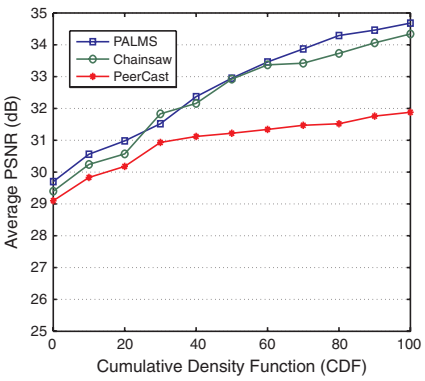
1. The peak signal-to-noise ratio (PSNR)
2. Percentage of data lost according to number of departure nodes
3. Comparison of time for the first segment to arrive
4. Comparison of overhead

5.5 Simulation Results

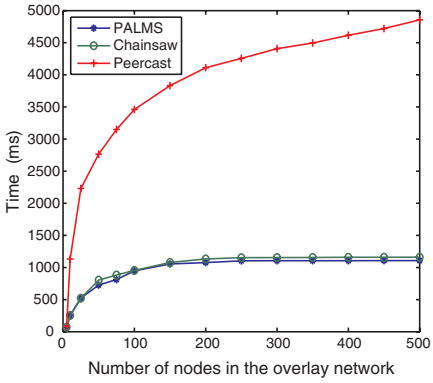
We compare the performances of PALMS against two existing protocols—a tree-based overlay multicast streaming protocol (PeerCast), and unstructured overlay multicast protocol (Chainsaw). PeerCast consists of end-system nodes on the Internet that execute multicast information dissemination applications. Each PeerCast peer acts both as client and server in terms of their roles in serving multicast requests. The PeerCast system uses the P2P membership protocol to organize the loosely coupled and widely distributed end-system nodes into a P2P network that carries the multicast service. The PeerCast peer-to-peer network is a distributed hash table (DHT)-based structured P2P network, while Chainsaw is a push-based randomly constructed graph streaming protocol. Each node in Chainsaw creates a list of desired segments and receives segments randomly from neighbor nodes.



(a) Data loses rate according to number of departure nodes



(b) Distribution of received quality based on PSNR



(c) Comparison of average time for the first segment to arrive

Fig. 5.6 Comparison simulation results

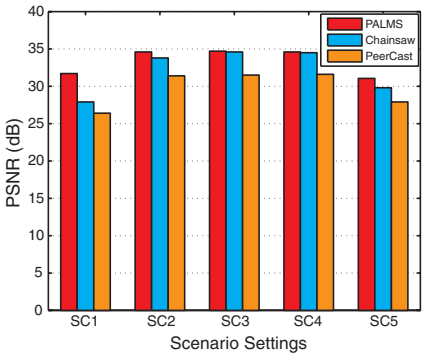
Figure 5.6(a) shows the result of comparison of percentage of data lost according to the number of departure nodes. The percentage of data lost for PALMS is smaller in comparison to PeerCast and Chainsaw. The simulation result shows that PALMS is much more robust in comparison to PeerCast and Chainsaw. PeerCast shows a higher percentage of segment loss as nodes are frequently redirected and reorganized in the tree structure during the departure of nodes. On the contrary, PALMS and Chainsaw do not require any restructuring because they consist of unstructured overlay networks. If a node leaves the network, new connections with other nodes can be established easily. With the node replacement policy adopted by PALMS, it's able to maintain a fixed number of connected neighbor nodes while discovering better connections. Moreover, with the combination of the pull and randomized push protocol, PALMS is able to reduce the percentage of segment loss during transmission.

Figure 5.6(b) depicts the averaged PSNR values for PALMS in comparison to PeerCast and Chainsaw. The peak signal-to-noise ratio (PSNR) is the measurement of the quality of received streamed video by the nodes in comparison to the original video quality. The streaming quality for live media streaming under PALMS shows better quality in comparison to PeerCast and Chainsaw. This is due to the fact that time is the main constraint for playout deadlines in live media streaming. Because PALMS behaves similarly to BitTorrent, nodes show the swarm-like effect towards newly generated media segments. Moreover, with the combination of pull and push protocol, media segments are selected heuristically and received before the playout deadline.

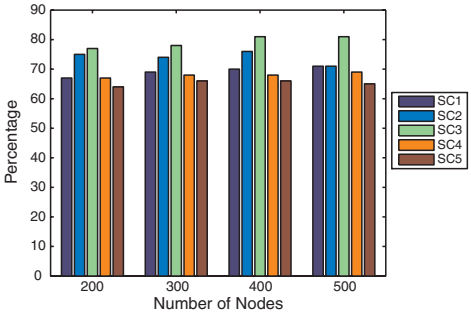
Figure 5.6(c) depicts the comparison of average time for the first media segment to arrive. As shown in the results, the average time for the first media segment to arrive for PeerCast increases logarithmically with the number of connected nodes. Primary because PeerCast is based on the overlay tree-structure, the more a node accepts new children, the more resources it shares among connected nodes. Evidently, it takes a longer time for the media segment to reach a child of PeerCast. Likewise, for the case of PALMS and Chainsaw, the unstructured overlay network approach is applied. PALMS and Chainsaw have a higher probability of forming connections with nodes with higher bandwidths and resources. Both PALMS and Chainsaw exhibit increments for its average time for the first media segment to arrive, but soon remain constant with the number of connected nodes. Based on the results, the average time for the first media segment to arrive is much shorter for PALMS and Chainsaw in comparison to PeerCast.

We also conducted simulations and comparisons on PALMS under several network scenarios to evaluate its performance. A comparison was made to evaluate the streaming quality for PALMS in comparison to PeerCast and Chainsaw. Streaming quality for PALMS depends on the cooperation of all connected nodes in the streaming network. The streaming quality is poor if the level of cooperation is low, even when the network is not heavily congested. Unlike traditional file sharing, live media streaming is bounded by strict time constraints. Thus, cooperation from only a few altruistic nodes cannot provide high-quality streaming to all other nodes in a large system. As a result, it is important to investigate the impact of free-riders on the overall streaming quality of PALMS, as well as PeerCast and Chainsaw.

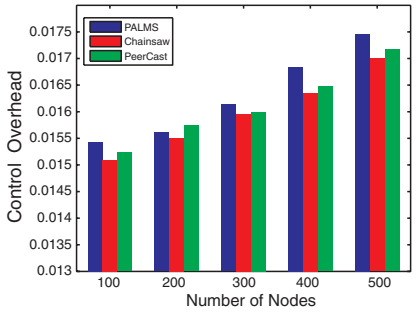
Figure 5.7(a) and Fig. 5.7(b) depict the result of PALMS under different network bandwidth scenarios. Parameters for each scenario are shown in Table 5.1. From the results, we can see the effect of free-riders toward the streaming quality of PALMS networks. Inevitably, free-riders decrease the streaming quality due to less cooperation among nodes for uploading received media segments to neighbor nodes. Cooperation among connected nodes is important for PALMS, because it is based on pull and push protocols. The increase of free-riders decreases the chances of media segments being pulled by receiver nodes. This, in turn, reduces the quality of the media streamed. However, in comparison to PeerCast and Chainsaw, the streaming quality of PALMS still shows a significantly better streaming quality. It is also interesting to point out that, based on different scenarios setting, PALMS exhibits the same streaming quality and is not affected by the heterogeneity of network bandwidths.



(a) Averaged PSNR for different scenarios



(b) Simulation results for different scenarios



(c) Comparison overhead for different number of nodes

Fig. 5.7 Comparison of simulation results

Figure 5.7(b) shows the percentage of media segments arrived within the playout deadline. It's expected that scenarios with a higher percentage of free-riders increases the media segment delay. Free-riders that are not willing to contribute their upload bandwidth will significantly affect the transmission of segments to all nodes. As a result, all the segments that are received outside of the playout deadline will be discarded. Nevertheless, with the score-based policy adopted by PALMS, the effect of free-riders is reduced.

Naturally, there is always a trade-off between failure resilience and maintenance overhead. In the context of PALMS, maintenance overhead involves the exchange of membership information among neighbor nodes as well as buffer content. Figure 5.7(c) depicts the comparison of overhead between PALMS, PeerCast, and Chainsaw. As expected, the overhead for PALMS is slightly higher. Nevertheless, the incurred overhead is still slow in comparison to the overall load for media segments.

5.6 Conclusions and Future Work

In this work, we presented a model for PALMS that incorporates the pull and push protocols. We provided a simple and efficient peer selection scheme for the pull and push algorithms. With the combination of pull and push methods, media segments can be retrieved efficiently and arrive before the playout deadline imposed by the live streaming constraint.

Leveraging the cooperation of all connected nodes, it is shown that PALMS experiences a significantly better streaming quality performance. Based on the simulation conducted, the results show that PALMS performs better in comparison to PeerCast and Chainsaw

We also examined and evaluated PALMS under different network scenarios. It is shown that the capacity of PALMS grows exponentially with the number of connected nodes. However, one of the main problem for P2P media streaming is the existence of free-riders. Free-riders consume resources and bandwidth from the system but do not provide any contribution. We adopted a simple score-based policy to discourage the existence of free-riders. The simulation results show that free-riders significantly reduce the streaming quality of other streaming protocols in comparison to PALMS.

For future work, we intend to improve on the packet-scheduling aspect, as well as implementation of network coding for PALMS. Finally, we also intend to implement and test PALMS on the PlanetLab Internet testbed.

References

1. Adar E and Huberman BA (2000) Free Riding on Gnutella. First Monday, 5(10), October
2. Agarwal S and Dube S (2006) Gossip Based Streaming with Incentives for Peer Collaboration. ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia, December, pp. 629–636
3. Akamai Technologies Inc, Delivery a better Internet. [Online] Available: http://www.akamai.com/
4. Banerjee S, Bhattacharjee B, and Kommareddy C (2002) Scalable Application layer multicast. SIGCOMM '02: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, August, pp. 205–217
5. Castro M, Druschel P, Kermarrec A-M, Nandi A, Rowstron A, and Singh A (2003) SplitStream: High-Bandwidth Multicast in Cooperative Environments. SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, Bolton Landing, NY. USA, October, pp. 298–313
6. Chu Y-H, Rao SG, and Zhang H (2000) A case for end system multicast. Measurement and Modeling of Computer Systems, June, pp. 1–12
7. Cohen B (2003) Incentives build robustness in BitTorrent. P2P Economics Workshop, Berkeley, CA
8. Deshpande H, Bawa M, and Garcia-Molina H (2001) Streaming live media over peers. Tech. Rep. 2001-31, CS Dept, Stanford University
9. Tran DA, Hua KA, and Do TT (2004) A Peer-to-Peer Architecture for Media Streaming. IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, January, pp. 121–133

10. Guo Y, Suh K, and Towsley D (2003) A Peer-to-Peer on-Demand Streaming Service and Its Performance Evaluation. ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo, Washington, DC. pp. 649–652
11. Hefeeda M, Habib A, Rotev R, Xu D, and Bhargava B (2003) PROMISE: peer-to-peer media streaming using CollectCast. MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia, Berkeley, CA, November, pp. 45–54
12. Kostic D, Rodrigues A, Albrecht J, and Vahdat A (2003) Bullet: high bandwidth data dissemination using an overlay mesh. SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, Bolton Landing, NY, October, pp. 282–297
13. Padmanabhan VN, Wang HJ, and Chou PA (2003) Resilient Peer-to-Peer Streaming. ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols, Atlanta, GA, November, pp. 16–27
14. Pai V, Kumar K, Tamilmani K, Sambamurthy V, and Mohr AE (2005) Chainsaw: Eliminating Trees from Overlay Multicast. IPTPS, no. 3640, pp. 127–140
15. Saroui S, Gummadi PK, and Gribble SD (2002) A measurement study of peer-to-peer file sharing systems. Proceedings of Multimedia Computing and Networking
16. Song J, Sha C, and Zhu H (2004) Nash Equilibria in Parallel Downloading with Multiple Clients. ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04), pp. 94–101
17. Sun Q and Garcia-Molina H (2004) SLIC : A selfish link based incentive mechanism for unstructured peer-to-peer network, Distributed Computing Systems, pp. 506–515
18. Ye S and Makedon F (2004) Collaboration-aware peer-to-peer media streaming. MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia, NY. pp. 412–415
19. Zegura EW (1996) GT-ITM: Georgia Tech Internetwork topology models (software). http://www.cc.gatech.edu/project
20. Zhang J, Liu L, Pu C, and Ammar M (2004) Reliable end system multicasting with a heterogeneous overlay network. *Technical Report GIT-CERCS-04-19*, CERCS, Georgia Institute of Technology, April
21. Zhang X, Liu J, Li B, and Yum TP (2005) CoolStreaming/DONet : A data-driven overlay network for efficient live media streaming. INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, March, pp. 2102–2111
22. Network Simulator, Ns-2. http://www.isi.edu/nsnam/ns/

Chapter 6
A Novel Dynamical Compression of XML Data for Transmission on Networks

Xu Huang and Dharmendra Sharma

6.1 Introduction

Extensible Markup Language (XML) [1] derives from SGML [2], which is an ISO standard (ISO 8879). It uses plain text to encode a hierarchical set of information using verbose tags to allow the XML document to be understood without any special reader. XML also allows the use of Schema documents [3] and Namespaces [4] to create a well defined contract describing what a single XML document means and to what it applies. The self-contained nature of XML and the strong contract provided by its schemas have made XML a very important data format for capturing information declaratively and semantically. It is used as a way to store and transport information, as a way of preparing data for publishing [5], translating to another language or structure [5] and even as a message layer between computer systems [6]. It is increasingly finding itself used in a number of different business markets, such as managing financial communications [7, 8]; and XML is often considered a useful tool for building internationalized business languages to allow better communication across vendors, platforms, and languages [9]. XML is often used on Internet and intranet facing webservers to request and deliver webservices used in wider applications. This is seeing wider adoption through the development of Enterprise Service Buses (ESBs) [10] to take advantage of the Service Orientated Architectures (SOAs) [11] that are being implemented in many organisations. To make matters worse, SOAs wrap XML document requests and responses within a Simple Object Access Protocol (SOAP) envelope describing what the XML document is for and detailing any errors that may have occurred during the SOA operation [6, 12, 13].

The devices using XML have also been expanding to include a broad range of newer devices, such as ubiquitous computing devices like Radio Frequency Identification (RFID) [14, 15] and Pocket PCs (PPC). These devices often use languages

Xu Huang and Dharmendra Sharma
School of Information Sciences and Engineering, University of Canberra, Canberra, ACT, 2617, Australia
e-mail: {Xu.Huang,DharmendraSharma}@canberra.edu.au

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

such as Java 2 Micro Edition (J2ME) [16] to parse XML [17]. This is used to communicate between the devices and the outside world, often over a wireless network connection. These devices often need to work in a real-time or near real-time manner, servicing a request as soon as it is made. An example of this would be with a RFID tagged stock item passing though a scanner.

XML is also being used to transfer information to wireless devices such as smartphones [18] using the eXtensible HTML Mobile Profile (XHTML Mobile), a combination of XML and HTML [19, 20]. This allows a single source of information to provide content for both a mobile platform, such as smartphones, and in a more traditional form, such as on a web browser in a PC.

XML is even being examined to determine if it can pass rich details regarding a user's access privileges to data sources, reducing the need for web based applications to look up a user's details in an Access Control, allowing for lower server overheads [21].

With all of the applications for which XML is being used, it is taking up an increasing volume of network traffic; this was expected to rise to about 25 percent of network traffic in 2006 from almost 0 percent in 2002 [22], as shown in Fig. 6.1.

The increase in XML traffic impacts the availability of the network to other forms of transmission. This will require some level of control of the XML document transmissions to ensure a certain Quality of Service (QoS) [23] for the whole network.

The increasing popularity of XML and the need to use it in near real-time processes means that we need to make sure the format is as efficient as possible. One of the greatest efficiency problems with XML documents is the size of the plain text structure they use, resulting in longer network transmission times.

There have been a number of methods suggested to reduce the size of XML documents. Examples of these methods include replacing the names of elements used in the XML document, such as with the XML Compression (ZXML) method [24]; using a binary representation of the XML document [25–27]; compressing the XML document [28–30]; and using some kind of middleware to control when an XML document is compressed [31, 32]. In this chapter we shall highlight our own methods and a middleware solution designed to control the compression of an XML document to be transmitted across a network [33–36].

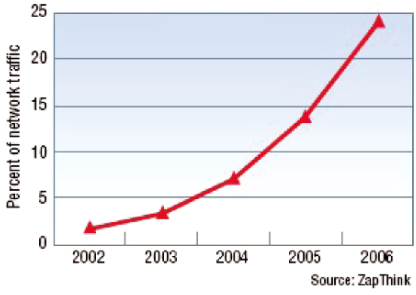


Fig. 6.1 Increase in XML usage

Therefore we find that on the one hand XML has become an increasingly important data standard for use in organizations as a way to transmit data [7, 8, 12, 13] and has also attracted the attention of people who are working in the areas of wireless communications, in particular for so-called small wireless devices. Additionally XML is being used to enable web services and similar, often custom, RPC functionality to allow greater access to data across multiple systems within an organization and allowing the possibility of future systems to be created from collections of such RPC functionality.

However, on the other hand, XML is a verbose, text based format with strict requirements on structure and is often criticized for its large space requirements. This large size can be particularly problematic for use in transmission across a network, where network bandwidth restrictions can cause significant delays in receiving the transmission, which has drawn great attention from users of wireless communications.

One solution to this problem is to look at reducing the size of these transmissions by rendering them in a binary format, such as by using XMill or Gzip to compress an XML document. However, such methods can take longer as compressing and decompressing may take more time than what is saved transmitting the smaller XML document.

We were motivated to find a way to improve the inefficient size of XML documents used to transmit information across a network. We wanted to determine an efficient method of controlling the size of XML documents that are transmitted across a network to reduce the total transmission time. We also wanted to develop a systematic method of controlling, with our conditions, the XML document when it is to be transmitted.

We started by investigating the different means of controlling how XML documents are transmitted across a network. This involved a literature review to determine what work has already been done in this area. The next objective was to determine if there are any alternative methods to those already examined that could be used to control how XML documents behave. Having determined a possible new method, our third objective was to develop a program to implement it in a network. Once the program was developed it was tested in several stages, using simple tests to see if our method holds any value, seeing how it works in an (assumed) unchanging network environment, how it works in a simple network environment where the program is allowed to react to changes in the network, and finally using long-running tests in a simple network environment. The program will be compared to an existing popular method to see if our method offers any benefits.

While this study looks at the use of compression to reduce the transmission time of XML documents, only a few methods of compression, such as GZip and Xmill, were considered at this stage; and this study did not consider in depth the effect of using a tweaked compression algorithm to better reduce the size of the XML document. Another limitation in the study is that only a simple network consisting of two computers, a server, and a client, connected by a router, was used to test how well the methods discussed perform. While this simple network had its characteristics

changed by varying its activities and the NIC throughputs, it does not show the full range of network loadings that are expected in a business network.

One solution to this problem is to look at reducing the size of these transmissions by rendering them in a binary format, such as by using XMill or Gzip to compress an XML document. However, such methods can take longer, as compressing and decompressing may take more time than what is saved transmitting the smaller XML document.

Another solution to this problem may be the Network Adaptable Middleware (NAM) proposed by Ghandeharizadeh et al [32], even though there are some ways to directly compress, such as column-wise compression and row-wise compression for large message sizes [42]. This solution estimates the time it will take to compress, transmit in binary format, and decompress a document compared to an estimate of how long it would take to transmit the document as uncompressed text. The estimates are based on a persistent collection of information on how the system has performed in the past and provides an accurate estimate on whether or not it would be faster to compress the document before transmission.

We have introduced another way of determining when to compress an XML document before transmitting it, in our One Pass Technique (OPT) and have extended OPT to Dynamic Adaptive Threshold Transmission (DATT) [33–45]. In this technique we determine a threshold size value for the networks. Any XML document that is smaller than this threshold will be sent uncompressed while any XML document larger than this size will be compressed before it is sent.

Performance on networks depends on various parameters, such as traffic situations, bandwidths, transferring rates, etc. We shall use the adapted dynamic threshold to represent the characteristics of the running networks, by which the transferring XML data on networks will be controlled under optimum conditions in terms of transferring decision time defined in the next sections. The following sections are as follows: in Section 2, we shall briefly review the established OPT technique and show that it is possible to improve it. In Section 3 the Dynamic Adaptive Threshold Transmission (DATT) for XML data on networks will be demonstrated, together with the experimental setup design, which is the natural research project, extended from the previous research results. We shall present the conclusion of this chapter in Section 4; in particular, we are going to highlight the “dynamic” issue as the extended part of our previous paper.

6.2 Threshold Method and Its Application

Because we shall focus on the behavior of networks for the transmission of XML documents, we first need to examine a number of network characteristics. These characteristics are described here.

When we focused on connectivity on a network we were interested in how XML is transmitted across a network. More specifically, given XML’s use over Internet

connections, we were interested in how XML is transmitted across a TCP/IP [39] connection. It is well known that the most important parameters of any network are its bandwidth, throughput, buffer size, and latency, which are also used to assess a network.

TCP/IP splits the data being sent across the network into a series of data packets. These packets are then transmitted across the network separately. TCP/IP uses a congestion window to determine how many packets it is able to send at once across the network. This congestion window starts with a low transmission rate of one segment, as described by the receiver's Maximum Segment Size (MSS) [39]. This rate is then doubled each time the packets are successfully sent until the transmission rate reaches the receiver's maximum allowable rate, that is, until the data being transmitted fills the receiver's buffer size, though obviously the sender's buffer size will need to be large enough to contain the data to be transmitted or network congestion will cause it to stop increasing.

The network's bandwidth limit lists its maximum theoretical throughput. An Ethernet connection rated at 100 Mbps could theoretically move data at 100 Mbps. The real-world speeds are, however, often much lower than this theoretical limit.

The latency of a network describes how long a data packet takes to move from one end of the network to the other or, more formally, it is the time between initiating a request for data and the beginning of the actual data transfer. In a network, this occurs as a data packet is stored, analyzed, and transferred [40].

A network's throughput describes how much data can be passed across a network per unit of time. For a TCP/IP network this is a function of the buffer size and the network latency, or in other words the maximum throughput is defined as [41, 42]:

$$Throughput_{max} = buffer\ size/latency. \tag{6.1}$$

Network congestion and other factors may lead to individual data packets not arriving at the receiving destination. This requires the lost packets to be retransmitted and leads to a real-world measured throughput that is lower than the maximum value calculated here.

It is well known that XML has been actively involved in Business to Business (B2B) processes. We briefly examine B2B in this section.

B2B transactions are ones that take place between separate businesses or enterprises. These transactions may include purchasing or the offering of services. These types of interactions are becoming increasingly important in real-world applications. ESBs are often deployed in organizations to allow services to be called both internally and from external organizations, and these interactions need to be efficient.

In order to allow one business to talk to another business to request a product or a service they need to have a common language. XML is rapidly becoming the default language used in these situations [43].

XML schemas [30] describe how a service may be called and what it's response will look like. For example, one business may send an XML document to another business that complies with the agreed schema to perform a search on client data

held by the second business. The second business then sends the client data back to the first as XML complying with the results schema. This means that the first business is able to perform a search and read results from a service owned by the second business.

As these B2B services can potentially carry large volumes of information, the XML documents they transmit can be very large. As an example, a collection of data on a company's available products can consume hundreds of kilobytes, or even several megabytes in an XML document. This becomes one of the major problems in using XML-driven applications in B2B.

Before we establish our Dynamic Adaptive Threshold Transmission (DATT) for XML data on networks, we need briefly to recall our previous method, titled "One Pass Technique" (OPT) and show that OPT needs to be changed if we want it work well on a network. Then we will extend our previous results to current DATT in the next section.

In contrast to the five network factors that contribute to the latency time of delivering a query output [31] based on the analysis of the one-gigabyte TPC-H benchmark [43], our method presented here utilizes an established "threshold" for the current working status and then has a "one-pass" transmission. We defined a threshold value for the network such that the transmitted time, for XML documents whose size has been compressed (such as via Gzip) and then uncompressed, will be comparable. To determine what this value could be, we first need to determine the network's characteristics. As those characteristics will evolve with time, the threshold value needs to dynamically change with the network.

Before OPT can be used on a network we need to determine the threshold value by making a number of XML transfers of different sizes across the network. The transmissions need to be made both with the document compressed, using Gzip as an example, (and decompressed where it is received) and by transmitting the document without compression. An estimate of how long it takes to transmit a document of a given size can then be determined by curve fitting to these results. The threshold value is set when the estimated time to transmit it without compression is equal to the estimated time to transmit it with compression. In some situations this may result in a threshold value that will require compression of all documents or one that will never require compression of a document.

There are a number of factors that can prevent OPT from yielding the best result for all cases. The threshold value will only be valid for the network bandwidth it is calculated for, so if that bandwidth changes, a threshold value will give an inaccurate result and a new threshold value will need to be determined.

The compression and decompression times are dependent on the CPU load. If the load on a CPU is heavier (or lighter) than it was when calculating the threshold value, the OPT may not make the appropriate decision on whether or not to use compression on the XML document. Similarly the technique works best with a homogenous set of CPUs. Different CPUs will take different time periods to compress and decompress the XML document. The compression/decompression time of two low-end CPUs on a network will be different from the compression/decompression

time of two high-end CPUs on the same network using the same threshold value. This can also lead to the OPT making a wrong decision on whether or not to compress the document.

The OPT can also be affected by changes in the network's traffic density. If the network is under a heavier load than it was when the threshold value was calculated, the technique is more likely to transmit an uncompressed XML document when a compressed document would have been faster; and with a lighter network load, compressed XML transmissions are more likely to occur when an uncompressed transmission would have been faster. OPT is best used in a homogenous environment where the network bandwidth is well known and network traffic is reasonably stable.

Since a threshold depends on many factors on the network, if OPT works for a network, it must be changed from time to time, depending on the current status of the network; namely, it must be dynamically changed to control the transfer rate on the network. This is the basic idea of our Dynamic Adaptive Threshold Transmission (DATT) for XML data on networks.

6.3 XML Data on Networks and Empirical Results

It's noted that there is another solution to this problem, which is called Network Adaptable Middleware (NAM) [31, 32]. This solution estimates the time that it will take for the whole process to compress, transmit in binary format, and decompress a document, compared to the time that it would take to transmit the document without compressing it. These estimates are based on a persistent collection of information on how the system has performed in the past and provides a reasonably accurate estimate of whether it would be faster to compress the document before transmission or not.

Whenever NAM transmits an XML document, it uses records of the times taken for different events to estimate how long such events are expected to take for the current XML document. NAM then decides if it should send the XML document compressed as a result of the comparison, which can be expressed as:

$$\begin{aligned} &\textit{If } t_{\textit{Uncompressed Transmission}} > t_{\textit{Document Compression}} + t_{\textit{Compressed Transmission}} \\ &\qquad + t_{\textit{Document Decompression}} \\ &\textit{Then transmit\_compressed}, \\ &\textit{Else transmit\_uncompressed}. \end{aligned} \tag{6.2}$$

where t is the time.

So if the time taken to transmit the XML document uncompressed is less than the time taken to compress, transmit, and then decompress the document, the XML document will be sent uncompressed.

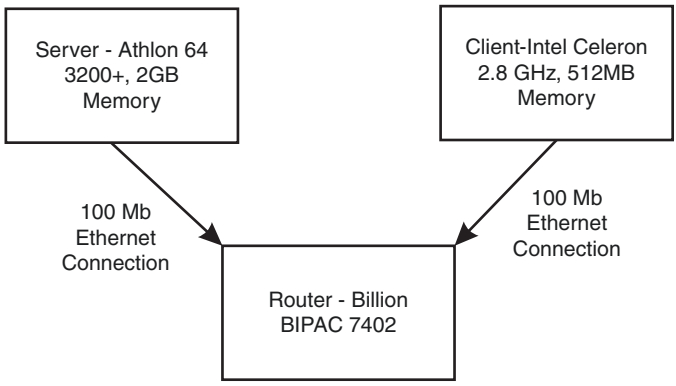


Fig. 6.2 The experimental setup diagram for DATT

As the estimate must be calculated whenever a document is to be sent, NAM can spend a significant amount of time in determining if the document should be compressed or not. Additionally, as the estimates are based on the size of the XML document being transmitted, NAM assumes that when two different XML documents with the same physical size in bytes are compressed the result will also have the same physical size.

In order to have a Dynamic Adaptive Threshold Transmission (DATT) for XML data on networks, we make a programmed process to check the current network working situation that depends on the current traffic parameters discussed in the above sections. Then the current threshold is worked out based on the same principle described for OPT in Section 2. The obtained threshold will replace the previous one to work (control) the traffic communications. Since the threshold is monitored dynamically, the adaptive threshold will always keep a record of the times taken in transferring the data.

In order to investigate our Dynamic Adaptive Threshold Transmission (DATT) for XML data on networks, we design our experimental work as shown in Fig. 6.2.

The connection was made across the router using a raw TCP connection created for each transmission. The client used a listener to listen for the incoming files (port 9013). The server was running a Cron style task scheduler to initiate communication and deliver the file.

A number of XML documents (1200 files) were gathered to test using a time-based threshold as shown in Fig. 6.3 to decide on when to compress a document and when not to. These files were of different sizes. An application program was written to transmit these documents a number of times across a network using a threshold value. Any XML document with a size greater than the threshold value was transmitted compressed while all other XML documents were sent uncompressed. The algorithm used is:

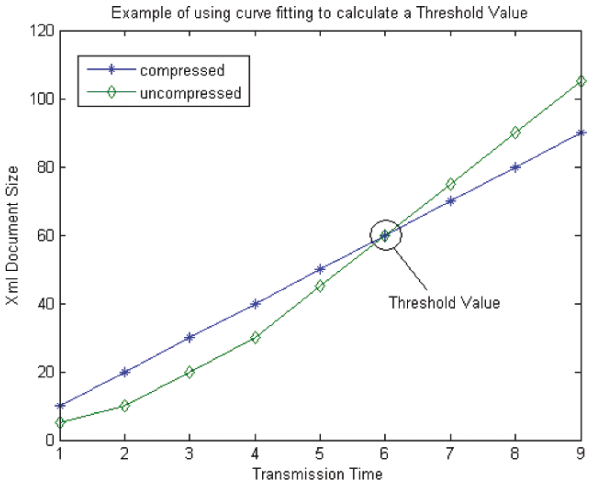


Fig. 6.3 Curve fitting example sets of uncompressed and compressed transmission data to determine the threshold value

Hence we have the protocol as shown below:

$$\textit{If } \text{Size}_{\text{Document}} > \text{Size}_{\text{Threshold}} \textit{ Then } \text{transmit\_compressed},$$
$$\textit{Else } \text{transmit\_uncompressed}$$

6.4 Experimental Setup

We created a number of SOAP XML documents containing data that may be found in a typical data load, starting with a SOAP document that contained an empty body. We then created a number of SOAP XML documents that contained a simple content structure.

This content structure was used to describe SOAP documents that contained one, two, ten, twenty, fifty, or one hundred instances of this simple structure. The empty SOAP XML document and the SOAP XML document with one instance of the simple data structure are listed in Figs. 6.4 and 6.5, respectively.

We then recreated these SOAP XML documents as simple comma-separated value (CSV) files holding the same information as shown in Fig. 6.6.

As the SOAP XML documents and the CSV files hold the same information it is easy to compare their sizes when they are uncompressed and when they are compressed using the commercially available compression tools WinZip [28] and WinRAR [29]. This will give an idea of what benefits may be attained by passing XML documents compressed and show if it is possible to get the transmission size of XML documents near to the size of other, less structured text formats.

```
<?XML version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  XMLns:SOAP-ENV="http://schemas.XMLsoap.org/soap/envelope/"
  XMLns:SOAP-ENC="http://schemas.XMLsoap.org/soap/encoding/"
  XMLns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body                                                SOAP-
ENV:encodingStyle="http://schemas.XMLsoap.org/soap/encoding/">
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 6.4 An empty SOAP XML document

```
<?XML version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  XMLns:SOAP-ENV="http://schemas.XMLsoap.org/soap/envelope/"
  XMLns:SOAP-ENC="http://schemas.XMLsoap.org/soap/encoding/"
  XMLns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body                                                SOAP-
ENV:encodingStyle="http://schemas.XMLsoap.org/soap/encoding/">
    <ApplicationMessages>
        <ApplicationMessage>
            <Details>
                <Name>
                    Someone
                </Name>
                <Date>
                    01/01/2005
                </Date>
                <AnotherName>
                    Someone Else
                </AnotherName>
                <AnotherDate>
                    23/10/1945
                </AnotherDate>
            </Details>
            <Information>
               <Text>
                  This is some text
               </Text>
               <IntegerData>
                  9
               </IntegerData>
               <FloatingData>
                    4.678
                </FloatingData>
                <Description>
                    This is a descriptional field
                </Description>
                <AnotherInteger>
                    413
                </AnotherInteger>
                <MoreDescriptiveText>
                    This is another descriptional field
                </MoreDescriptiveText>
            </Information>
        </ApplicationMessage>
    </ApplicationMessages>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 6.5 A SOAP XML document with a single entry

```
Someone,01/01/2005,Someone      Else,23/10/1945,This      is      some
text,9,4.678,This is a descriptional field,413,This is another de-
scriptional field
```

Fig. 6.6 A CSV file holding a single entry

6.5 Results

We carried out the experiments with the SOAP XML documents and the CSV files. Table 6.1 shows the results seen when compressing an empty SOAP message.

Table 6.2 shows the compression of a SOAP message that contains a single entry. Compressing the SOAP message results in a size reduction of about 50 percent, but it is still significantly larger than the CSV file.

Table 6.3 shows the compression of a SOAP message with two entries. With two data entries the size cost saving is up to 30 percent. The CSV file is still significantly smaller, particularly when similarly compressed.

With the ten data entries seen in Table 6.4 the size cost saving is now around 18 percent. The CSV file is actually larger than the compressed SOAP file and the compressed CSV files offer only a small size cost improvement.

Table 6.1 Size of an empty SOAP message

	Uncompressed file	WinZip compressed	WinRAR compressed
SOAP	363 bytes	311 bytes	279 bytes

Table 6.2 Size of a SOAP message with one entry

	Uncompressed file	WinZip compressed	WinRAR compressed
SOAP	1,136 bytes	523 bytes	564 bytes
SOAP, per data entry	773 bytes	212 bytes	285 bytes
CSV	138 bytes	221 bytes	187 bytes

Table 6.3 Size of a SOAP message with two entries

	Uncompressed file	WinZip compressed	WinRAR compressed
SOAP	1,878 bytes	622 bytes	574 bytes
SOAP, per data entry	758 bytes	156 bytes	148 bytes
CSV	298 bytes	271 bytes	240 bytes
CSV, per data entry	379 bytes	136 bytes	120 bytes

Table 6.4 Size of a SOAP message with ten entries

	Uncompressed file	WinZip compressed	WinRAR compressed
SOAP	8,384 bytes	1,484 bytes	1,358 bytes
SOAP, per data entry	802 bytes	117 bytes	108 bytes
CSV	2,153 bytes	933 bytes	923 bytes
CSV, per data entry	215 bytes	93 bytes	92 bytes

The twenty data entries shown in Table 6.5 show the size cost saving is now 14 percent and there is decreased difference between the compressed SOAP document compared to the compressed CSV file.

Table 6.6 shows the results of a SOAP message with fifty entries. The size cost saving by compressing the SOAP message is down to 11 percent of the uncompressed size. The compressed SOAP document is now only about 12 percent larger than the compressed CSV file.

The results of one hundred data entries in a SOAP message are shown in Table 6.7. The size cost saving is still about 11 percent of the uncompressed SOAP message size. The compressed SOAP document is now only about 9 percent larger than the compressed CSV file.

All the observations show that the efficiency of the compression applied to an XML document improves as more data instances are added. This would be expected, given that the structured nature of the XML format means that each data instance would include the same opening and closing element tags. What was interesting is how the compressed SOAP XML documents began to approach the size

Table 6.5 Size of a SOAP message with twenty entries

	Uncompressed file	WinZip compressed	WinRAR compressed
SOAP	16,623 bytes	2,352 bytes	2,175 bytes
SOAP, per data entry	813 bytes	102 bytes	95 bytes
CSV	4,572 bytes	1,667 bytes	1,654 bytes
CSV, per data entry	229 bytes	83 bytes	83 bytes

Table 6.6 Size of a SOAP message with fifty entries

	Uncompressed file	WinZip compressed	WinRAR compressed
SOAP	41,426 bytes	4,742 bytes	4,515 bytes
SOAP, per data entry	821 bytes	89 bytes	84 bytes
CSV	11,915 bytes	3,853 bytes	3,851 bytes
CSV, per data entry	238 bytes	77 bytes	77 bytes

Table 6.7 Size of a SOAP message with one hundred entries

	Uncompressed file	WinZip compressed	WinRAR compressed
SOAP	83,320 bytes	8,971 bytes	8,798 bytes
SOAP, per data entry	830 bytes	87 bytes	85 bytes
CSV	24,708 bytes	7,903 bytes	7,900 bytes
CSV, per data entry	247 bytes	79 bytes	79 bytes

of the compressed CSV files. As more data instances are added to the SOAP XML document and to the CSV file, the difference in their compressed sizes becomes negligible. The SOAP XML documents' richly structured format would make this a preferable format for describing large sets of data.

A similar application was set up to transfer the documents using the NAM methodology (Ghandehazrizadeh, 2003). NAM uses measured network and computer characteristics to compare estimates on how long it would take to transmit an uncompressed document against an estimate of how long it would take to transmit a compressed document. The algorithm used is:

$$\begin{aligned} &\textit{If } \mathrm{Time}_{\text{Uncompressed Transmission}} > \mathrm{Time}_{\text{Document Compression}} \\ &\quad + \mathrm{Time}_{\text{Compressed Transmission}} + \mathrm{Time}_{\text{Document Decompression}} \\ &\textit{Then } \text{transmit\_compressed}, \\ &\textit{Else } \text{transmit\_uncompressed}. \end{aligned}$$

The experiment was conducted using a client PC (754pin Athlon64 3200+ @2.05GHz with 1GB RAM), one Server PC (Celeron D 2.8@2.79GHz with 512MB RAM) connected by a Router (Billion BIPAC 7402G) over a 100 MBit Ethernet connection.

For the DATT the time taken is calculated by the following:

$$\begin{aligned} &\textit{calculation time} + \textit{compression time} + \textit{transfer time} + \textit{decompression time} \\ &\quad + \textit{threshold calculation time} \end{aligned}$$

In order to obtain good statistics and fairly distributed results, a set of twenty-nine runs were carried out for each technique, namely DATT and NAM, sandwiched for one hour. For such a setup the whole running process covers more than 41 hours without breaking.

In order to change the working environment from time to time, the network was disturbed while it was processing various activities, such as downloading files of different sizes, browsing the Internet, playing audio on the computer, etc. In order to determine the characteristics of the network before the applications, these procedures were run: solving the quadratic equations used to get the time and size estimates NAM uses in its decision algorithm, and determining the threshold value for the current network traffic load for the OPT. When the threshold value was found at the particular time it will be used for controlling the XML data transferred on the network.

The average results for the decision time for DATT are shown in Fig. 6.7. Since the process of determining the threshold has been passed out to a separate process, the decision times for DATT are very short.

All the running results were recorded in fivecategories, namely, the original file size in bytes, the decision time (for DATT and NAM, that is the time deciding whether the current file should be compressed and sent or just sent, according to the principle of DATT or NAM, respectively), compression time, transferring time, and decompression time.

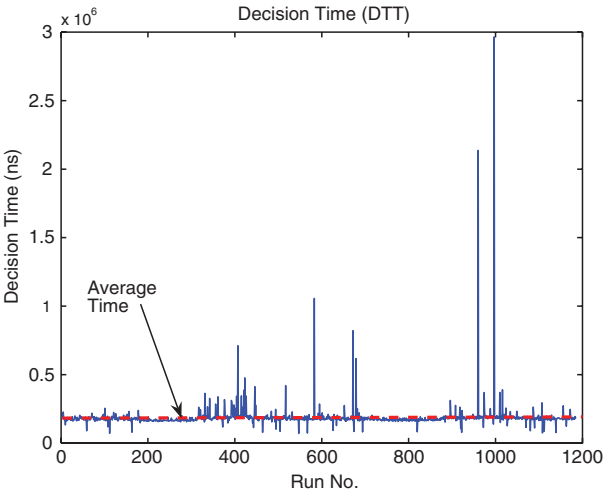


Fig. 6.7 The experimental results for the average decision time of the DATT described in Section 3

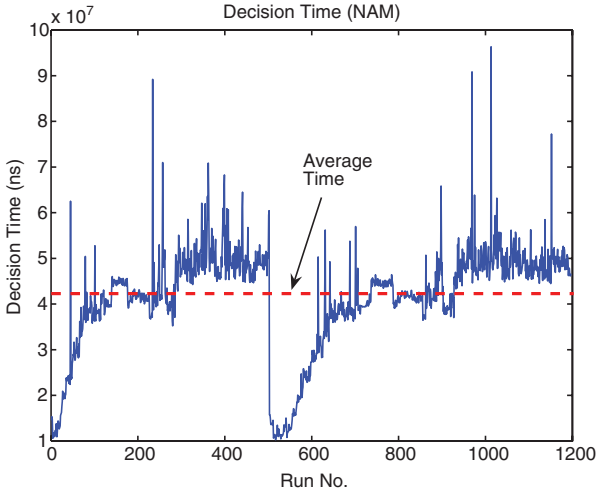


Fig. 6.8 The experimental results for the average decision time of the NAM described in Section 3

The average results for the decision time for NAM are shown in Fig. 6.8. It is seen that, as NAM accumulates more data from which to make a decision, the time it takes to actually make the decision increases to allow all the data to be read.

In both diagrams, the average decision times are marked as dashed lines.

It is important to highlight two items. One is that the run numbers for the two figures are about 1200, which are the "average" results from 41 hours running as described in above. The other is that the plotted run order should not be mean-ness due to the fact that we put two results, compressed files and uncompressed files,

together. In other words, when the input file is picked up and sent to be judged either by the DATT or the NAM, the output will be sent to two groups: one is compressed (if the file was compressed and then sent) another one is uncompressed (if the file was not compressed and directly sent). Because we care about the average time taken by the decision rather than when the decision was made, for plotting purposes we just put them together. Also one may find the plotting results seem to be "periodic" results, in particular for Fig. 6.8; however, these are not real time results.

In these experiments, the average decision time for DATT is 0.187622 milliseconds and for NAM it is 41.381926 milliseconds, which means NAM takes 220.56 times longer than DATT to make a decision.

It is also very interesting to note that the experimental results show that the number of compressed files with DATT is 587 out of 1201 running files, which gives a compression ratio of 0.488. In contrast to DATT, NAM has 510 files compressed out of 1219 running files, which gives a compression ratio of 0.418. Therefore, the compression ratio of NAM is about 86.4 percent that of DATT. This shows that, in comparison to DATT, NAM is always (or on average) making cautious decisions to keep itself in optimum states but causes heavier network traffic, which means that DATT will make higher quality network transfers for XML data on networks. This improvement of DATT compared to NAM is about 0.25 percent.

In terms of the performances of both the DATT and NAM techniques in this experimental setup, both used the same software for compressing and decompressing, but the processing was different due to the different principles used by each technique. Thus, the data shows that, on average, the compressing time and decompressing time are about balanced, since the time taken by DATT divided by the time taken by NAM is 0.91 for compressing and 0.895 for decompressing. However, the DATT technique is a more stable process for compressing, as shown by the fact that the standard deviation of the compressing time is almost ten times smaller than that of NAM. Even the standard deviation of the decompressing time is fairly comparable, which is understandable since for DATT, when the threshold is obtained, the rest of the job is much easier than for NAM. This is further evidence to show that the DATT technique will allow running networks to provide a better quality of service.

One of the questions that needs to be answered is how frequent the period time taken for DATT for the nature of "dynamic" issue is. Since the total time taken for DATT and NAM varies depending on the traffic situations in the networks. For example when the traffic is busy the DATT will take more time than that for NAM since the "threshold" calculation together with the comparisons then decision made will be dearer than that for NAM. In contrast the traffic situation is reasonable relaxant the time costs for DATT will be less than that for NAM due to the "threshold" calculation and comparisons will be easier. We take the traffic distributions as Poisson distributions and from the PDM of traffic distribution from 8:00 am to 8:00 pm as "effective" distribution as the common cases. The data shows that when the traffic ranges between 40 percent of the peck the DATT will be suggested to use otherwise the NAM will be suggested.

6.6 Conclusion

We have extended our previous research results to Dynamic Adaptive Threshold Transmission (DATT) for XML data on networks. We compared this technique (DATT) to another control technique, the Network Adaptable Middleware (NAM), and found that the DATT technique is much better than NAM in terms of decision time taken, which was about 220.56 times faster for the DATT than for the NAM. Also DATT will give running networks better performance by as much as one quarter in comparison with NAM. The simulation results suggest that in real life the two methods may be combined, giving even better results.

References

1. *http://www.w3.org/XML/*
2. *http://www.sgmlsource.com/8879/index.htm*
3. *http://www.w3.org/XML/Schema, W3C XML Schema*
4. *http://www.w3.org/TR/REC-xml-names/*
5. *http://www.w3.org/Style/XSL/*
6. *http://www.w3.org/TR/soap12*, SOAP version 1.2
7. Fan M, Stallaert J, Whinston AB (2000) The Internet and the future of financial markets. Communications of the ACM 43(11):83–88
8. Rabhi FA, Benatallah B (2002) An Integrated service architecture for managing capital market systems. IEEE Network 16(1):15–19
9. *http://www.xml.com/pub/a/2003/05/21/xmleurope.html*
10. *http://www306.ibm.com/software/info1/websphere/index.jsp?tab=landings/esbbenefits*, ESB—Enterprise Service Bus
11. *http://www-306.ibm.com/software/solutions/soa/*, Service Oriented Architecture—SOA
12. Curbera F, Duftler M, Khalaf R, Nagy W, Mukhi N, Weerawarana S (2002) Unraveling the web services web: An introduction to SOAP, WSDL, UDDI. IEEE Internet Computing 6(2):86–93
13. Kohloff C, Steele, R (2003) Evaluating SOAP for high performance business applications: real-time trading systems. *http://www2003.org/cdrom/papers/alternate/P872/p872\ kohlhoff.html*, accessed 22 March 2005
14. *http://www.epcglobalinc.org/standards/specs/*
15. *http://acmqueue.com/modules.php?name=Content& pa=showpage&pid=216*
16. *http://java.sun.com/javame/index.jsp*
17. *http://developers.sun.com/techtopics/mobility/midp/articles/parsingxml/*
18. *http://www.phonescoop.com/glossary/term.php?gid=131*
19. *http://www.xml.com/pub/a/2004/04/14/mobile.html*
20. *http://www.w3.org/TR/2006/WD-xhtml2-20060726/*
21. Geer D (2005) Computer 38(4):16–18, Computer Society Press, Los Alamitos, CA, USA
22. Miklau G, Suciu D (2002) Cryptographically enforced conditional access for XML. Fifth international workshop on the web and databases (WebDB 2002) Madison, Wisconsin
23. *http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm#wp1024961*
24. *http://www.xml.com/pub/r/904*
25. *http://www.w3.org/TR/exi-measurements/*, Efficient XML interchange measurements. Note—W3C working draft July 2006
26. *http://www.agiledelta.com/w3c_binary_XML_proposal.html*, Theory, benefits and requirements for efficient encoding of XML documents. AgileDelta Inc

27. *http://www.w3.org/TR/2005/REC-xop10-20050125/*, XML-binary optimized packaging—W3C Recommendation 25 January 2005
28. *http://www.winzip.com/*, WinZip—The Compression utility for Windows
29. *http://www.rarlab.com/*, WinRAR archiver
30. Liefke H, Suciu D (1999) XMill: an efficient compressor for XMLL data. Technical Report MSCIS-99-26, University of Pennsylvania
31. Ghandeharizadeh S, Papadopoulos C, Cai M, Chintalapudi KK (2002) Performance of networked XML-driven cooperative applications. In: Proceedings of the second international workshop on cooperative internet computing, Hong Kong, China
32. Ghandeharizadeh S, Papadopoulos C, Cai M, Zhou R, Pol P (2003) NAM: a network adaptive middleware to enhance response time of web services. MASCOTS 2003:136
33. Ridgewell A, Huang X, Sharma D (2005) Evaluating the size of the SOAP for integration in B2B. The ninth international conference on knowledge-based intelligent information & engineering systems, Melbourne, Australia, Part IV, pp 29
34. Huang X, Ridgewell A, Sharma D (2006) Efficacious transmission technique for XML data on networks. International Journal of Computer Science and Network Security 6(3):14–19
35. Huang X, Ridgewell A, Sharma D (2006) A Dynamic threshold technique for XML data transmission on networks. The tenth international conference on knowledge-based intelligent information & engineering systems, Bournemouth, UK
36. Huang X, Ridgewell A, Sharma D (2007) A Dynamic adaptive threshold transmission for XML data on networks. The fourth IEEE consumer communications and networking conference, Las Vegas, USA
37. *http://www.w3.org/TR/2006/REC-XML11-20060816/#sec-TextDecl*, W3C XML 1.1 specification
38. *http://www.unicode.org/standard/standard.html*, Unicode Stanard
39. *http://condor.depaul.edu/∼jkristof/technotes/tcp.html*, TCP/IP
40. *http://www.techweb.com/encyclopedia/defineterm.jhtml?term=latency&x=& y=*
41. *http://acm.org/sigcomm/sigcomm97/papers/p086.pdf*, Vern Paxson V (1997) End-to-end internet packet dynamics. Network Research Group, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, *vem@ee.lbl.gov*
42. Iyer RR, Wilhite D (1994) Data compression support in databases. In: Proceedings of the 20th international conference on very large databases
43. Poess M, Floyd C (2000) New TPC benchmarks for decision support and web commerece. ACM SIGMOD Record 29(4)
44. Huang X, Ridgewell A, Sharma D (2006) A Dynamic threshold technique for XML data transmission on networks. The tenth international conference on knowledge-based intelligent information & engineering systems, Bournemouth, UK. Gabrys B, Howlett RJ, Jain LC (eds) KES 2006 Part III. LNAI 4253, pp 1163–1167, Springer-Verlag, Berlin
45. Huang X, Sharma D (2007) A New efficient transmission for XML data on networks. International multiconference of engineers and computer scientists Hong Kong. In: Proceedings of the international multiconference of engineers and computer scientists, Volume II, pp 1238–1241

Chapter 7
An Authenticated Key Management Scheme for Hierarchical Wireless Sensor Networks

Firdous Kausar, Ashraf Masood, and Sajid Hussain

7.1 Introduction

Due to recent advances in electronic industry, wireless sensors can be used in various ubiqutous and pervasive applications such as military, security, healthcare [1,2], industry automation, and environmental and habitat monitoring [3,4]. Wireless sensor networks (WSNs) consist of a large number of low-power nodes, with limited processing, communication, and storage resources [5]. Due to limited resources of WSNs, it is challenging to incorporate basic security functions such as authentication, access control, data integrity, privacy, and key distribution. For instance, asymmetric cryptography such as RSA or elliptic curve cryptography (ECC) is unsuitable for most sensor architectures due to high energy consumption and increased code storage requirements. To avoid the use of asymmetric cryptography, several alternative approaches have been developed to perform key management on resource-constrained sensor networks, such as random key predistribution schemes, plain text key exchange schemes, and transitory master key schemes

In WSNs, hierarchical clustering provides scalability, self-organization, and energy efficient data dissemination [6]. A number of cluster formation protocols have been proposed, but most existing protocols assume benign environments, and are vulnerable to attacks from malicious nodes. In this chapter, we use an existing ring structure, energy-efficient, clustering architecture (RECA) [7] to divide nodes into clusters. However, we modify its initial cluster formation algorithm by providing an authentication mechanism so that no malicious node can take part in the cluster formation process. The proposed authenticated key management (AKM) scheme enables only legitimate nodes to join the network.

Firdous Kausar and Ashraf Masood
College of Signals, National University of Science and Technology, Rawalpindi, Pakistan
e-mail: firdous.imam@gmail.com,mashraf@comsat.net.pk

Sajid Hussain
Acadia University, Jodrey School of Computer Science, Nova Scotia, Canada
e-mail: sajid.hussain@acadiau.ca

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

The rest of the chapter is organized as follows: Section 7.2 discusses key distribution schemes, which are relevant to the proposed scheme. In Sect. 7.3, the AKM scheme is described in detail, such as nodes authentication prior to joining a network, secure cluster-formation along with details about shared key discovery, and new node addition. Section 7.4 focuses on the performance and security analysis of the AKM scheme. Section 19.7 summarizes the major contributions of this work.

7.2 Related Work

There are many key management protocols that are proposed for WSN. Eschenauer and Gligor [8] propose a probabilistic key predistribution technique to bootstrap the initial trust between sensor nodes. The main idea is to have each sensor randomly pick a set of keys from a key pool before deployment. Then, to establish a pairwise key, two sensor nodes only need to identify the common keys that they share. Chan et al. further extended this idea and proposed the q-composite key predistribution [9]. This approach allows two sensors to set up a pairwise key only when they share at least q common keys. Chan et al. also developed a random pairwise keys scheme to defeat node capture attacks.

Carman et al. [5] study the performance of a number of key management approaches in sensor networks on different hardware platforms. Perrig et al. [10] develop a security architecture for sensor networks (SPINS), where each sensor node shares a secret key with the base station. Because two sensor nodes cannot directly establish a secret key, they can use a base station as a trusted third party to set up the secret key.

Basagni et al. [11] present a key management scheme to secure the communication by periodically updating the symmetric keys shared by all sensor nodes. However, this scheme assumes a tamper-resistant device to protect the key, which is not always available in sensor networks.

Blundo et al. [12] proposed several schemes that allow any group of t parties to compute a common key, while being secure against collusion between some of them. These schemes focus on saving communication costs without memory constraints being placed on group members. When $t = 2$, one of these schemes is actually a special case of Blom's scheme [13].

Availability of some information on the sensor distribution in the field helps to improve the security of the key predistribution schemes. Some location-aware schemes are proposed in [14] and [15]. These techniques divide the target field into nonoverlapping square areas and randomly deploy the sensors in every area. The exact location of a sensor in any area is unknown, but there is knowledge about the identity of sensors in every area. This information helps to eliminate the dependency of keys between nonadjacent cells.

Karlof and Wagner [16] identify security goals for routing in sensor networks and analyzed the vulnerabilities as well as the countermeasures for a number of existing routing protocols. Zhu et al. [17] give localized encryption and authentication proto-

col (LEAP), which is a complete key management framework for static WSNs that includes mechanisms for securing node-to-base station traffic, base station-to-nodes traffic, local broadcasts, and node-to-node (pairwise) communications.

There is active research on hierarchical clusters. For instance, Hu et al. [18] offer a secure power-efficient clustered-topology routing algorithm (SPECTRA), which integrates routing and key management to provide an energy efficient security and routing solution. Chan et al. [19] describe the problem of designing a clustered distributed sensor network (DSN) for specific scenarios, where the probability of node compromise in different deployment regions is known a priori. They use the a priori probability to design a variant of random key pre-distribution method that improves the resilience and reduces the compromised communications as compared to other related schemes. Oliveira et al. [20] propose SecLEACH, which shows how a random key predistribution can be used for secure communication in hierarchical (cluster-based) protocols, such as LEACH [21].

7.3 The AKM Scheme

We propose an authentication and key management scheme for hierarchical clusters in WSNs.

7.3.1 Terms and Assumptions

Some of the terms and assumptions that are needed for the proposed scheme are as follows:

7.3.1.1 Network Model

The WSN consists of a three-tier architecture consisting of a base station (BS), cluster heads, and cluster members. The base station is a secure node that is not prone to failure. It has virtually unlimited computational, communication, and memory resources. Further, it is assumed that base stations can transmit directly to every sensor node.

Sensor nodes, however, are battery-constrained and inexpensive nodes. They have limited communication, processing, and memory storage resources. Each sensor node can act as a clusterhead (CH) or a cluster member. The CH is chosen based on a RECA algorithm given below. A cluster member communicates directly with its clusterhead (CH); there is no communication between sensors. In other words, there is one-hop communication between a cluster member and the CH. Further-more, clusterheads can communicate with each other directly and to the base station. Figure 7.1 shows a WSN with three clusters. The cluster heads transmit

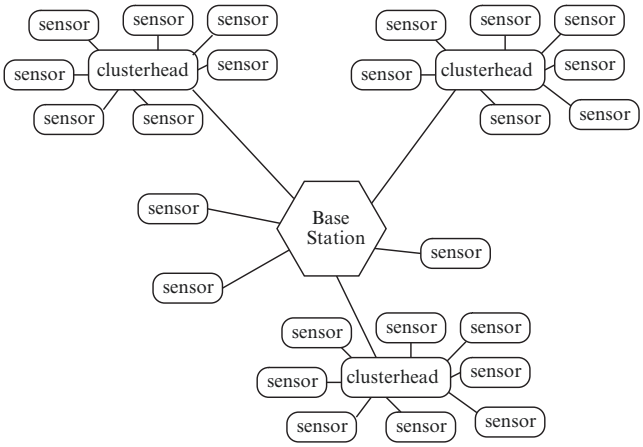


Fig. 7.1 Wireless sensor network

directly to the base station, and cluster members communicate with their clusterheads only. Further, there are a few isolated sensors that do not belong to any cluster and they communicate directly with the base station.

7.3.1.2 Threat Model

We assume that an adversary can access all the data (including keys) of a compromised sensor node. Furthermore, an adversary can eavesdrop on the traffic, inject new messages, replay and change old messages, or spoof other identities.

7.3.1.3 Authentication Key (K_{Auth_i})

The authentication key is a unique pair-wise shared key between a sensor node and the base station. During the initial setup phase (i.e., just after sensor deployment), sensor nodes and CHs must authenticate themselves with the BS using their corresponding authentication keys. The authentication key can only be used once in the lifetime of a node, and the key is deleted after authentication from the base station.

7.3.1.4 Network Key (K_{Net})

The base station uses a network key to provide a secure broadcast for WSN. All messages transmitted by the base station are encrypted by the network key. Furthermore, the base station refreshes the network key periodically.

7.3.1.5 Cluster Key ($K_{CH_{i,A}}$)

Each cluster member uses a unique pair-wise shared key with its clusterhead. The cluster key provides confidential communication between a cluster member and its clusterhead.

7.3.1.6 Security Level

The security level gives the probability that a randomly chosen link is not compromised when a node, which is not the either end of the link, is compromised.

7.3.1.7 Key Pool (P)

A key pool is a large pool of random symmetric keys that is stored at the base station. The base station assigns a particular number of keys to each sensor node, where keys are randomly chosen from the key pool without replacement. S is the number of keys in the key pool.

7.3.1.8 Key Ring (R)

The keys that are assigned to each node from the key pool is called a key ring. The number of keys in the ring R is given by m.

7.3.1.9 Unsupervised Node

The nodes that do not belong to any cluster are called unsupervised nodes. These nodes communicate directly with the base station. For convenience, a summary of notations and symbols used in the paper are given in Table 7.1.

7.3.2 Key Predeployment Phase

First, the base station generates a large pool of S keys and their IDs prior to network deployment. Each node is then assigned a ring of m number of keys, which are drawn from the pool pseudorandomly without replacement.

For each node X, we use a pseudorandom function (PRF) to generate its unique ID called id_X, which is then used to seed a pseudorandom number generator (PRNG) of a large enough period to produce a sequence of m numbers $(n_1, n_2, \cdots, n_m)$. RID_X is the set of key IDs assigned to X, which can be obtained by mapping each number in the sequence to its corresponding value modulus N $\big($i.e., $n_i \; mod \; N$, where

Table 7.1 Symbol definition

Notation	Definition
BS	Base Station
CH	Cluster Head
K_{Auth}	Authentication Key
K_{Net}	Network Key
P	A pool of keys
S	A number of keys in a key pool
R_X	Set of the keys in node X's key ring
RID_X	Set of key IDs in node X's key ring
m	A number of keys in a key ring
$K_{X,Y}$	A shared key between X and Y
$K_{CurrNet}$	Current Network Key
$K_{RefreshNet}$	Refreshed Network Key
id_A	Identity of node A
$E_K(m)$	An encryption of message m with key K
$nonce$	A random number String
$mac_K(msg)$	MAC calculated using key K
d_A	Data from node A
$F(d_i, d_j, \cdots, d_n)$	Data aggregation function
adv	Advertisement message announced by CH

$i \in [1,m]\big)$. Also prior to deployment, each node is assigned a unique pairwise key (K_{Auth_i}) shared with the BS.

7.3.3 Initial Setup Phase

During the initial setup phase, all sensor nodes and CHs are authenticated. A node is authenticated by using the latest network key. To get the latest network key, node A_i sends a request to the BS. The request consists of node's id, nonce, and MAC, where MAC is calculated by using that node's authentication key K_{Auth_i}, as shown in Fig. 7.2. The BS authenticates the node by verifying the MAC using that node's authentication key associated with it's node ID. The BS replies to the node with the latest network key K_{Net}, which is encrypted with K_{Auth_i} of the requesting node, and marks the K_{Auth_i} of that node as *used* in its database. The node receives the message from BS and uses it K_{Auth_i} to decrypt the network key. Also, the node A_i deletes K_{Auth_i} from its memory after joining the network.

7.3.4 Cluster Organization Phase

After authentication and receiving K_{Net} from base station, nodes enter into the cluster organization phase. Figure 7.3 shows the algorithm for initial cluster formation.

$A_i \Rightarrow$ BS: $Msg\left(id_{A_i}, nonce, MAC_{K_{Auth_i}}(id_{A_i} \| nonce)\right)$

Fig. 7.2 Node authentication $BS \Rightarrow A_i$: $E_{K_{Auth_i}}(K_Net)$

```
/* Initial State */
γ: the expected number of nodes in one cluster.
t_slot: duration of one time slot.
head: TRUE if the node is a cluster head; otherwise FALSE.
member: TRUE if the node is a cluster member; otherwise FALSE.

procedure InitCluster()
1: for i=0 to ⌈log2γ⌉ do
2:   if (!head) AND (!member) then
3:     generate α ∈ [0,1]
4:     if α < min (2^i/γ, 1) then
5:       advertise itself as a cluster head
6:       broadcast id_CH, nonce, adv, mac_KNet(id_CH||nonce)
7:       return
8:     end if
9:   end if
10:   listen for a duration of t_slot
11:   if (!head) AND (adv_rcvd > 0) then
12:     choose id_KCH,Ai, ∀ id_KCH,Ai ∈ (RID_CH ∩ RID_Ai)
13:     send id_Ai, id_CH, id_KCH,Ai, join_req, mac_Ks(id_Ai||id_CH||id_KCH,Ai||nonce) to CH
14:     join the cluster using suitable RSSI and LQI values
15:   end if
16: end for
```

Fig. 7.3 Initial cluster formation algorithm

Table 7.2 Possible node states

Head	Member	Status
true	false	cluster head
false	true	cluster member
false	false	uninitialized
true	true	erroneous

The algorithm is executed on every node in the network. The *head* and *member* variables determine the clusterhead and cluster member states, respectively. Table 7.2 shows the possible states for a node. For a clusterhead, the values of *head* and *member* are true and false, respectively. However, for an uninitialized node, both *head* and *member* are false. Furthermore, if both *head* and *member* are true, it shows an erroneous state that is not possible in a typical case.

The γ determines the estimated number of nodes in a cluster that can be estimated as $\gamma = \frac{N \times \pi \times r^2}{A}$, where N is the total number of nodes in a network, A is the network area, and r is the minimum transmission range.

If a node is uninitialized, it generates a random number α between 0 and 1. If α is less than a given threshold, $\min\left(\frac{2^i}{\gamma}, 1\right)$, the node acts as a clusterhead and broadcasts

1: $CH \Rightarrow * : id_{CH}, nonce, adv, mac_{K_{Net}}(id_{CH} \| nonce)$
/* A_i chooses $id_{K_{CH,Ai}}$, $\forall\ id_{K_{CH,Ai}} \in (RID_{CH} \cap RID_{A_i})$ */
2: $A_i \Rightarrow CH : id_{A_i}, id_{CH}, id_{K_{CH,Ai}}, join\_req, mac_{K_{CH,Ai}}(id_{A_i} \| id_{CH} \| id_{K_{CH,Ai}} \| nonce)$
3: $A_i \Rightarrow CH : id_{A_i}, id_{CH}, E_{K_{Net}}(E_{K_{CH,Ai}}(d_{A_i}))$
4: $CH \Rightarrow BS : id_{CH}, E_{K_{Net}}(E_{K_{BS,CH}}(F(d_i, d_j, \cdots, d_n)))$

Fig. 7.4 Messages transferred between sensor nodes, CHs, and the BS

an advertisement message consisting of id_{CH}, nonce, *adv*, and $\mathrm{mac}_{K_{Net}}(\mathrm{id}_{CH} \| \mathrm{nonce})$, where MAC is generated using latest K_{Net}. Upon receiving the CH advertisement message *adv*, the BS authenticates *adv* and generates a new key K_{BS,CH_i}, which is transmitted to the CH encrypting it with K_{Net}.

The nodes listen to the clusterhead advertisements for a period t_{slot}. Upon receiving clusterhead advertisement message *adv*, node A_i a) authenticates the CH by verifying the MAC, using the latest K_{Net}, b) computes the set of CH key IDs (using the pseudorandom scheme described above), and c) identifies whether it shares a key $K_{CH,i}$ with that CH.

At the end of t_{slot}, the node chooses the CH with whom it shares a key and has the best received signal strength and link quality. Then, it sends a join request message to the selected CH; the join request is protected by MAC, using $K_{CH,i}$ and includes the nonce from CH broadcast (to prevent replay attack), as well as the ID of the shared key ($K_{CH,i}$) chosen to protect this link (so that the receiving CH knows which key to use to verify the MAC).

Figure 7.4 shows the messages transferred between sensor nodes, CHs, and the BS. First, a CH broadcasts the advertisement. Second, a node A_i joins a cluster based on the received signal strength and shared key. Third, a node transmits its data to the CH. Finally, the CH sends the aggregated data to the BS.

Node-to-CH communications are protected using two keys: K_{Net} and $K_{CH,i}$. First, the data is encrypted using the latest K_{Net}. Second, the encrypted data is again encrypted using $K_{CH,i}$. Then, the data is forwarded to the CH. The CHs decrypt the received data, perform data aggregation, and send the aggregated result to BS, where the result is first encrypted with K_{Net}, and then with K_{BS,CH_i}.

7.3.4.1 Unsupervised Nodes

At the end of the clustering process, it is expected that a fraction of the ordinary nodes will not be matched with a CH because of key sharing constraints. These nodes are called unsupervised nodes. The unsupervised nodes will communicate directly with the BS. The unsupervised node X sends a special message $(id_X, req, nonce, mac_{K_{Net}}(id_X \| req \| nonce))$ to BS requesting a pair-wise shared key. The BS generates a key $K_{BS,X}$ and sends it to the corresponding unsupervised node, encrypting it with K_{Net}. The node X decrypts the message and will use the key $K_{BS,X}$ for future communication with the BS. Furthermore, every message sent to the BS is first encrypted with K_{Net} and then with $K_{BS,X}$. The number of unsupervised nodes depends on the size of key pool, the size of the key ring, and the number of CHs.

7.3.4.2 Refreshing the Network Key

All nodes in the network are continuously and periodically authenticated. This is achieved through the periodic refreshing of the K_{Net}. The network key is valid only for an *epoch*. An epoch is defined as a period of time that is less than the predicted time required for node compromise. The epoch time is dependent on the network environment. After that epoch, BS generates a new network key $K_{RefreshNet}$ and broadcasts it by encrypting with the current network key $K_{CurrNet}$. The nodes in the network receive the broadcast message, decrypt it using the current network key, and get the refreshed network key. Now the refreshed network key will be used as the current network key—i.e., as a K_{Net} in the network.

7.3.5 New Node Addition

When a new node (say, u) joins a network, it first gets a current network key from BS after going through the authentication process. Then, it broadcasts a *hello* message consisting of its ID, a nonce, and a MAC calculated on these two values using K_{Net}. The CHs, which receive hello messages, authenticate the node by verifying the MAC, and then send the advertisement message *adv*. The node u verifies the MAC of receiving advertisement messages and computes the set of CH key IDs (using the pseudorandom scheme described above), chooses the closest CH with whom it shares a key $K_{CH,u}$, and sends a join request message. This message is protected by MAC using $K_{CH,u}$, and includes the nonce from CH broadcast (to prevent replay attack), as well as the ID of shared keys ($\mathrm{id}_{K_{CH,u}}$) chosen to protect this link as described above.

7.4 Analysis

This section analyzes the AKM scheme and compares it with the other related approaches. Given a WSN, the amount of storage allocated for keys in each node is likely to be a preset constraint, which makes the size of the key ring m a fixed parameter. Once m is set, the choice of S will impact the security level and the probability of key sharing among nodes in the network.

Given a (S,m)-network, where each node is assigned m keys from a key pool of size S, the security level sl can be defined as follows:

$$sl = 1 - \frac{m}{S} \tag{7.1}$$

which gives the probability that a randomly chosen link is not compromised, when a node that is not either end of the link is compromised. For a fixed m, sl is proportional to S, as shown in Fig. 7.5.

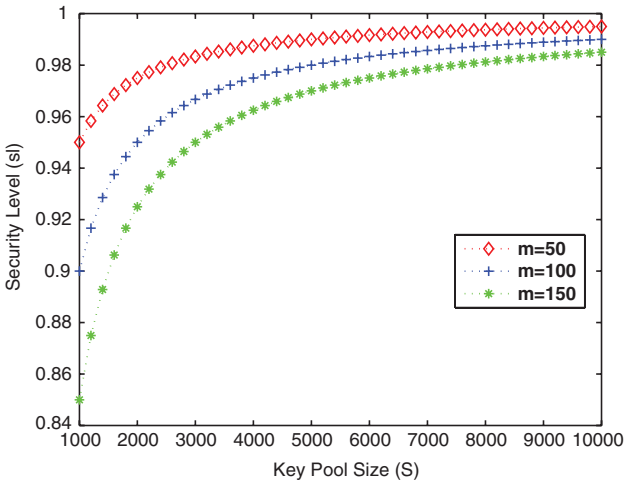


Fig. 7.5 Security level

Given any two nodes in a (S,m)-network, the probability (P_s) that they will share a key is given by:

$$P_s = 1 - \overline{P_s} \tag{7.2}$$

where $\overline{P_s}$ is the probability that they will not share a key, is given by:

$$\overline{P_s} = \frac{[(S-m)!]^2}{S!(S-2m)!} \tag{7.3}$$

Figure 7.6 shows the variation in probability of key sharing with respect to key pool size S and ring size m. For a fixed m, P_s decreases as the value of S is increased. For fixed S, however, the P_s increases as the value of m increases.

In our scheme, only a fraction of CHs are probabilistically accessible by an ordinary node. P_s and the number of CHs h in the network can also determine the expected number of unsupervised nodes—i.e., the probability that an ordinary node will be unsupervised. Given P_s and h, the probability of the number of unsupervised nodes is given by:

$$P_0 = (1 - P_s)^h \tag{7.4}$$

In a network with N nodes, it is then expected that $N \times P_0$ nodes will be unsupervised. Figure 7.7 shows a fraction of unsupervised nodes as a function of h under different values of sl. As h increases, the number of unsupervised nodes decrease rapidly. Furthermore, as security level sl increases, the number of unsupervised nodes also increase.

Table 7.3 gives probability P_s of key sharing as a function of security level sl and key pool size S. For a fixed value of S, as sl increases, the value of P_s decreases accordingly. However, for fixed sl, as the value of S increases, the value of P_s also increases.

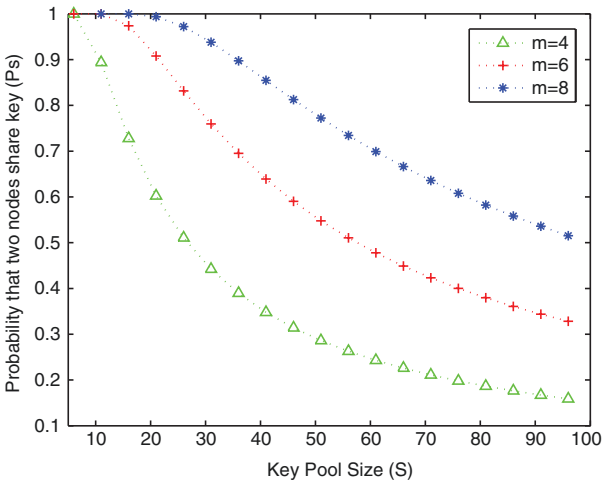


Fig. 7.6 Probability of key sharing

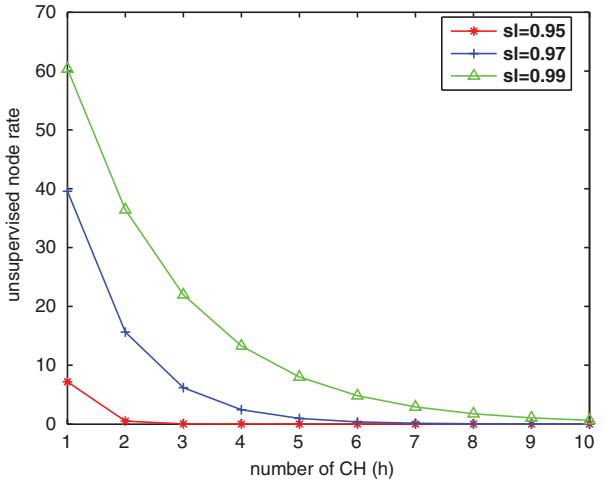


Fig. 7.7 Unsupervised node rate

Table 7.3 Probability P_s of key sharing as a function of security level *sl* and key pool size *S*

	Security Level (**sl**)		
S	**0.95**	**0.97**	**0.99**
1000	0.9280	0.6044	0.0960
2000	0.9948	0.8436	0.1828
5000	0.9999	0.9903	0.3964

7.4.1 Security Analysis

In key distribution schemes, resiliency against node capture measures how much of the network (its communication links) is compromised when a node is compromised. It is a critical performance measure that gauges the robustness of a solution. Our scheme is more resilient against node capture in comparison to other random key predistribution schemes, provided that compromised node re-enter into the network after the epoch in which network key has been refreshed. If a node gets compromised, it is possible for the adversary to know all the keys stored at that node. If we expect that the attacker requires a fixed amount of time to compromise the node, the network key would have changed to a new one before the attacker could use the compromise keys.

The tampered node cannot rejoin the network, because K_{Auth_i} is used only once during the lifetime of a node at the time of joining the network to get the current network key. The advantage of deleting K_{Auth_i} (after authentication) is that if a node is being captured by an adversary, and the adversary wants to re-enter the node into the network after the time period in which the K_{Net} has been refreshed, it will fail to get the current K_{Net} because its K_{Auth} has been deleted. But if the adversary re-enters the compromised node into the network before refreshing the current K_{Net}, the resiliency of AKM scheme will be same as given in Eschenauer et al. [8]

Our scheme provides secure cluster formation process and prevents the malicious nodes from joining the network. As shown in an initial cluster formation algorithm, node to CH authentication is achieved by verifying the MAC calculated using current K_{Net} by the CH in the advertisement message. It also provides CH to member authentication by verifying the MAC calculated using $K_{CH,i}$ by cluster members in the join_req message.

The global authentication is achieved by periodically refreshing the network key; this feature allows every entity in the network to be confirmed or authenticated continuously, and reduces the chances of compromise because compromise has to occur before the system is authenticated again.

Confidentiality and authentication at every step in the network is provided by using two keys. First, any message passed within the network is encrypted with the network key, and then by the shared key. Therefore, to participate within the network, the current network key must be known. If a node is lacking the current network key, no information can be sent or received. Multiple keys are important in the network because they make compromise exponentially difficult. Not only must a compromised node have knowledge of two different keys (i.e., network key and shared key), but it must also know exactly when to use them.

7.5 Conclusion

Key establishment, a fundamental building block in cryptography, is defined as any process whereby a shared secret key becomes available to two or more par-

ties for subsequent cryptographic use. This paper proposes a variant of a random key predistribution scheme for bootstrapping the clustered WSNs. Clustering approaches have been found useful in providing scalable data aggregation, security, and coding for large scale WSNs. In our proposed scheme, we show that it is more resilient to node capture in comparison to other random key distribution schemes, provided that compromised nodes rejoin the network after the epoch in which network key has been refreshed. It also provides global and continuous authentication of nodes in the network by periodically refreshing the network key. Finally, it provides multiple levels of encryption by using more than one encryption key and secure cluster formation algorithm.

References

1. Gao T, Greenspan D, Welsh M, Juang RR, Alm A (2005) Vital signs monitoring and patient tracking over a wireless network. In: The 27th Annual International Conference of the IEEE EMBS, Shanghai, China
2. Gu L, Jia D, Vicaire P, Yan T, Luo L, Tirumala A, Cao Q, He T, Stankovic JA, Abdelzaher T, Krogh BH (2005) Lightweight detection and classification for wireless sensor networks in realistic environments. In: The 3rd ACM Conference on Embedded Networked Sensor Systems, San Diego, USA
3. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Communications Magazine
4. Kahn J, Katz R, Pister K (1999) Next century challenges: Mobile networking for smart dust. In: The ACM International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, USA
5. Carman DW, Kruus PS, Matt BJ (2000) Constrains and approaches for distributed sensor network security. In: Technical report, NAI Labs
6. Zhao F, Guibas L (2004) Wireless sensor networks. Elsevier Inc 23–24
7. Li G, Znati T (2007) Reca: a ring-structured energy efficient cluster architecture for wireless sensor networks. International Journal of Sensor Networks **2** 34–43
8. Eschenauer L, Gligor VD (2002) A key management scheme for distributed sensor networks. In: ACM CCS
9. Chan H, Perrig A, Song D (2003) Random key pre-distribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy. 197–213
10. Perrig A, Szewczyk R, Tygar J, Victorwen, Culler DE (2001) Spins: Security protocols for sensor networks. In: Seventh Annual Int'l Conf. on Mobile Computing and Networks
11. Basagni S, Herrin K, Bruschi D, Rosti E (2001) Secure pebblenets. In: ACM Int'l Symp. on Mobile ad hoc networking and computing. 156–163
12. Blundo C, Santis AD, Herzberg A, Kutten S, Vaccaro U, Yung M (1993) Perfectly-secure key distribution for dynamic conferences. In: CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, London, UK, Springer-Verlag 471–486
13. Blom R (1985) An optimal class of symmetric key generation systems. In: Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques, New York, NY, USA, Springer-Verlag New York, Inc. 335–338
14. Liu D, Ning P (2003) Location-based pairwise key establishments for static sensor networks. In: SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, New York, NY, USA, ACM Press 72–82

15. Wadaa A, Olariu S, Wilson L, Eltoweissy M (2004) Scalable cryptographic key management in wireless sensor networks. In: ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04), Washington, DC, USA, IEEE Computer Society 796–802
16. Karlof C, Wagner D (2003) Secure routing in wireless sensor networks: Attacks and countermeasures. In: IEEE Int'l Workshop on Sensor Network Protocols and Applications
17. Zhu S, Setia S, Jajodia S (2003) Leap: Efficient security mechanisms for large-scale distributed sensor networks. In: ACM Conference on Computer and Communications Security. 62–72
18. Hu F, Siddiqui W, Cao X (2005) Spectra: Secure power-efficient clustered-topology routing algorithm in large-scale wireless micro-sensor networks. International Journal of Information Technology **11**
19. Chan SP, Poovendran R, Sun MT (2005) A key management scheme in distributed sensor networks using attack probabilities. In: IEEE Globecom
20. Oliveira LB, Wong HC, Bern M, Dahab R, Loureiro AAF (2006) Sec leach: A random key distribution solution for securing clustered sensor networks. In: 5th IEEE international symposium on network computing and applications. 145–154
21. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. In: IEEE Hawaii Int. Conf. on System Sciences. 4–7
22. Zhu S, Xu S, Setia S, Jajodia S (2003) Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach. In: ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols, Washington, DC, USA, IEEE Computer Society 326

Chapter 8
A New Approach in Scheduling for Differentiated Services Networks

Ghasem Mirjalily and Mahnaz Ahmadian

8.1 Introduction

In previous decades, the Internet has moved from a limited low bandwidth network to a sophisticated infrastructure supporting many new applications such as video conferencing and Internet telephony. These applications have diverse QoS requirements. Support of QoS in such a packet-switched network requires a broad range of functions, such as priority mechanisms, scheduling disciplines, traffic shaping schemes, QoS signaling, and routing algorithms.

One of the most important parameters in QoS is the delay of packets. The absolute guarantee model and the proportional service model are two methods to guarantee the delay. In the absolute guarantee model, a fixed maximum service delay for each class needs to be enforced. In the proportional service model, a fixed ratio between the delays seen by the different service classes can be enforced. To reach the delay constraints, routers allocate the rates of classes by scheduling algorithms.

Fair allocations are used in preliminary scheduling algorithms. Fair allocations are typically achieved by using per-flow queuing mechanisms such as Fair Queuing [1, 2] and its many variants [3–5]. Several other scheduling algorithms have been proposed which involve picking a packet from each input in a probabilistic [6, 7] or round robin [8, 9] fashion. The Internet Engineering Task Force (IETF) has introduced differentiated services (DiffServe) [10, 11] to provide relative delay constraints for different classes. DiffServe defines a set of traffic classes each of which is designed to serve applications with similar QoS demands. A traffic class describes the per hop behavior (PHB) that the packets of this class should experience in each network node. The per hop behavior determines the priority, the maximum delay in the transmission queues, the link-sharing bandwidth, and the probability of

Ghasem Mirjalily and Mahnaz Ahmadian
Yazd University, Yazd, Iran
e-mail: mirjalily@yazduni.ac.ir

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

a packet being dropped. The DiffServe model ensures high scalability by separating the operations performed in the borders of the network from those accomplished in the core network.

Scheduling algorithms can be placed into two groups. In the first group, the service rate is dynamically changed in response to the number of waiting packets in each queue [12, 13]. This method is basically an advanced versions of a GPS [14]. In the second group, the priorities of queues are adaptively modified to satisfy the limits. Some algorithms in this group are waiting time priority [13], mean delay proportional [15], proportional queue control mechanism [16], and hybrid proportional delay scheduler [17]. Only a few of these algorithms consider proportional loss probability. One of these is the proportional loss rate dropper [18], which has the problem of time dependency between loss calculation and loss proportion.

Some recent works consider joint queue management and scheduling as a single problem [12]. In [17], the RIO algorithm is discussed as the superposition of two RED algorithms which are applied to the in-band and out-band traffic. Liebherr et al [19] propose a powerful algorithm named JoBS. This algorithm is used in the QoSBox, which is a machine guaranteeing QoS in the backbone of the DiffServ architecture. A new algorithm proposed in [20], named equivalent differentiated services, uses the waiting time priority [13] and proportional loss [18] to control queue input and output.

Control theory is used in computer networks as a powerful tool to improve robustness, accuracy, and throughput. For example, Christian et al [12] proposed a controller that performs buffer management and rate allocation to provide loss and delay differentiation for traffic classes at a router. Both relative and absolute (if possible) QoS requirements are supported by this algorithm. To satisfy relative delay constraints (RDC), it uses a proportional controller in its delay feedback loop. To improve this algorithm, we have used PID and fuzzy PID controllers and will show their better performance by simulation [21].

In the next section we briefly introduce the delay feedback loop. In Sect. 8.3 we replace the proportional controller used in [12] with a PID controller, and in Sect. 8.4 we tune its parameters using fuzzy systems. Finally in Sect. 8.5, by offering simulation results, we show that our proposed methods outperform ordinary proportional controllers.

8.2 Delay Feedback Loop

In this section, we explain the scheduling algorithm introduced by Christian et al to satisfy RDCs. In DiffServe, the task of the scheduler in a router is to assign service rates to different classes of incoming traffic, while this assignment is subject to delay constraints. To formulate the delay feedback loop, assume that all the traffic that enters the input queue of a router is from one of the N defined classes. Classes with lower indices are served better. If we define $D_i(n)$ as the delay of class i at time n; the RDCs can be written as follows:

$$\frac{D_{i+1}(n)}{D_i(n)} = k_i \tag{8.1}$$

That leads us to the following equation system [21]:

$$\begin{aligned} D_2(n) &= k_1 D_1(n) \\ &\vdots \\ D_N(n) &= \left(\prod_{j=1}^{N-1} k_j\right) D_1(n) \end{aligned} \tag{8.2}$$

Now we define $m_i = \prod_{j=1}^{i-1} k_j$ for $\forall i > 0$ and $m_1 = 1$, so we can define weighted delay classes as follows:

$$D_i^*(n) := g_i D_i(n) \tag{8.3}$$

in which:

$$g_i = \prod_{k=1, k\neq i}^{N} m_k \tag{8.4}$$

By multiplying each line of (2) with g_i, we can see that if the following relation is held, RDCs are satisfied:

$$\forall i, j, \forall n : D_j^*(n) = D_i^*(n) \tag{8.5}$$

Equation (5) is equivalent to the following equations:

$$\begin{aligned} &\forall i, j, \forall n : D_i^*(n) = \bar{D}^*(n) \\ &\bar{D}^*(n) = \frac{1}{N}\sum_i D_i^*(n) \end{aligned} \tag{8.6}$$

$\bar{D}^*(n)$ can be used as the set point of all delay feedback controllers. In the delay feedback loop, the object is to tune the output bit rate of each class such that relative delay of classes are assured. The delay feedback loop is defined as follows:

$$e_i(n) = \bar{D}^*(n) - D_i^*(n) \tag{8.7}$$

where $e_i(n)$ is the error of the delay constraint of class i. It must be noted that the sum of all errors is equal to zero:

$$\sum_i e_i(n) = N\bar{D}^*(n) - \sum_i D_i^*(n) = 0 \tag{8.8}$$

If the RDCs are completely satisfied, $e_i(n) = 0$ for all classes. If the output bit rate of each class is $r_i(n)$, the controller will determine the required adjustment in $r_i(n)$ using $e_i(n)$. If the required adjustment in $r_i(n)$ is shown by $\Delta r_i(n)$, the controller can be expressed as follows:

$$\Delta r_i(n) = f(e_i(n)) \tag{8.9}$$

$$r_i(n+1) = r_i(n) + \Delta r_i(n) \tag{8.10}$$

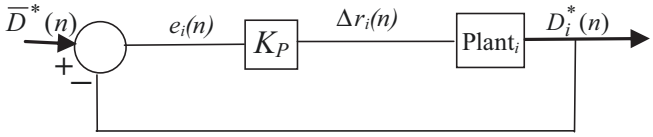


Fig. 8.1 Delay feedback loop for class i

where $f(.)$ is the delay feedback controller. An ordinary algorithm uses a proportional controller. Therefore, (9) would be:

$$\Delta r_i(n) = k_p(n).e_i(n) \tag{8.11}$$

where $K_p < 0$ is the proportional gain of the controller. Figure 8.1 shows the block diagram of the delay feedback loop for class i. In this diagram, $Plant_i$ consists of those parts of the network that can affect the delay of class i.

8.3 Developing Algorithm Using PID Controller

We replaced the proportional controller by a PID controller. The integral part of our controller helps to eliminate steady state error while the derivative part improves the transient response. The block diagram of our method is shown in Fig. 8.2.

By replacing the proportional controller with a PID controller, (11) will be transformed to (12), in which $K_d < 0$ and $Ki < 0$ are differential and integral gains of the controller, respectively.

$$\Delta r_i(n) = k_p(n).e_i(n) + K_d(n)(e_i(n) - e_i(n-1)) + \frac{K_i(n)}{n}\sum_{k=1}^{n} e_i(k) \tag{8.12}$$

For simple time invariant systems, to tune PID gains, one can use well known time and frequency response methods such as Ziegler-Nichols that are described in [22, 23]. Nevertheless, these methods are useless in our application since network dynamics are so complex and depend on so many factors that always change. In [12], K_p is selected dynamically, based on experiment, and depends on link bandwidth, utilization, and RDCs. This proportional gain is called K_p-Dynamic. We have used the same K_p for our controller, and we selected K_i, and K_d by multiplying K_p with the constant coefficients C_i and C_d:

$$\begin{aligned} K_p &= K_{p-Dynamic} \\ K_i &= C_i.K_p \\ K_d &= C_d.K_p \end{aligned} \tag{8.13}$$

Simulation results show better performance of our PID controller over the proportional controller.

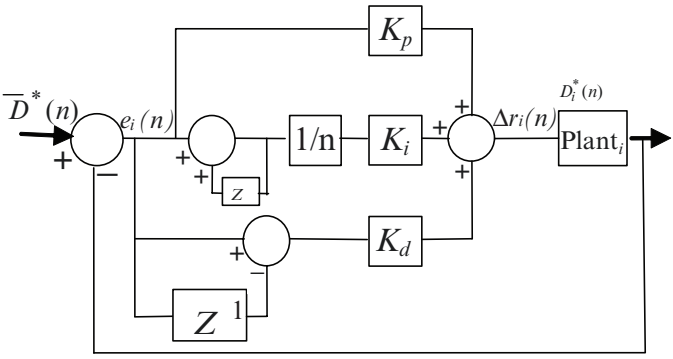


Fig. 8.2 Scheduling with PID controller

8.4 Developing Algorithm Using Fuzzy PID Controller

The PID controller surpasses an ordinary proportional controller. However, it is possible to enhance the PID controller's performance by online tuning of PID gains. We will follow the approach proposed by Zhao, Tomizuka, and Isakaf for fuzzy gain scheduling of our PID controller [24, 25]. The block diagram of the proposed method is shown in Fig. 8.3.

For tuning of gains, we first define ranges $[K_{pmin}, K_{pmax}] \subset R$ and $[K_{dmin}, K_{imax}] \subset R$, such that $K_p \in [K_{pmin}, K_{pmax}]$ and $K_d \in [K_{dmin}, K_{dmax}]$. For convenience, K_p and K_d are normalized to the range between zero and one by the linear transformation:

$$K'_p = \frac{K_p - K_{pmin}}{K_{pmax} - K_{pmin}} \tag{8.14}$$

$$K'_d = \frac{K_d - K_{dmin}}{K_{dmax} - K_{dmin}} \tag{8.15}$$

For simplicity, we assume:

$$K_i = \frac{K_p^2}{\alpha K_d} \tag{8.16}$$

So the parameters that should be tuned are K'_p, K'_d, and α. The inputs of our fuzzy systems are e(n) and $\Delta e = e(n) - e(n-1)$. Thus as shown in Fig. 8.3, our fuzzy system consists of three subsystems (2 input, 1 output) for PID gains tuning.

According to [24], we use three groups of rules, each consisting 49 rules, to tune K'_p, K'_d, and α. These rules are shown in tables I, II, and III. The corresponding membership functions for e(n), $\Delta e(n)$, K'_p, K'_d, and α are shown in Figs. 8.4, 8.5, and 8.6.

We combine the 49 rules in each set using a product inference engine, a singleton fuzzifier, and a center average defuzzifier; that is, the parameters K'_p, K'_d, and α are tuned online according to (17), (18), and (19).

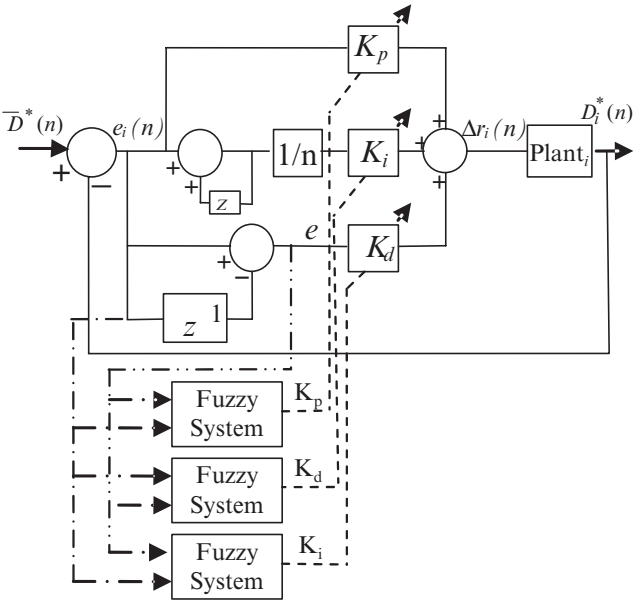


Fig. 8.3 Scheduling with fuzzy PID controller

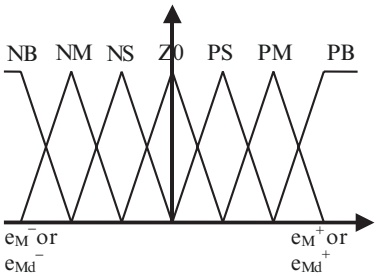


Fig. 8.4 Membership functions of e(n) and Δe(n)

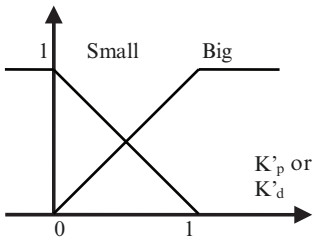


Fig. 8.5 Membership functions of K'_p, K'_d

Fig. 8.6 Membership functions of α

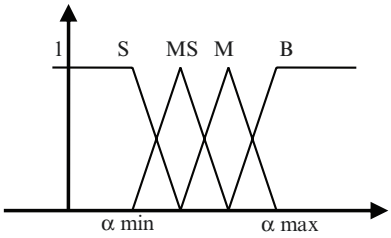


Table 8.1 Fuzzy tuning rules for K'_p

		e(n)						
		NB	NM	NS	Z0	PS	PM	PB
Δe(n)	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	Z0	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

Table 8.2 Fuzzy tuning rules for K'_d

		e(n)						
		NB	NM	NS	Z0	PS	PM	PB
Δe(n)	NB	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	Z0	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

$$K'_p(n+1) = \frac{\sum_{l=1}^{49} y_p^{-1} \mu_{A^l}(e(n)) \mu_{B^l}(\Delta e(n))}{\sum_{l=1}^{49} \mu_{A^l}(e(n)) \mu_{B^l}(\Delta e(n))} \tag{8.17}$$

$$K'_d(n+1) = \frac{\sum_{l=1}^{49} y_d^{-1} \mu_{A^l}(e(n)) \mu_{B^l}(\Delta e(n))}{\sum_{l=1}^{49} \mu_{A^l}(e(n)) \mu_{B^l}(\Delta e(n))} \tag{8.18}$$

$$\alpha(n+1) = \frac{\sum_{l=1}^{49} y_\alpha^{-1} \mu_{A^l}(e(n)) \mu_{B^l}(\Delta e(n))}{\sum_{l=1}^{49} \mu_{A^l}(e(n)) \mu_{B^l}(\Delta e(n))} \tag{8.19}$$

where A^l and B^l are shown in Fig. 8.4 and y_p^{-1}, and y_α^{-1} are centers of the corresponding fuzzy sets in Figs. 8.5 and 8.6 (according to the rules in Tables 8.1, 8.2, and 8.3).

Simulation results show that by fuzzy tuning of PID gains, the performance of the controller is enhanced.

Table 8.3 Fuzzy tuning rules for α

		e(n)						
		NB	NM	NS	Z0	PS	PM	PB
Δe(n)	NB	S	S	S	S	S	S	S
	NM	MS	MS	S	S	S	MS	MS
	NS	M	MS	MS	S	MS	MS	M
	Z0	B	M	MS	MS	MS	M	B
	PS	M	MS	MS	S	MS	MS	M
	PM	MS	MS	S	S	S	MS	MS
	PB	S	S	S	S	S	S	S

8.5 Performance Evaluation

To evaluate the performance of the proposed controllers, our methods were implemented by an NS2 simulator [26]. Figure 8.7 shows the topology of our first simulation. In this scenario, the connections between S_i and D_i $(i = 1 \ldots 4)$ are UDP. A constant bit rate (CBR) traffic with a 2.6Mb/s sending rate is attached to each source S_i. CBR sources are ON during all simulation time.

There are four classes of service, and delay constraints are as follows:

$$\frac{D_2}{D_1} = k_1,\ \frac{D_3}{D_2} = k_2,\ \frac{D_4}{D_3} = k_3$$
$$k_1 = k_2 = k_3 = 4$$

Figure 8.8 shows the relative delay of classes (in 0.5s intervals) for an ordinary algorithm using a proportional controller. Figure 8.9 shows the same trends when a PID controller is used, and Fig. 8.10 shows the relative delay of classes when a fuzzy PID controller is used.

To quantitatively compare the algorithms, we define the cost function as follows:

$$Err\% = \frac{100}{3n} \cdot \sum_{t=1}^{n} \sum_{i=1}^{3} \left| \frac{\frac{D_{i+1}(t)}{D_i(t)} - k_i}{k_i} \right| \tag{8.20}$$

According to this equation, Err% of using proportional, PID and fuzzy PID controllers are 15.5, 13.2, and 9.3 percent, respectively. As we can see, the Err% of using PID has been better than those using either an ordinary algorithm or fuzzy online tuning of PID gains.

We complement our test bed measurements by simulating a network with multiple hops and propagation delays. To that effect, we simulated a network with a topology as shown in Fig. 8.11. We have four routers, each with a maximum buffer size of 500 packets, connected by three 45 Mbps links, and sources and sinks connected to the routers by independent 100 Mbps links. Each 45 Mbps link has a propagation delay of 3 ms, and each 100 Mbps link has a propagation delay of 1 ms. There are four classes of traffic. The composition of the traffic mix is given in Table 8.4.

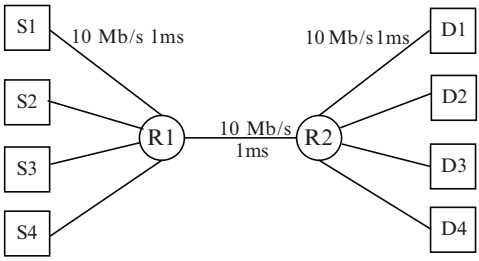


Fig. 8.7 First simulation topology

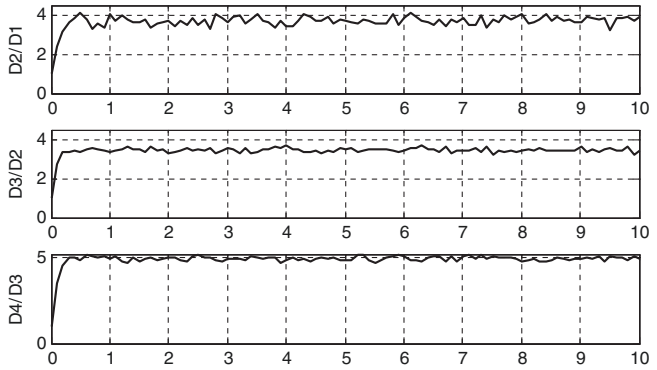


Fig. 8.8 Delay constraints using proportional controller

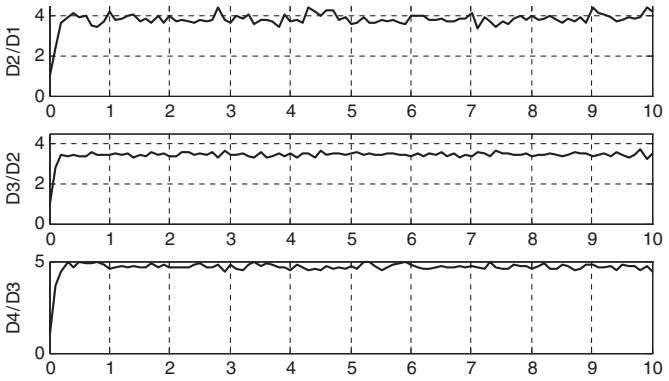


Fig. 8.9 Delay constraints using PID controller

Traffic consists of a mix of TCP (Reno) and UDP flows. TCP flows are either greedy, to model long file transfers, or on-off flows with exponentially distributed on and off periods, to model short, successive file transfers (e.g., HTTP requests). UDP flows are on-off flows using a Pareto distribution for the on and off periods, with a shape parameter = 1.9. All flows start transmitting at time t = 0 s and all flows consist of packets with a fixed size of 500 bytes. The experiment lasts 10 seconds of simulated time.

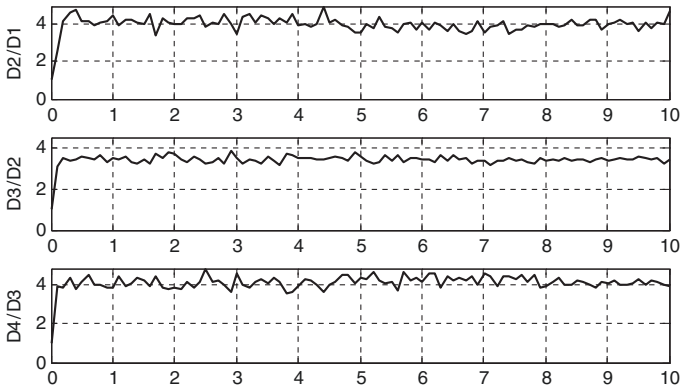


Fig. 8.10 Delay constraints by using fuzzy PID controller

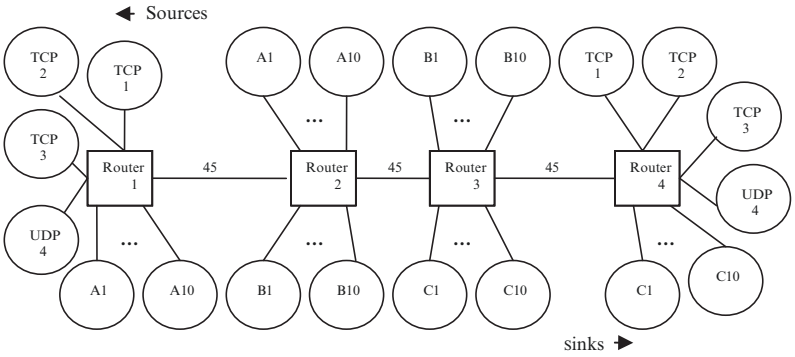


Fig. 8.11 Second simulation topology

Table 8.4 Mixed traffic used in second simulation

Flow	Class	Type				
		Protocol	Traffic	On	Off	Shape
TCP-1	1	TCP	Greedy	N/A	N/A	N/A
TCP-2	2	TCP	Greedy	N/A	N/A	N/A
TCP-3	3	TCP	Greedy	N/A	N/A	N/A
UDP-4	4	UDP	Pareto on-off	10 ms	10 ms	1.9
A-1	1	TCP	Exponential on-off	1000 pkts	200 ms	N/A
A-2,3	2	TCP	Exponential on-off	1000 pkts	200 ms	N/A
A-4,5,6	3	TCP	Exponential on-off	1000 pkts	200 ms	N/A
A-7,8,9,10	4	UDP	Pareto on-off	120 ms	200 ms	1.9

From Table 8.4, classes 1, 2, and 3 consist only of TCP traffic, and Class 4 consists only of UDP traffic. Initially Class 1 contributes 10 percent of the aggregate cross-traffic, Class 2 contributes 20 percent, Class 3 contributes 30 percent, and Class 4 contributes 40 percent.

Here, the delay constraints are as follows:

$$\frac{D_3}{D_2} = k_2, \ \frac{D_4}{D_3} = k_3$$

$$k_2 = k_3 = 3$$

Figure 8.12 shows the relative delay of classes for an ordinary algorithm using a proportional controller. Figure 8.13 shows the same trends when a PID controller is used, and Fig. 8.14 shows the relative delay of classes when a fuzzy PID controller is used.

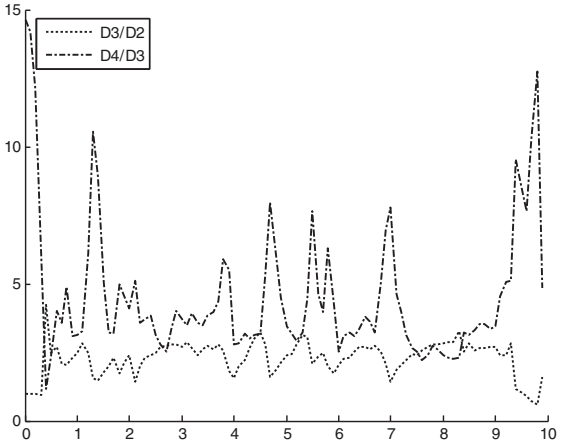


Fig. 8.12 Delay constraints using proportional controller

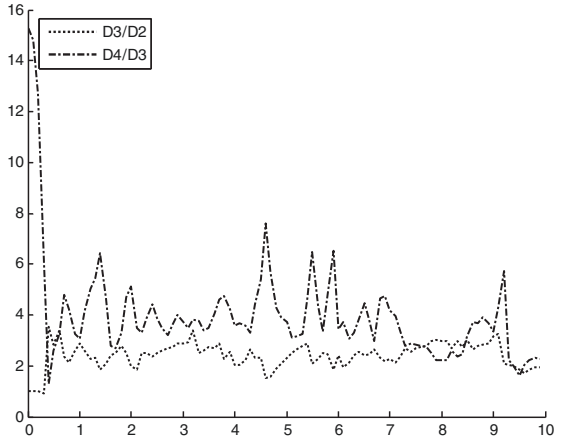


Fig. 8.13 Delay constraints using PID controller

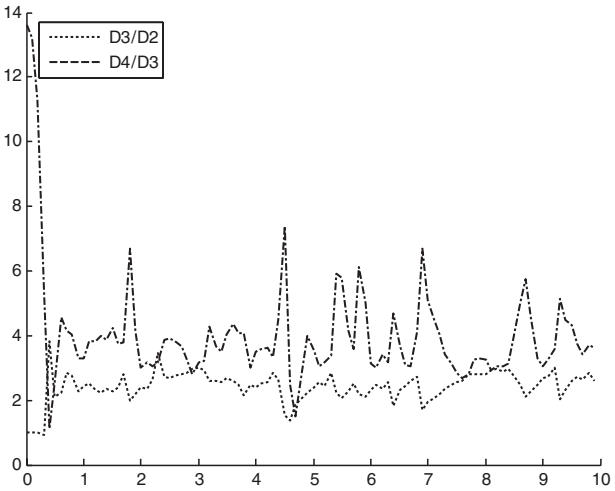


Fig. 8.14 Delay constraints using fuzzy PID controller

8.6 Conclusion

For better achievement in relative delay constraints, a PID controller was developed to control the sending rate of each class. Then we developed a fuzzy method to tune the parameters of the PID controller. Simulation results proved better performance of the scheduler, as expected. When the fuzzy tuning of parameters is done, no matter what the plant is, it ensures that the algorithm behaves properly in unexpected network conditions.

References

1. Demers A, Keshav S, Shenker S (1990) Analysis and simulation of a fair queuing algorithm Journal of Internetworking Research and Experience, pp 3–26, October
2. Parekh A, Gallager R (1992) A generalized processor sharing approach to flow control—the single node case. In: Proceedings of the INFOCOM'92
3. Bennett JCR, Zhang H (1996) WF2Q: Worst-case fair weighted fair queuing. In: Proceedings of IEEE INFOCOM'96, pp 120–128, March
4. Golestani S (1994) A self-clocked fair queuing scheme for broadband applications. In: Proceedings of IEEE INFOCOM'94, Toronto, CA, pp 636–646, June
5. Shreedhar M, Varghese G (1995) Efficient fair queuing using deficit round robin. In: Proceedings of SIGCOMM'95, Boston, MA, pp 231–243, September
6. Anderson TE, Owicki SS, Saxe JB, Thacker CP (1993) High speed switch scheduling for local area networks. ACM, Transactions on Computer Systems, pp 319–352, November
7. Goudreau MW, Kolliopoulos SG, Rao SB (2000) Scheduling algorithms for input-queued switches: randomized techniques and experimental evaluation. In: Proceedings of IEEE INFOCOM'00, Tel Aviv, Israel, pp 1634–1643, March

8. McKeown N, Anantharam V, Walrand J (1996) Achieving 100% throughput in an input-queued switch. In: Proceedings of IEEE INFOCOM'96, San Francisco, CA, pp 296–302, March
9. McKeown N, Izzard M, Mekkittikul A, Ellersick W, Horowitz M (1997) Tiny tera: a packet switch core. IEEE Micro, pp 26–33, January/February
10. Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W (1998) An Architecture for differentiated service. IETF RFC2475, December
11. Li T, Rekhter Y (1998) A Provider architecture for differentiated services and traffic engineering. IETF RFC2430, October
12. Christian N, Liebehrr J, Abdelzaher TF (2001) A Quantitative assured forwarding service. Technical Report, University of Virginia
13. Dovrolis C, Stiliadis D, Ramanathan D (1999) Proportional differentiated services: delay differentiation and packet scheduling. In: Proceedings of ACM SIGCOMM'99, pp 109–120, Boston, MA, August
14. Parekh AK, Gallagher RG (1993) A generalized processor sharing approach to flow control in integrated services networks: the single-node case. IEEE/ACM Transactions on Networking 1(3):344–357, June
15. Athuralia S, Li VH, Low SH, Yin Q (2001) REM: active queue management. IEEE Network, May-June
16. Matta I, Guo L (2000) Differentiated predictive fair service for TCP flows. Technical Report, Computer Science Department, Boston University, May
17. Bodin U, Jonsson A, Schelen O (2001) On creating proportional loss differentiation: predictability and performance. In: Proceedings of IWQoS, Karlsruhe, Germany, pp 372–386
18. Dovrolis C, Ramanathan P (2000) Proportional differentiated services, part II: loss rate differentiation and packet dropping. In: Proceedings of IWQoS, Pittsburgh, pp 52–61
19. Liebeherr J, Christin N (2001) JoBS: Joint buffer management and scheduling for differentiated services. In: Proceedings of IWQoS 2001, Karlsruhe, Germany, pp 404–418, June
20. Firoiu V, Boudec JL, Towsley D, Zhang Z (2001) Advances in internet quality of service. Technical Report
21. Mirjalily G, Saadat R, Ahmadian M (2007) Improving JoBS algorithm using PID and fuzzy PID controllers. In: Proceedings of IMECS'07, Hong Kong, pp 1405–1409
22. Astrom KJ, Hagglund H (1995) PID controllers: theory, design, and tuning, 2nd ed, Instrument Society of America
23. Astrom KJ, Hagglund T, Hang CC, Ho WK (n.d.) Automatic tuning and adaptation for PID controllers: a survey. IFAC J Control Eng Practice 1(4)
24. Zhao ZY, Tomizuka M, Isaka S (1993) Fuzzy gain scheduling of our PID controllers. IEEE Trans On Systems, Man, and Cyber 23(5):1392–1398
25. Wang L-X (1996) A Course in fuzzy systems and control. Prentice-Hall
26. NS-2 network simulator available at *http://www.isi.edu/nsnam/ns*

Chapter 9
Novel Cooperative Transmission Technique of OFDMA Systems with Multiple Antennas for Broadband Wireless Access

Myung-Sun Baek, Young-Hwan You, and Hyoung-Kyu Song

A combination of OFDM with FDMA (called OFDMA) is regarded as a promising solution for improving the performance of BWA with multiple users. Although transmit diversity is clearly advantageous in OFDMA systems, it may not be possible that practical OFDMA systems have more than two antennas due to size, cost, or hardware limitations. Cooperative networks aim to utilize the available terminals as a collection of distributed antennas for the transmission of signals from one end of the network to the other. This paper focuses on design and performance evaluation of IEEE 802.16 MIMO-OFDMA systems with cooperative diversity. In this chapter, for high-rate and reliable transmission, efficient transmission codes of cooperative transmission are proposed for each transmission case (single transmission case or mutual transmission case).

9.1 Introduction

The IEEE 802.16 is an emerging suite of standards for point-to-multipoint (PMP) broadband wireless access (BWA). This BWA through the IEEE 802.16 standard seems to be a promising alternative for last-mile access in crowded urban areas or sub urban areas where installation of cable-based infrastructure is economically or technically infeasible. IEEE 802.16 is likely to emerge as a preeminent technology for cost-competitive ubiquitous broadband wireless access supporting fixed, nomadic, portable and fully mobile operations offering integrated voice, video, and data services [1–3]. The 802.16 standard defines the specifications related to the convergence sublayer (CS), medium access control (MAC) layer, and physical layer (PHY). Its PHY supports four physical modes: WirelessMAN-SC for any applicable

Myung-Sun Baek, Young-Hwan You, and Hyoung-Kyu Song
uT Communication Research Institute, Sejong University
e-mail: sabman@sdc.sejong.ac.kr,yhyou@sejong.ac.kr,songhk@sejong.ac.kr

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

frequencies between 10 and 66 GHz, Wireless-MAN-SCa for licensed frequencies below 11 GHz, WirelessMAN-OFDM for orthogonal frequency-division multiplexing, and Wireless- MAN-OFDMA [1, 4]. OFDMA is based on OFDM and inherits its immunity to inter symbol interference (ISI) and frequency selective fading. Therefore, a combination of OFDM with FDMA (called OFDMA) is regarded as a promising solution for improving the performance of BWA with multiple users. Recently, multiple-input multiple-output (MIMO) techniques using space diversity for OFDMA system were suggested [5, 6] and were selected for IEEE 802.16 standard [1]. Although transmit diversity is clearly advantageous on wireless systems, it may not be possible that practical systems have more than two antennas due to size, cost, or hardware limitations. To overcome the limitation, it has been demonstrated that cooperative diversity provides another effective way of improving spectral and power efficiency of the wireless networks without the additional complexity of multiple antennas [7, 8]. The basic idea behind cooperative diversity rests on the observation that in a wireless environment, the signal transmitted by the source nodes is overheard by other nodes, which can be defined as partners. The source and its partners can jointly process and transmit their information, creating a virtual antenna array. Although many cooperative diversity techniques have been investigated, research for cooperation of MIMO nodes has not been conducted sufficiently. Even with multiple antennas available, cooperative diversity could provide substantial additional diversity benefits. In the OFDMA uplink systems, the cooperative diversity is a very efficient and powerful technique because users use different frequency bands. In this paper, we design a MIMO-OFDMA uplink of an IEEE 802.16 system with cooperative diversity techniques. Space diversity [11, 13] is considered for the MIMO architectures. For high-rate and reliable transmission, efficient transmit codes are proposed for each transmission case (single transmission or mutual transmission). The base station can detect the transmit signal of each user easily, because the transmission codes of cooperative diversity are based on space-time block codes scheme. The outline of this paper is as follows. After describing the OFDMA system model, proposed cooperative diversity techniques are addressed. Simulation results and conclusions are finally shared.

9.2 IEEE 802.16 OFDMA System Model

In OFDMA systems, the available bandwidth (BW) is partitioned into N orthogonal subcarriers and each user accesses a cluster of subcarriers. The N subcarriers include all available subcarriers and virtual subcarriers in the guard band (GB). The OFDMA uplink system with N_u users assigns the d-user the subband defined as $\beta_d = \{k | M_d(d-1) \leq k < M_d \cdot d - 1\}$, where k is a subcarrier index and $M_d = N/N_u$ is the number of subcarriers assigned to each user. For the d-th user, the data symbols $\{S_k^{(d)}\}$ are mapped into the symbol $\{X_k^{(d,\beta_d)}\}$ as

$$X_k^{(d,\beta_d)} = \begin{cases} S_k^{(d)}, & k \in \beta_d \\ 0, & k \notin \beta_d \end{cases} \tag{9.1}$$

which are processed by the inverse fast Fourier transform (IFFT) during the symbol duration T. For transmission, the last N_G samples are preceded as the guard interval (GI) and $N + N_G$ samples are transmitted during the duration $T_S = T + T_G$, where N_G is the length of the GI and T_G is its duration. The transmitted signal of the n-th subcarrier for the d-th user in the time domain is given by

$$x_n^{(d)} = \frac{1}{N} \sum_{k \in \beta_d} X_k^{(d,\beta_d)} exp\left\{ j\frac{2\pi kn}{N} \right\}, -N_G \leq n < N \tag{9.2}$$

The transmitted signal by each user goes through a multipath fading channel and the received signal can be expressed as

$$r^{(n)} = \frac{1}{N} \sum_{d=1}^{N_u} \sum_{k \in \beta_d} H_k^{(d,\beta_d)} X_k^{(d,\beta_d)} exp\left\{ j\frac{2\pi kn}{N} \right\} + w^{(n)} \tag{9.3}$$

where w_n is the zero-mean additive with Gaussian noise (AWGN) with two-sided power spectral density of $N_0/2$, and the discrete frequency response $H_k^{(d,\beta_d)} = \alpha_k^{(d,\beta_d)} e^{j\theta_k^{(d,\beta_d)}}$. It is assumed that the uplink channel response for each user varies independently from all other channels.

Then, the reconstructed FFT output for the k-th subcarrier of d-th user is given by

$$R_k^{(d,\beta_d)} = H_k^{(d,\beta_d)} X_k^{(d,\beta_d)} + W_k \tag{9.4}$$

9.3 MIMO-OFDMA System Model

In this paper, we consider an interactive OFDMA network with both a base station and a subscriber (or mobile) station with two antennas. Both a base station and subscriber transmit the Alamouti's space-time block-coded signal by using two antennas. The transmission matrix for d-th user in the frequency domain is given by

$$\mathbf{G}^{(d,\beta_d)} = \begin{bmatrix} \mathbf{X}_1^{(d,\beta_d)}(1) & \mathbf{X}_2^{(d,\beta_d)}(2) \\ -\mathbf{X}_1^{(d,\beta_d)*}(2) & \mathbf{X}_2^{(d,\beta_d)*}(1) \end{bmatrix} \tag{9.5}$$

where each row is transmitted from a certain antenna at the same time, and each column is transmitted from a same antenna within two time slots. In Eqn. 9.5, $\mathbf{X}_i^{(d,\beta_d)}(l) = [X_{i,0}^{(d,\beta_d)}(l)\ X_{i,1}^{(d,\beta_d)}(l)\ \cdots\ X_{i,k}^{(d,\beta_d)}(l)\ \cdots\ X_{i,M_d}^{(d,\beta_d)}(l)]$ is the l-th OFDMA signal through i-th transmit antenna for d-th user at subband β_d.

The received signal for the d-th user in the frequency domain is as follows:

$$\mathbf{R}^{(d,\beta_d)} = \mathbf{G}^{(d,\beta_d)} \begin{bmatrix} \mathbf{H}_{1,1}^{(d,\beta_d)} & \mathbf{H}_{1,2}^{(d,\beta_d)} \\ \mathbf{H}_{2,1}^{(d,\beta_d)} & \mathbf{H}_{2,2}^{(d,\beta_d)} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix} \tag{9.6}$$

where $\mathbf{H}_{i,j}^{(d,\beta_d)}$ is the frequency response of the multipath fading channel from i-th transmit antenna to j-th receive antenna. Then, the received signal $\mathbf{R}^{(d,\beta_d)}$ can be rewritten as follows:

$$\mathbf{R}^{(d,\beta_d)} = \begin{bmatrix} \mathbf{R}_1^{(d,\beta_d)} & \mathbf{R}_2^{(d,\beta_d)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{(d,\beta_d)}(1) & \mathbf{R}_2^{(d,\beta_d)}(1) \\ \mathbf{R}_1^{(d,\beta_d)}(2) & \mathbf{R}_2^{(d,\beta_d)}(2) \end{bmatrix} \tag{9.7}$$

where $\mathbf{R}_j^{(d,\beta_d)}(l)$ is the l-th received signal through j-th received antenna, which can be written as follows:

$$\mathbf{R}_j^{(d,\beta_d)}(l) = \mathbf{X}_1^{(d,\beta_d)}(l)\mathbf{H}_{1,j}^{(d,\beta_d)} + \mathbf{X}_2^{(d,\beta_d)}(l+1)\mathbf{H}_{2,j}^{(d,\beta_d)}, \tag{9.8}$$

$$\mathbf{R}_j^{(d,\beta_d)}(l+1) = -\mathbf{X}_1^{(d,\beta_d)*}(l+1)\mathbf{H}_{1,j}^{(d,\beta_d)} + \mathbf{X}_2^{(d,\beta_d)*}(l)\mathbf{H}_{2,j}^{(d,\beta_d)}. \tag{9.9}$$

The detection of each signal is the same as ordinary STBC detection.

9.4 Cooperative Transmission Technique for MIMO-OFDMA Uplink System

In this section, we propose some decode-and-forward cooperative techniques for the MIMO-OFDMA uplink. Figure 9.1 shows how the cooperative communication works, where $H_{U1,B}$ and $H_{U2,B}$ are uplink channels from User1 to base station and from User2 to base station, and $H_{U1,U2}$ and $H_{U2,U1}$ are inter user channels from User1 to User2 and from User2 to User1. First, the source device tries to find the best coworker among several candidates. Second, the selected device receives the signals from the source device and executes some process. Third, the processed signals are retransmitted to the base station. Last, the base station receives two independently faded signals. It can be viewed that the source device utilizes antennas of the relay device. In the proposed cooperative communication system for OFDMA uplink, there are two transmission modes. The first transmission mode is single transmission mode. In single transmission mode, only one user transmits its signal and the cooperative partner relays the signal from source user to base station. Second-transmission mode is mutual transmission mode. In mutual transmission mode, each wireless user is assumed to transmit data as well as act as a cooperative agent for another user.

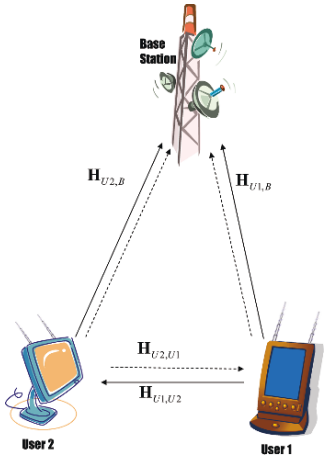


Fig. 9.1 Cooperative communication for MIMO-OFDMA uplink system

9.4.1 Single Transmission Mode

Single transmission mode is executed when only User1 wants to transmit its signal. The relaying node (User2) helps in the communication between User1 and the base station. Thus, User2 receives a noisy version of User1's transmitted signal and transmits a decoded version of this signal to the base station. This cooperation introduces space diversity, because the base station receives two copies, independently faded. As mentioned in Sect. 9.2, all nodes transmit space time coded signals with two transmit antennas. Therefore, User1 transmits the transmit matrix $\mathbf{G}^{(1,\beta_1)}$ in Eqn. 9.5. Then, User2 receives and relays the transmit matrix of User1 to the base station. In this mode, an efficient transmission code is proposed in Table 9.1. In Table 9.1, $\mathbf{X}^{(1,\beta_1)}(l)$ is l-th transmitted OFDMA symbol of User1, and $\hat{\mathbf{X}}^{(1,\beta_2)}(l)$ is the l-th decoded and relayed symbol by User2, where superscript (d,β_m) denotes the transmitted signal of d-th user through the frequency band β_m of m-th user. In Time 1, User1 transmits the signals $\mathbf{X}_1^{(1,\beta_1)}(1)$ and $\mathbf{X}_2^{(1,\beta_1)}(2)$ by using its two antennas, and User2 receives the signal from User1. The signals $-\mathbf{X}_1^{(1,\beta_1)*}(2)$ and $\mathbf{X}_2^{(1,\beta_1)*}(1)$, which are a pair of STBC, are transmitted by User1 in Time 2 and the decoded signals $\hat{\mathbf{X}}_1^{(1,\beta_2)}(1)$ and $\hat{\mathbf{X}}_2^{(1,\beta_2)}(2)$ are relayed by User2 through β_2 (the frequency band of User2) in Time 3. In this technique, full rate transmission is possible, because each user does not use the frequency band of another user.

9.4.2 Mutual Transmission Mode

In this mutual transmission mode, two users transmit their own data and relay data of another user. Table 9.2 represents the efficient transmission code for the mutual

Table 9.1 Cooperative transmission code for signal transmission mode

Time	Frequency band for User (β_1)		Frequency band for User (β_2)	
	User1		User2	
	Tx1	Tx2	Tx1	Tx2
1	$X_1^{(1,\beta_1)}(1)$	$X_2^{(1,\beta_1)}(2)$		
2	$-X_1^{(1,\beta_1)*}(2)$	$X_2^{(1,\beta_1)*}(1)$		
3	$X_1^{(1,\beta_1)}(3)$	$X_2^{(1,\beta_1)}(4)$	$\hat{X}_1^{(1,\beta_2)}(1)$	$\hat{X}_2^{(1,\beta_2)}(2)$
4	$-X_1^{(1,\beta_1)}(4)$	$X_2^{(1,\beta_1)*}(3)$	$-\hat{X}_1^{(1,\beta_2)*}(2)$	$\hat{X}_2^{(1,\beta_2)*}(1)$
$\vdots$	$\vdots$	$\vdots$	$X_1^{(1,\beta_2)}(3)$	$X_2^{(1,\beta_2)}(4)$
$L-1$	$X_1^{(1,\beta_1)}(L-1)$	$X_2^{(1,\beta_1)}(L)$	$-\hat{X}_1^{(1,\beta_2)*}(4)$	$\hat{X}_2^{(1,\beta_2)*}(3)$
L	$-X_1^{(1,\beta_1)*}(L)$	$X_2^{(1,\beta_1)*}(L-1)$	$\vdots$	$\vdots$
$L+1$			$\hat{X}_1^{(1,\beta_2)}(L-1)$	$\hat{X}_2^{(1,\beta_2)}(L)$
$L+2$			$-\hat{X}_1^{(1,\beta_2)*}(L)$	$\hat{X}_2^{(1,\beta_2)*}(L-1)$

Table 9.2 Cooperative transmission code for mutual transmission mode

Time	Frequency band for User (β_1)				Frequency band for User (β_2)			
	User1		User2		User1		User2	
	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
1	$X_1^{(1,\beta_1)}(1)$	$X_2^{(1,\beta_1)}(2)$			$X_1^{(2,\beta_2)}(1)$	$X_2^{(2,\beta_2)}(2)$		
2	$-X_1^{(1,\beta_1)*}(2)$	$X_2^{(1,\beta_1)*}(1)$			$-X_1^{(2,\beta_2)}(2)$	$X_2^{(2,\beta_2)*}(1)$		
3	$X_1^{(1,\beta_1)}(3)$	$X_2^{(1,\beta_1)}(4)$	$\hat{X}_1^{(1,\beta_1)}(1)$	0	$X_1^{(2,\beta_2)}(3)$	$X_2^{(2,\beta_2)}(4)$	$\hat{X}_1^{(2,\beta_2)}(1)$	0
4	$-X_1^{(1,\beta_1)*}(4)$	$X_2^{(1,\beta_1)*}(3)$	0	$-\hat{X}_2^{(1,\beta_1)}(1)$	$-X_1^{(2,\beta_2)*}(4)$	$X_2^{(2,\beta_1)*}(3)$	0	$-\hat{X}_2^{(2,\beta_2)}(1)$
5	$-X_1^{(1,\beta_1)*}(1)$	0	$\hat{X}_1^{(1,\beta_1)*}(3)$	$X_2^{(1,\beta_1)}(4)$	$-X_1^{(2,\beta_2)*}(1)$	0	$\hat{X}_1^{(2,\beta_2)*}(3)$	$\hat{X}_2^{(2,\beta_2)}(4)$
6	0	$-X_2^{(1,\beta_1)*}(1)$	$-\hat{X}_1^{(1,\beta_1)*}(4)$	$\hat{X}_2^{(1,\beta_1)}(3)$	0	$X_2^{(2,\beta_2)*}(1)$	$-\hat{X}_1^{(2,\beta_2)*}(4)$	$\hat{X}_2^{(2,\beta_2)}(3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

transmission mode. In Time 1-2, each user transmits its own space-time block coded signal. This Time is called broadcasting time. In this broadcasting time, each user announces its transmission signal to relay to user. In Time 3-6, each user transmits its own signal through its own frequency band and relays the signal of another user through the frequency band of another user. In the relayed time (Time 3-6), to separate each OFDMA symbol, particular transmission code is required, because transmitted data by User1 and relayed data by User2 are overlapped in the same band. Therefore, STBC of four transmit antenna is used [13]. The mutual transmission mode can not achieve full rate transmission (the transmission rate is 2/3) because the transmitted signal with non-full rate is used to separate the transmitted

signal. The receiver can detect and decode the transmitted signal easily by using STBC decoding.

9.5 Cooperative Receiving Techniques for MIMO-OFDMA Uplink Systems

In this section, receiving techniques for cooperative diversity are presented for each transmission mode (single transmission mode or mutual transmission mode). Because the proposed transmission modes are based on STBC, the receiving techniques are the same as the STBC ML decoding techniques of 2 and 4 transmit antennas.

9.5.1 Single Transmission Mode

This technique uses only 2×2 STBC decoding [11], because each user transmits and relays the space time-coded signal with two transmit antennas in each frequency band. At time slots $l-1$ and l, the base station receives the transmitted signal by User1 (source node) through the frequency band β_1 for User1. The received signal can be presented as follows:

$$\begin{bmatrix} \mathbf{R}_{B_1}^{(1,\beta_1)}(l-1) \\ \mathbf{R}_{B_1}^{(1,\beta_1)}(l) \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^{(1,\beta_1)}(l-1)\ \mathbf{X}_2^{(1,\beta_1)}(l) \\ -\mathbf{X}_1^{(1,\beta_1)*}(l)\ \ \mathbf{X}_2^{(1,\beta_1)*}(l-1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{U1_1,B_1}^{(1,\beta_1)} \\ \mathbf{H}_{U1_2,B_1}^{(1,\beta_1)} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix}, \tag{9.10}$$

$$\begin{bmatrix} \mathbf{R}_{B_2}^{(1,\beta_1)}(l-1) \\ \mathbf{R}_{B_2}^{(1,\beta_1)}(l) \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^{(1,\beta_1)}(l-1)\ \mathbf{X}_2^{(1,\beta_1)}(l) \\ -\mathbf{X}_1^{(1,\beta_1)*}(l)\ \ \mathbf{X}_2^{(1,\beta_1)*}(l-1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{U1_1,B_2}^{(1,\beta_1)} \\ \mathbf{H}_{U1_2,B_2}^{(1,\beta_1)} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix} \tag{9.11}$$

where $\mathbf{H}_{U1_i,B_j}^{(1,\beta_1)}$ is the channel impulse response from i-th transmit antenna of User1 to the j-th receive antenna of base station and $\mathbf{R}_{B_j}^{(1,\beta_1)}$ is the received signal of the j-th receive antenna of base station in the frequency band (β_1) for User1. At the same time, User2 (relay node) receives the same signal from User1. The received signal of User2 can be expressed as

$$\begin{bmatrix} \mathbf{R}_{U2_1}^{(1,\beta_1)}(l-1) \\ \mathbf{R}_{U2_1}^{(1,\beta_1)}(l) \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^{(1,\beta_1)}(l-1)\ \mathbf{X}_2^{(1,\beta_1)}(l) \\ -\mathbf{X}_1^{(1,\beta_1)*}(l)\ \ \mathbf{X}_2^{(1,\beta_1)*}(l-1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{U1_1,U2_1}^{(1,\beta_1)} \\ \mathbf{H}_{U1_2,U2_1}^{(1,\beta_1)} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix}, \tag{9.12}$$

$$\begin{bmatrix} \mathbf{R}_{U2_2}^{(1,\beta_1)}(l-1) \\ \\ \mathbf{R}_{U2_2}^{(1,\beta_1)}(l) \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^{(1,\beta_1)}(l-1)\ \mathbf{X}_2^{(1,\beta_1)}(l) \\ \\ -\mathbf{X}_1^{(1,\beta_1)*}(l)\ \ \mathbf{X}_2^{(1,\beta_1)*}(l-1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{U1_1,U2_2}^{(1,\beta_1)} \\ \\ \mathbf{H}_{U1_2,U2_2}^{(1,\beta_1)} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1 \\ \\ \mathbf{W}_2 \end{bmatrix} \tag{9.13}$$

where $\mathbf{H}_{U1_i,U2_j}^{(1,\beta_1)}$ is the channel impulse response from i-th transmit antenna of User1 to the j-th receive antenna of User2 and $\mathbf{R}_{U2_j}^{(1,\beta_1)}$ is the received signal of j-th receive antenna of User2 in the frequency band (β_1) for User1. The User2 decodes and relays the received signal from User1.

At time slots $l+1$ and $l+2$, the base station receives the transmitted signal $\mathbf{G}^{(1,\beta_1)}(l+1)$ of User1 (source node) in the frequency band β_1 for User1. The transmitted signal of User1 is as follows:

$$\mathbf{G}^{(1,\beta_1)}(l+1) = \begin{bmatrix} \mathbf{X}_1^{(1,\beta_1)}(l+1) & \mathbf{X}_2^{(1,\beta_1)}(l+2) \\ \\ -\mathbf{X}_1^{(1,\beta_1)*}(l+2)\ \mathbf{X}_2^{(1,\beta_1)*}(l+1) \end{bmatrix}. \tag{9.14}$$

At the same time (time slots $l+1$ and $l+2$), the base station receives the relayed signal $\hat{\mathbf{G}}^{(1,\beta_2)}(l)$ of User2 in frequency band β_2 for User2, which can be written as follows:

$$\begin{bmatrix} \mathbf{R}_{B_1}^{(1,\beta_2)}(l-1) \\ \\ \mathbf{R}_{B_1}^{(1,\beta_2)}(l) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{X}}_1^{(1,\beta_2)}(l-1)\ \hat{\mathbf{X}}_2^{(1,\beta_2)}(l) \\ \\ -\hat{\mathbf{X}}_1^{(1,\beta_2)*}(l)\ \ \hat{\mathbf{X}}_2^{(1,\beta_2)*}(l-1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{U2_1,B_1}^{(1,\beta_2)} \\ \\ \mathbf{H}_{U2_2,B_1}^{(1,\beta_2)} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1 \\ \\ \mathbf{W}_2 \end{bmatrix}, \tag{9.15}$$

$$\begin{bmatrix} \mathbf{R}_{U2_2}^{(1,\beta_2)}(l-1) \\ \\ \mathbf{R}_{U2_2}^{(1,\beta_2)}(l) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{X}}_1^{(1,\beta_2)}(l-1)\ \hat{\mathbf{X}}_2^{(1,\beta_2)}(l) \\ \\ -\hat{\mathbf{X}}_1^{(1,\beta_2)*}(l)\ \ \hat{\mathbf{X}}_2^{(1,\beta_2)*}(l-1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{U2_1,B2_2}^{(1,\beta_2)} \\ \\ \mathbf{H}_{U2_2,B_2}^{(1,\beta_2)} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1 \\ \\ \mathbf{W}_2 \end{bmatrix} \tag{9.16}$$

where $\mathbf{H}_{U2_i,B_j}^{(1,\beta_2)}$ is the channel impulse response from i-th transmit antenna of User2 to the j-th receive antenna of base station and $\mathbf{R}_{B_j}^{(1,\beta_2)}$ received signal of j-th receive antenna in the frequency band (β_2) for User2. The base station decodes and combines the received signal of User2 with the decoded signal of User1.

9.5.2 Mutual Transmission Mode

In this mutual technique, 2×2 and 4×2 STBC decoding schemes are used. As shown in Table 9.2a, 2×2 STBC decoding is used 1-2 times, and 4×2 STBC decoding is used 3-6 times. In time slots 1 and 2, the received signal of the base

station is the same as Eqn. 9.11 in a frequency band of User1. At time slots 3-6, the received signal of the first antenna of the base station is provided as

$$\begin{bmatrix} \mathbf{R}_{B_1}^{(1,\beta_1)}(3) \\ \mathbf{R}_{B_1}^{(1,\beta_1)}(4) \\ \mathbf{R}_{B_1}^{(1,\beta_1)}(5) \\ \mathbf{R}_{B_1}^{(1,\beta_1)}(6) \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^{(1,\beta_2)}(3) & \mathbf{X}_2^{(1,\beta_2)}(4) & \hat{\mathbf{X}}_1^{(1,\beta_2)}(1) & 0 \\ \mathbf{X}_1^{(1,\beta_2)*}(4) & \mathbf{X}_2^{(1,\beta_2)*}(3) & 0 & -\hat{\mathbf{X}}_2^{(1,\beta_2)}(1) \\ -\mathbf{X}_1^{(1,\beta_2)*}(1) & 0 & \hat{\mathbf{X}}_1^{(1,\beta_2)*}(3) & \hat{\mathbf{X}}_2^{(1,\beta_2)}(4) \\ 0 & \mathbf{X}_2^{(1,\beta_2)*}(1) & -\hat{\mathbf{X}}_1^{(1,\beta_2)*}(4) & \hat{\mathbf{X}}_2^{(1,\beta_2)}(3) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{U1_1,B_1}^{(1,\beta_1)} \\ \mathbf{H}_{U1_2,B_1}^{(1,\beta_1)} \\ \mathbf{H}_{U2_1,B_1}^{(1,\beta_1)} \\ \mathbf{H}_{U2_2,B_1}^{(1,\beta_1)} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix} \tag{9.17}$$

Using the same method, the received signal $[\mathbf{R}_{B_2}^{(1,\beta_1)}(3) \cdots \mathbf{R}_{B_2}^{(1,\beta_1)}(6)]$ of the second antenna and the received signals $[\mathbf{R}_{B_1}^{(1,\beta_2)}(3) \cdots \mathbf{R}_{B_1}^{(1,\beta_2)}(6)]$ and $[\mathbf{R}_{B_2}^{(1,\beta_2)}(3) \cdots \mathbf{R}_{B_2}^{(1,\beta_2)}(6)]$ of frequency band β_2 for User2 can be obtained. The received signal can be detected by using conventional STBC ML decoding [11, 13].

9.6 Simulations

To simulate the cooperative MIMO-OFDMA system performance, a Rayleigh fading channel, without LoS, and a Rician fading channel with LoS of ETSI is used. The entire BW of 20MHz is divided into 256 subchannels, and each user occupies 32 successive subcarriers (i.e., $N = 256, N_u = 32$, and $M_d = 8$). The first and second users are also regarded as the cooperative users. In this simulation, we assume that the SNR values of uplink channels are the same ('SNR of $\mathbf{H}_{U1,B}$' = 'SNR of $\mathbf{H}_{U2,B}$' = SNR_B), and the SNR values of inter-user channels are the same ('SNR of $\mathbf{H}_{U1,U2}$' = 'SNR of $\mathbf{H}_{U2,U1}$' = SNR_U). And, we define the SNR_D as SNR deference value between the uplink channel SNR (SNR_B) and the inter-user channel SNR (SNR_U). The SNR_D can be expressed as

$$SNR_D = SNR_U - SNR_B \tag{9.18}$$

The high SNR_D denotes that the inter-user channel SNR (SNR_U) is much better than uplink channel SNR (SNR_B).

Figures 9.2 and 9.3 show the BER performance of cooperative diversity techniques with both single transmission mode and mutual transmission mode according to the SNR_D. In the figures, if the SNR_D is small (which means that the uplink

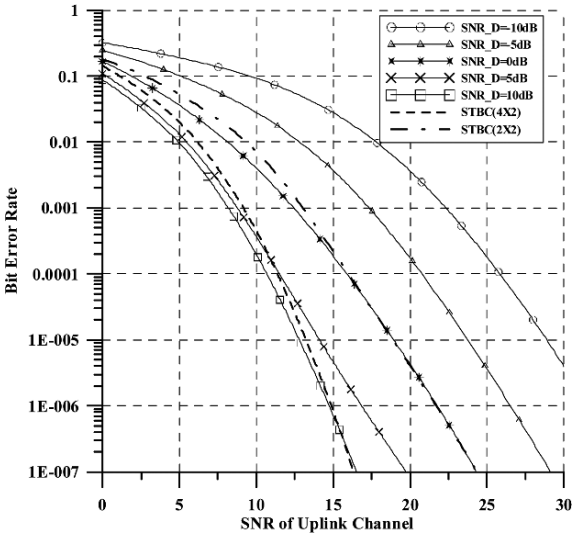


Fig. 9.2 BER performance of single transmission mode according to SNR_D

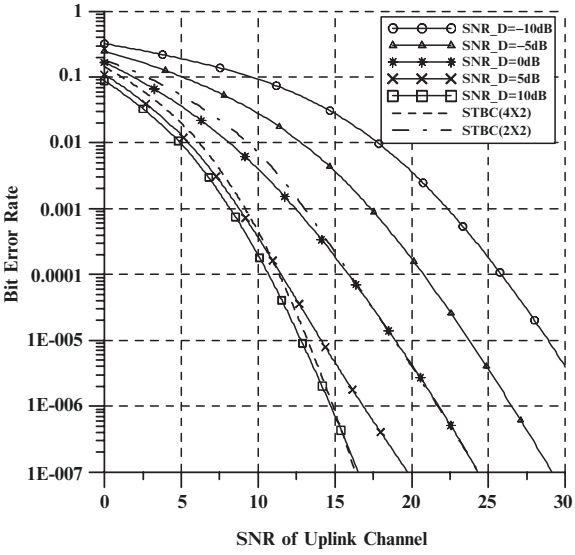


Fig. 9.3 BER performance of double transmission mode according to SNR_D

SNR is very higher than SNR_U) the cooperative communication mode gives a bad outcome. When the SNR_D is higher than 0dB, the proposed scheme gives a better performance than a standard 2×2 STBC system. We find the fact that the proposed techniques with $SNR_D = 10$dB has almost the same performance as a system that uses STBC with 4×2. The performances of both single transmission mode (Fig. 9.2) and mutual transmission mode (Fig. 9.3) are very similar. The reason why the performances of single transmission mode and mutual transmission mode are similar is that the diversity gain of the two transmission modes (single transmission mode and mutual transmission mode) is the same (diversity gain $D = 8 : 4\text{Tx} \times 2\text{Rx}$). However, the single transmission mode has full rate and simple decoding. It can be deduced that the selection of the best coworker is a problem of searching for the device that has high channel SNR.

Figure 9.4 illustrates the effect of the change of the SNR_D on the BER performance of cooperative diversity techniques. In the case of $SNR_D > 0$dB, the two cooperation modes have better performance than that of 2×2 STBC-OFDMA system; and in the case of $SNR_D > 10$dB, the two cooperation mode has better performance than that of 4×2 STBC-OFDMA system. As SNR_D increases, the performances of two cooperation modes become better. However, when $SNR_D > 10$dB, the performance improvement is decreased and the performance converges on the performance of 4×2 STBC. In the general OFDMA uplink system, the distance of inter-user is much lower than that of user to base station—the SNR_U is much higher than that the SNR_B. Therefore, the proposed cooperative technique can improve the MIMO-OFDMA uplink system performance.

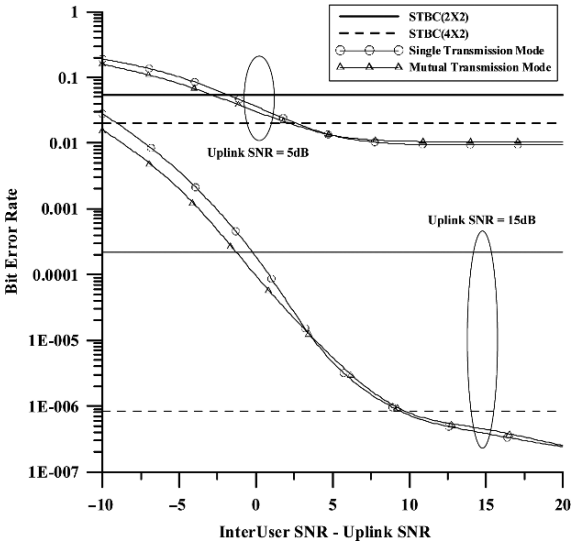


Fig. 9.4 BER performance according to SNR_D for proposed scheme

9.7 Conclusions

MIMO-OFDMA is a very appealing technique for achieving high bit-rate data transmission and providing significant capacity gain in wireless channels. However, it may not be possible that practical OFDMA systems have more than two antennas due to size, cost, or hardware limitations. In this chapter, we presented the cooperative diversity techniques in a MIMO OFDMA-based IEEE 802.16 system. From the results presented above, we showed that the IEEE 802.16 system with proposed cooperative diversity techniques has very high performance gain when the SNR of inter-user channels is higher than uplink channels. The presented results are also applied for several OFDMA-based systems such as digital audio broadcasting return channel terrestrial (DVB-RCT) and Korean wireless broadband (Wibro).

Acknowledgements This research is supported by the Ubiquitous Computing Network (UCN) Project, the Ministry of Information and Communication (MIC) and the 21st Century Frontier R&D Program in Korea.

References

1. Wireless MAN working group, http://wirelessman.org/
2. Iyengar R, Iyer P, Sikdar B (2005) Analysis of 802.16 based last mile wireless networks. IEEE GLOBECOM '05, vol 5, no 28, November pp. 3123–3127
3. Niyato D, Hossain E (2005) Queue-aware uplink bandwidth allocation for polling services in 802.16 broadband wireless networks. IEEE GLOBECOM '05, vol. 6, no 28, December pp. 3702–3706
4. Taesoo Kwon, Howon Lee, Sik Choi, Juyeop Kim, Dong-Ho Cho, Sunghyn Cho, Sangboh Yun, Won-Hyong Park, Kiho Kim (2005) Design and implementation of a simulator based on a cross-layer protocol between MAC and PHY layers in a WiBro Compatible IEEE 802.16e OFDMA system. IEEE Communication Magazine, vol 43, no 12, December pp 136–146
5. Stamoulis A, Zhiqiang L, Giannakis GB (2002) Space-time block-coded OFDMA with linear precoding for multirate services. Signal Processing, IEEE Trans. on, vol 50, no. 1, January pp. 119–129
6. Jung Min Choi, Kwak JS, Ho Seok Kim, Jae Hong Lee (2004) Adaptive subcarrier, bit, and power allocation algorithm for MIMO-OFDMA system. VTC 2004-Spring. 2004 IEEE, vol. 3, no. 17-19, May pp. 1801–1805
7. Nosratinia A, Hunter TE, Hedayat A (2004) Cooperative communication in wireless networks. IEEE Communication Magazine, October pp. 74–80
8. Canpolat O, Uysal M (2004) Super-orthogonal space-time trellis coded cooperative diversity systems. VTC 2004-Fall IEEE, vol. 4, Sep. pp. 2429–2433
9. Lee J-C, Han Y-H, Shin M-K, Jang H-J, Kim H-J (2006) Considerations of neighbor discovery protocol over IEEE 802.16 networks. ICACT 2006, vol. 2, no. 20–22, February pp. 951–955
10. Hongseok Jeon, Junghoon Jee (2006) IPv6 neighbor discovery protocol for common prefix allocation in IEEE 802.16. *ICACT 2006*, vol 3, no. 20–22, February pp. 1661–1663
11. Alamouti SM (1998) A simple transmit diversity technique for wireless communications. IEEE JSEC, vol. 16, no. 8, Oct. pp 1451–1458
12. Tarokh V, Jafarkhani H, Calderbank AR (1999) Space-time block codes from orthogonal designs. IEEE Trans. Info. Theory, vol. 45, no. 5, Jul. pp. 1456–1467

13. Trikkonen O, Hottinene A (2002) Square-matrix embeddable space-time block codes for complex signal constellations. IEEE Trans. Info. Theory, vol. 48, no. 2, February pp. 384–395
14. ETSI (1997) Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television (DVB-T) European Telecommunication Standard. ETS 300 744, Mar.

Chapter 10
HJM Tree for Security Analyses of Passive RFID Systems

Xu Huang

10.1 Introduction

Radio frequency identification (RFID) technology is not fundamentally new and concerns a whole range of applications. The first RFID application may have been the Royal British Air Force's "Identify Friend or Foe" system, which was used during the Second World War to identify friendly aircraft. RFID can be applied to a variety of tasks, structures, work systems, and contexts along the value chain, including business-to-business (B-2-B) logistics, internal operations, business-to-consumer (B-2-C) marketing, and after-sales service applications [1–6]. However, the boom that RFID technology enjoys today is basically due to the standardization [7] and development of low cost devices.

Like every wireless device, RFID systems bring with them security and privacy issues to all those people who have been working in this area. Security issues involve classic attacks, namely denial of service, impersonation of tags, or channel eavesdropping. These attacks are rendered more practicable because of the tags' lack of computational and storage capacity. There are many papers investigating these issues in various ways [7, 8, 10–12, 14]. Today's challenge is to find protocols (or deployments) which allow authorized parties to identify the tags without an adversary being able to track them, thus getting to the root of the privacy problem [8, 13]. It is well known that the reason not to use well known authentication protocols is that such protocols do not preserve the privacy of the provider. Asymmetric cryptography could easily solve this problem, but it is too heavy to be implemented within a tag.

Avoine [8] showed that if n is the number of tags managed by the system, $O(n)$ cryptographic operations are required in order to identify one tag for a traditional method; while Molnar and Wagner [9] make a reduction of the system's workload

Xu Huang
School of Information Sciences and Engineering, University of Canberra, ACT, 2617 Australia
e-mail: Xu.Huang@canberra.edu.au

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

from $O(n)$ to $O(\log n)$. Also Ohkubo et al. [12] present a protocol, relying on hash chains, that can result in a comparable workload for the system without degrading privacy, in comparison with the method provided by Molnar and Wagner [9].

Considering that the method presented by Ohkubo et al requires constructing hash chains from each n initial value [12], the lifetime of the tag is a priori limited to m identifications; when a tag is scanned by a reader, its field in the database can be refreshed. Therefore, the threshold m is the number of read operations on a single tag between two updates of the database. So even if an attacker can tamper with a tag, she/he will not be able to track its past events. However, the whole cost increase must be the tradeoff for nondegrading privacy.

In this paper, in order to keep the tag system lower in cost and reasonably higher in security, we first focus on the popular "tree-based" technique to establish the "nonrecombining tree," extended from the HJM model in the area of finance [15]. The simulation results show that the nonrecombining tree system deploying passive RFID succeeds to decrease significantly the whole attack succeeds (much less than 0.01 percent) and makes the whole system a closer real system.

10.2 Tree-Based Technique

In the following analyses, in order to compare the results with [8, 9], we are going to use the same system, namely a system that relies on a single computer that takes $\theta = 2{-}^{23}$ second to carry out a cryptographic operation, either hashing or encrypting 128-bit blocks. The tag number is 2^{20}. Identification of several tags is therefore sequential. Current implementations allow a single reader to read several hundred tags per second, which means that the system should spend at the most a few milliseconds to identify one tag.

Let ID (identification) be the tag's identifier that is stored in both the database of the system and the tag. They also share a secret key s. To initiate the authentication, the reader sends a random a to the tag. Next, the tag picks a random b and answers $\sigma. = \mathrm{ID} \oplus f_S(0,a,b)$, where f_S is a pseudorandom function. We show this protocol in Fig. 10.1.

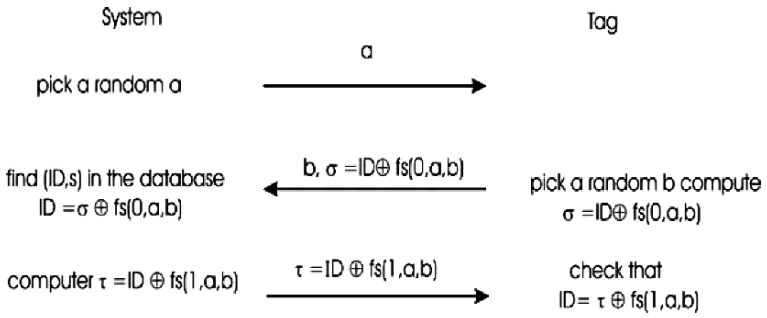


Fig. 10.1 The protocol of Molnar and Wagner [9]

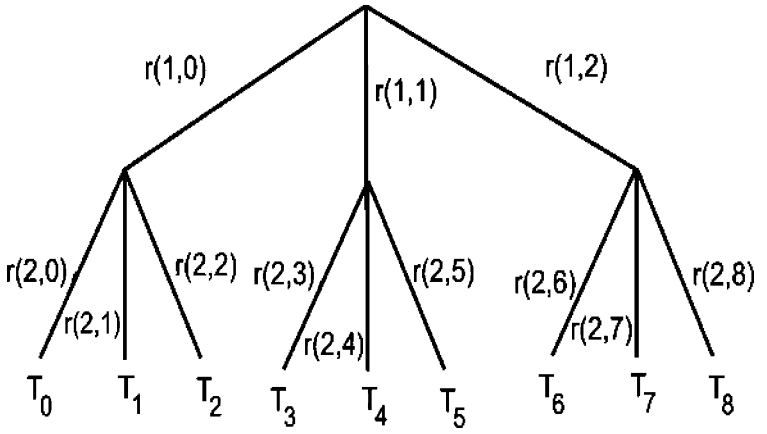


Fig. 10.2 A tree of tag information (secrets)

The technique suggested by Molnar and Wagner [9] relies on a tree structure in order to reduce the identification complexity.

As an example, Fig. 10.2 shows nine tags with two levels, and each tag has its information or "secret" that shows its position; for example, T_7 has its information [r(1,2), r(2,7)], which means that tag 7 can be located by level 1 branch 2 and level 2 branch 7. If the information for any tag can be found by an attacker, we define that the attack is successful and that the security of the tag has failed.

Let us consider the general case for the passive multi-grouped tags system. We let n be the number of tags managed by the system and $l := [\log_\delta \mathrm{n}]$ be the depth of the tree with a branching factor δ. Each edge in the tree is given a value with a randomly chosen secret, $r_{\mathrm{i,j}}$, where i is the level in the tree and j is the number of the branch. The average number of cryptographic operations required to identify one tag for traditional n tags is $n/2$ and we need $t_{\mathrm{tr}} = (n\theta)/2$. Here θ is the time taken by the computer to identify one tag in the system. For a passive multi-grouped tags system at each level i, the system has to search in a set of δ secrets for the one matching the tag's secret. Given that $[s_1, \ldots, s_1]$ denotes a secret for a tag, the system has thus to compute $\delta/2$ times $f_{\mathrm{s}}(0, a, b)$ on average at level i, meaning that $\delta l/2$ operations are required in order to identify one tag. We have $t_{\mathrm{MW}} = [(\delta\theta)/2] \log_\delta n$. Here t_{MW} denotes the time taken under the protocol of Molnar and Wagner.

When an attacker is able to tamper with one tag, there are three possible situations: (1) The attacker has one tag T_0 she/he can tamper with and thus obtain its complete secret (e.g., as the owner). For the sake of calculation simplicity, we assume that T_0 is put back and does not significantly affect the results. (2) The attacker chooses a target tag T and can query it as much as she/he wants but not tamper with it. (3) Given two tags T_1 and T_2 and $T \in \{T_1, T_2\}$, if the attacker can definitely know which one is T, he/she can succeed via querying, not tampering with them. We assume that the attacker cannot carry out an exhaustive search over the secret space. His only guess of a secret block of a given tag is to query it with

the blocks of secret he obtained by tampering with some tags. When she/he tampers with only one tag and obtains only one block of secret per level in the tree. If either T_1 or T_2 (but not both) have the same block as T_0, she/he is able to determine which of them is T. If neither T_1 nor T_2 has the same block as T_0, he/she cannot answer, even though she/he can move on to the next level of the tree because the authentication of the reader succeeds. We denote the secrets of T, T_0, T_1, and T_2 by $[s_1,\ldots,s_1]$, $[s^0{}_1,\ldots,s^0{}_1]$, $[s^1{}_1,\ldots,s^1{}_1]$, and $[s^2{}_1,\ldots,s^2{}_1]$, respectively. We consider a given level i where $s^1{}_\mathrm{i}$ and $s^2{}_\mathrm{i}$ are in the same subtree. We have four issues:

$$C_1^1 = ((s_1^0 = s_i^1) \wedge (s_i^0 \neq s_1^2)) \tag{10.1}$$

$$\mathrm{C}_\mathrm{i}^2 = ((s_i^0 \neq s_i^1) \wedge (s_i^0 = s_i^2)) \tag{10.2}$$

$$C_i^3 = ((s_i^0 \neq s_i^1) \wedge (s_i^0 \neq s_i^2)) \tag{10.3}$$

$$C_i^4 = (s_i^0 = s_i^1 = s_i^2) \tag{10.4}$$

where

(1) = the attacker succeeds,
(2) = the attacker succeeds,
(3) = the attacker definitively fails, and
(4) = the attacker fails at the level i but can move onto level $i+1$.

When the number of tags in the system is large, we assume that:

$$P(C_i^1) = P(s_i^0 = s_i^1) \times P(s_i^0 \neq s_i^{2)}). \tag{10.5}$$

The same assumption also applies to $C^2{}_\mathrm{i}$, $C^3{}_\mathrm{i}$, and $C^4{}_\mathrm{i}$, thus we have:

$$P(C_i^1 \vee C_i^2) = \frac{2(\delta-1)}{\delta^2} (1 \leq i \leq l) \text{ and } P(C_i^4) = \frac{1}{\delta^2}. \tag{10.6}$$

The overall probability P that the whole attack succeeds is as follows:

$$P(C_1^1 \vee C_1^2) + \sum_{i=2}^{l} \left[(P(C_i^1 \vee C_i^2) \times \prod_{j=1}^{i-1} P(C_j^4) \right] = 2(\delta-1) \frac{1-(\frac{1}{\delta^2})^l}{1-\frac{1}{\delta^2}} \frac{1}{\delta^2} \tag{10.7}$$

Because of $\delta^1 = n$, we have:

$$P = \frac{2}{\delta+1}(1-\frac{1}{n^2}). \tag{10.8}$$

When an attacker is able to tamper with more than one tag with some tag, we have five cases as shown below, where $\mathrm{K_i}$ denotes the set of blocks of this (one-level) subtree that are known by the attacker and Λ_i denotes the set of those which are unknown by the attacker. k_i denotes the number of blocks in $\mathrm{K_i}$.

$C_i^1 = ((s_i^1 \in \mathrm{K}_i) \wedge (s_i^2 \in \Lambda_i))$ then the attack succeeds
$C_i^2 = ((s_i^1 \in \Lambda_i) \wedge (s_i^2 \in \mathrm{K}_i))$ then the attack succeeds
$C_i^3 = ((s_i^1 \in \mathrm{K}_i) \wedge (s_i^2 \in \mathrm{K}_i) \wedge (s_i^1 \neq \mathrm{s_i}^2))$ then the attack succeeds
$C_i^4 = ((s_i^1 \in \Lambda_i) \wedge (s_i^2 \in \Lambda_i))$ then the attacks definitively fail
$C_i^5 = ((s_i^1 \in \mathrm{K}_i) \wedge (s_i^2 \in \mathrm{K}_i) \wedge^* (s_i^1 = \mathrm{s_i}^2))$ then the attacks at level i fails but can go level $\mathrm{i}+1$

Thus, following [9], we have all i such that $1 \leq i \leq l$:

$$P(C_i^1 \vee C_i^2 \vee C_i^3) = \frac{2k_i}{\delta}\left(1 - \frac{k_i}{\delta}\right) + \left(\frac{k_i}{\delta}\right)^2 \left(1 - \frac{1}{k_i}\right) \text{ and}$$

$$P(C_i^5) = \frac{k_i}{\delta^2} \tag{10.9}$$

So the overall probability P that the attack succeeds is obtained as below:

$$P = P(C_1^1 \vee C_1^2 \vee C_1^3) + \sum_{i=2}^{l}\left[(P(C_1^1 \vee C_1^2 \vee C_1^3) \times \prod_{j=1}^{i-1} P(C_j^5)\right] \tag{10.10}$$

$$= \frac{k_1}{\delta^2}(2\delta - k_1^- 1) + \sum_{i=2}^{l}\left[\frac{k_i}{\delta^2}(2\delta - k_i - 1)\prod_{j=1}^{i=1}\frac{k_j}{\delta^2}\right]$$

We have the relationship between k_1 and k_0, the number of tags tampered with by the attacker at the level 0, as follows:

$$k_1 = \delta\left[1 - \left(1 - \frac{1}{\delta}\right)^{k_0}\right] \text{ and } k_i = \delta\left[1 - \left(1 - \frac{1}{\delta}\right)^{g(k_i)}\right] (2 \leq i \leq l) \tag{10.11}$$

$$\text{with } g(k_i) = k_0 \prod_{j=1}^{i-1}\frac{1}{k_j}.$$

10.3 Multi-Grouped Tree Technique

Before we discuss the multi-grouped tree technique, we need to check the time taken for searching the whole system that was discussed in our previous section. If we follow the normal basic secret-key challenge-response protocol, the order of n cryptographic operations is required in order to identify one tag in the n tags system. If we use the technique suggested by Molnar and Wagner [9], in order to identify one tag the system with n tags has to take the total time period as $t_{\mathrm{MW}} = [(\delta\theta)/2]\log_\delta n$, as shown above. It is noted that the factor δ is a branching factor, which describes how many branch numbers there are for each group. One example is shown in Fig. 10.2, where $\delta = 3$. The shorter the time reaction for the deployed RFID system the better. However, from equation (10.10) we know that if δ is small

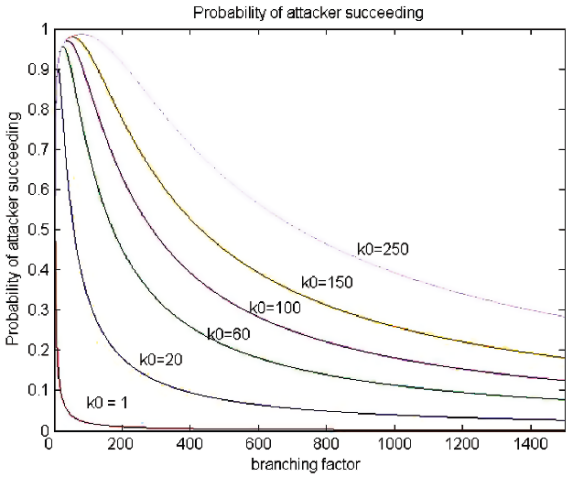


Fig. 10.3 The probability of an attacker succeeding for the different branching factors under different $k_0 (k_0 = 1,\ 20,\ 60,\ 100,\ 150,$ and $250)$

the overall probability that the attacker will succeed will be rapidly increased due to the probability that an attacker succeeds is inverse proportion to δ^2. Therefore on the one hand we do not want too much time to identify a tag; on the other hand we do want to decrease the probability of an attacker succeeding.

This implies that we have some trade-off between those two parameters, in order to set the whole system to an optimum condition that has a reasonably shorter reaction time period and also a reasonable security level.

Figure 10.3 shows the probability of an attacker succeeding at different k_0s$(k_0 = 1,\ 20,\ 40,\ 60,\ 100,\ 150,$ and $250)$ with the branching factors changing up to 1600. It clearly shows that when the branching factor is small the system would be extremely insecure (almost regardless of the "level factors").

Therefore we should always think about increasing the branch numbers to make sure the system is at a reasonable security level. It is also obvious that if the attacker has many tags to be tampered with, he/she will obtain enough information to get what he/she wants, as is shown in the different k_0 curves in Fig. 10.3. If we want to keep a RFID system sitting at a reasonable security level, we need to keep the k_0 as small as possible, which means we need to make it harder for an attacker to find a tag. As an example, it is easy to show that if k_0 is about 250, the system will be in a vulnerability state (the probability of an attacker succeeding is $>60\%$) regardless of the branching factor, because the attacker has enough information to break the system.

Figure 10.4 shows that the probability of an attacker succeeding is controlled by k_0 and branching factors δ. Also we can see that if the branching number is very small, the system would be extremely vulnerable $(>70\%)$ no matter what k_0 is, which means the attacker could use even a very small sample to obtain the information to break the system.

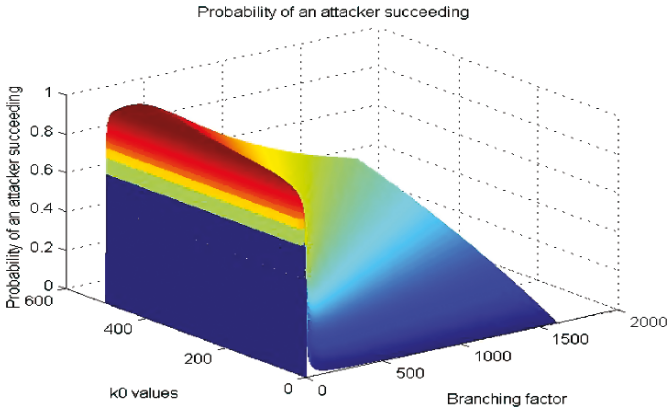


Fig. 10.4 The probability of an attacker succeeding is controlled by k_0 and branching factors

Therefore, from both figures we have learned that we need to build a multi-grouped tree keeping k_0 as small as possible (making it harder to find information for a tag in a system) and keeping the branching factor as big as we can, provided that the whole time taken meets the required conditions. In fact we may intuitively make a design for a RFID tree system to trade off the time taken and the branch factor to take the branch number after the turning point in Fig. 10.3.

Next we are going to introduce the Heath, Jarrow, and Morton (HJM) model in its original background, namely the "interest rate derivatives." Then we will obtain the "HJM tree" that is to be introduced into passive RFID systems for security analyses.

Before we introduce the HJM model, we have to explain some notations we are going to use as shown below:

$P(t,T)$:	price at time t of a zero-coupon bond with a principal of \$1.00 maturing at time T.
Ω_t:	vector of past and present values of interest rates and bond prices at time t that are relevant for determining bond price volatilities at that time.
$v(t,T,\Omega_t)$:	volatility of $P(t,T)$
$f(t,T_1,T_2)$	forward rate as seen at time t for the period between time T_1 and T_2
$r(t)$	short-term risk-free interest rate at time t
$dz(t)$	Wiener process driving term structure movements

Also we define the variable $F(t,T)$ as the limit of $f(t,T,T+\delta t)$ as it tends to zero.

When we assume just one factor, the risk-neutral process for $P(t,T)$ has the form shown below:

$$dP(t,T) = r(t)P(t,T)dt + v(t,T,\Omega_t)P(t,T)dz(t). \tag{10.12}$$

The expected return is $r(t)$ because a zero-coupon bond is a traded security providing no income. As its argument Ω_t indicates, the volatility, v, can in the most general

form of the model be any well-behaved function of past and present interest rates and bond prices. Because a bond's price volatility declines to zero at maturity, we must have:

$$v(t,T,\Omega_t) = 0. \tag{10.13}$$

Here the $v(t,T,\Omega_t) = 0$ condition is equivalent to the assumption that all discount bonds have finite drifts at all times. If the volatility of the bond does not decline to zero at maturity, an infinite drift may be necessary to ensure that the bond's price equals its face value at maturity.

$f(t,T_1,T_2)$ can be related to zero-coupon bond prices as follows:

$$f(t,T_1,T_2) = \frac{\ln[P(t,T_1)] - \ln[P(t,T_2)]}{T_2 - T_1}. \tag{10.14}$$

From equation (10.12) and Itô's lemma, we have:

$$d\ln[P(t,T_1)] = (r(t) - \frac{v(t,T_1,\Omega_t)^2}{2}dt + v(t,T_1,\Omega_t)dz(t). \tag{10.15}$$

And we have:

$$d\ln[P(t,T_2)] = (r(t) - \frac{v(t,T_2,\Omega_t)^2}{2}dt + v(t,T_2,\Omega_t)dz(t), \tag{10.16}$$

so that we have:

$$df(t,T_1,T_2) = \frac{v(t,T_2,\Omega_t)^2 - v(t,T_1,\Omega_t)^2}{2(T_2 - T_1)}dt + \frac{v(t,T_1,\Omega_t) - v(t,T_2,\Omega_t)}{T_2 - T_1}dz(t) \tag{10.17}$$

Equation (10.17) shows that the risk-neutral process for f depends solely on the v's; it depends on r and the P's only to the extent that the v's themselves depend on these variables.

When we put $T_1 = T$ and $T_2 = T + \delta$T into equation (10.17) and then take the limits as δT tends to zero, $f(t,T_1,T_2)$ becomes $F(t,T)$, the coefficient of $dz(t)$ becomes $v_{\mathrm{T}}(t,T,\Omega_t)$, and the coefficient of dt becomes:

$$\frac{1}{2}\frac{\partial[v(t,T,\Omega_t)^2]}{\partial T} = v(t,T,\Omega_t)v_T(t,T,\Omega_t), \tag{10.18}$$

where the subscript to v denotes a partial derivative. It follows that:

$$dF(t,T) = v(t,T,\Omega_t)v_T(t,T,\Omega_t)dt - v_T(t,T,\Omega_t)dz(t) \tag{10.19}$$

Once the function $v(t,T,\Omega_t)$ has been specified, the risk-neutral processes for the $F(t,T)$'s are known.

Equation (10.19) shows that there is a link between the drift and standard deviation of an instantaneous forward rate. Heath, Jarrow, and Morton were the first to point this out [16]. Integrating $v_\tau(t,\tau,\Omega_t)$ between $\tau = t$ and $\tau = T$, we obtain:

$$v(t,T,\Omega_t) - v(t,t,\Omega_t) = \int_t^T v_\tau(t,\tau,\Omega_t)d\tau. \tag{10.20}$$

Because $v(t,T,\Omega_\mathrm{t}) = 0$ this becomes:

$$v(t,T,\Omega_t) - v(t,t,\Omega_t) = \int_t^T v_\tau(t,\tau,\Omega_t)d\tau. \tag{10.21}$$

If $m(t,T,\Omega_\mathrm{t})$ and $s(t,T,\Omega_\mathrm{t})$ are the instantaneous drift and standard deviation of $F(t,T)$, so that:

$$dF(t,T) = m(t,T,\Omega_t)dt + s(t,T,\Omega_t)dz, \tag{10.22}$$

then it follows from equation (10.19) that:

$$m(t,T,\Omega_t) = s(t,T,\Omega_t)\int_t^T s(t,\tau,\Omega_t)d\tau. \tag{10.23}$$

We now consider the one-factor continuous-time model for forward rate in equation (10.19). Because

$$F(t,t) = F(0,t) + \int_0^t dF(\tau,t), \tag{10.24}$$

and $r(t) = F(t,t)$, it follows from equation (10.19) that:

$$r(t) = F(0,t) + \int_0^t v(\tau,t,\Omega_\tau)v_t(\tau,t,\Omega_t)d\tau + \int_0^t v_t(\tau,t,\Omega_\tau)dz(\tau) \tag{10.25}$$

In the stochastic calculus we can interpret what is going on by replacing the integral signs with summation signs and d's with δ's. For example,

$$\int_0^t v(\tau,t,\Omega_\tau)v_t(\tau,t,\Omega_\tau)d\tau \tag{10.26}$$

becomes

$$\sum_{i=1}^n v(i\delta t,t,\Omega_i)v_t(i\delta t,t,\Omega_i)\delta t, \tag{10.27}$$

where $\delta t = t/n$. Therefore, we can differentiate with respect to t and using $v(t,T,\Omega_t) = 0$, we have:

$$dr(t) = F_t(0,t)dt + \left(\int_0^t \left[v(\tau,t,\Omega_\tau) v_{tt}(\tau,t,\Omega_\tau) + v_t(\tau,t,\Omega_\tau)^2 \right] d\tau \right) dt \quad (10.28)$$
$$+ \left(\int_0^t v_{tt}(\tau,t,\Omega_\tau) dz(\tau) \right) dt + [v_t(\tau,t,\Omega_\tau)|_{\tau=t}] dz(t)$$

It is interesting to examine the terms on the right-hand side of this equation. The first and forth terms are straightforward. The first term shows that one component of the drift in r is the slope of the initial forward rate curve. The fourth term shows that the instantaneous standard deviation of r is $v_t(t,T,\Omega_\tau)|_{\tau=t}$. The second and third terms are more complicated, particularly when v is stochastic. The second term depends on the history of v because it involves $v(t,T,\Omega_\tau)$ when $\tau < t$. The third term depends on the history of both v and dz.

The second and third terms are liable to cause the process for r to be non-Markov. The drift of r between time t and $t + \delta t$ is liable to depend not only on the value of r at time t, but also on the history of r prior to time t. This means that, when we attempt to construct a tree for r, it is nonrecombining. An up movement followed by a down movement does not lead to the same node as a down movement followed by an up movement.

This highlights the key problem in implementing a general HJM model. We have to use Monte Carlo simulation. Trees create difficulties. When we construct a tree representing term structure movements, it is usually nonrecombining. Assuming the model has one factor and the tree is binomial as in Fig. 10.6, there are 2^n nodes after n time steps. If the model has two factors, the tree must be constructed in three dimensions and there are then 4^n nodes after n time steps. For $n = 30$, the number of terminal nodes in a one-factor model is therefore about 10^9; in a two-factor model it is about 10^{18}.

Now we can introduce the HJM tree into our system, consider the one-factor continuous-time model, for simplicity, we may have the first and fourth terms straightforwardly as shown in Fig. 10.6. Let us follow the previous format shown in Fig. 10.2 to establish a multi-grouped tree frame for the nonrecombining tree as a matrix, and say the size is $w \times q$. Each element will have a sub-tree and the total tree will be $n = w \times q$. Each sub-tree has δ branches with l levels, as shown in Fig. 10.5:

$$\text{multi - tree} = \begin{bmatrix} T_{w_{1,1}} & T_{w_{1,2}} & \cdots & T_{w_1,q_1} \\ T_{w_2,1} & T_{w_2,2} & \cdots & T_{w_2,q_2} \\ \cdots & \cdots & \cdots & \cdots \\ T_{w_i,1} & \cdots & \cdots & T_{w_i,q_i} \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Fig. 10.5 Sub-tree frame

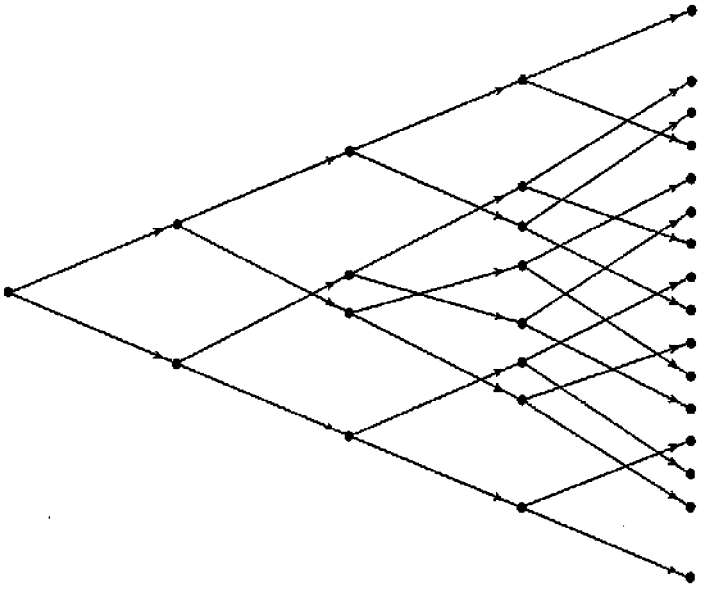


Fig. 10.6 A nonrecombining tree such as that arising from the general HJM model construction

where each sub-tree, say T_{ij}, will be constructed as shown in Fig. 10.6.

Therefore we may have $n = \delta_1^{l1}.\ \delta_2^{l2} \ldots$, which makes the k_0 that we were talking about above divide by factors. If we take each sub-tree (Fig. 10.6) as the same for our simplest case and total y groups, the final sub-tree k_0 would be $k_0' = k_0/y$. It is noted that the level and nodes are relations of 2^n nodes after n time steps. Also if the model has two factors, the tree must be constructed in three dimensions and there are then 4^n nodes after n time steps. As an example, let $n = 30$. The number of terminal nodes in a one-factor model is about 10^9 and in a two-factor model it is about 10^{18}. In this case we have a very small probability of an attacker succeeding. For example, if we take the results from [8], $k_0 = 1$ with 1000 tags, the probability of an attacker succeeding is 0.1 percent. Now if we just divide by 500 sub-tree, under the above conditions, the final probability of an attacker succeeding is 0.02 percent, which is a reasonable level.

10.4 Conclusion

We have focused on the very popular tree-based technique to keep the lower cost passive RFID systems staying at the lower price, first presenting the "sub-tree" construction frame. A careful analysis shows that our RFID systems can be acceptable (the probability of an attacker succeeding is 0.01%) for a real system. Figures 10.3 and 10.4 show that the passive RFID system has a trade-off between identifying

one tag in the whole system (reaction time), t_{CR} and the tree branch factor δ, and related security level. We suggest that, based on our analyses, a deployment of passive RFID systems could take after the turning point to keep the whole system at the reasonable security level and at the same time to have a reasonable reaction time. We further introduced a nonrecombining tree arising from the general HJM model and extended it to design a multi-grouped tree system to show that it would significantly decrease the overall probability of an attacker succeeding by increasing the branch factors and decreasing the number of tags tampered with by the attacker.

References

1. Huber GP (1991) Organizational learning: contributing processes and the literatures. Org Sci 2(1):88–115
2. Standford V (2003) Pervasive computing goes the last hundred feet with RFID systems, IEEE Perv Comp 2:2
3. Angeles R (2005) RFID technologies: supply-chain applications and implementation issues, Info Sys Mgmt 22:1 pp 51–65
4. Thompson CA (2004) Radio frequency tags for identifying legitimate drug products discussed by tech industry, Amer J Health-Sys Pharm 61(14): 1430–1431
5. Yang G, Jarvenpaa SI (2005) Trust and radio frequency identification (RFID) adoption within an alliance, In: Sprague R, ed Proc 38th Hawaii Intl Conf Sys Sci, Big Island, HI, January 2005, pp 855–864, IEEE Comp Soc Press, Los Alamitos, CA, USA
6. Bono S, Green M, Stubblefield A, Juels A, Rubin A, Szydlo M (2005) Security analysis of a cryptographically-enabled RFID device, In: 14th USENIX security symposium, pp 1–16, Baltimore, Maryland, USA
7. Electronic Product Code Global Inc *http://www.epcglobalinc.org*
8. Avoine G (n.d.) Security and privacy in RFID systems. Online bibliography available at *http://lasecwww.epfl.ch/∼gavoine/rfid/*
9. Molnar D, Wagner D (2004) Privacy and security in library RFID: issues, practices, and architectures. In: Pfitzmann B, Liu P, ed, Conference on computer and communications security - CCS'04, pp 210–219, ACM Press, Washington, DC, USA
10. Golle P, Jakobsson M, Juels A, Syverson P (2004) Universal reencryption for mixnets. In: Okamoto T, ed, The Cryptographers track at the RSA conference, CT-RSA, Lecture notes in computer science, 2964:163–178, San Francisco, California, USA, Springer-Verlag
11. Henrico D, Muller P (2004) Tackling security and privacy issues in radio frequency identification devices. In: Ferscha A, Mattern F, ed, Pervasive computing, Lecture Notes in Computer Science 3001:219–224, Vienna Austria, Springer-Verlag
12. Ohkubo M, Suzuki K, Kinoshita S (2003) Cryptographic approch to "privacy-friendly" tags. In: RFID privacy workshop, MIT, Massachusetts, USA
13. Avoine G, Oechslin P (2005) A scalable and provably secure hash based RFID protocol. In: International workshop on pervasive computing and communication security, PerSec, pp 110–114, Kauai Island, Hawaii, USA, IEEE Computer Society Press
14. Huang X, Sharma D (2006) Investigating security in multi-tree based technique. In:. Gabrys B, Howlett RJ, Jain LC (eds) RFID systems. KES 2006 Part III. LNAI 4253:1151–1156. Springer-Verlag, Berlin, Heidelberg
15. Hull J (2000) Options, futures, and other derivatives, Fifth Ed, Prentice-Hall, pp 574–577
16. Heath D, Jarrow R, Morton A (1992) Bond pricing and the term structure of interest rates: a new methodology, Econometrica 60(1):77–105

Chapter 11
Reliability, Capacity, and Energy Efficiency: A Comprehensively Optimized MAC Protocol for Industrial Wireless Sensor Networks[1]

Zhihao Xing, Peng Zeng, and Hong Wang

11.1 Introduction

Wireless Sensor Networks (WSN) are a special kind of network composed of large numbers of nodes which communicate and collaborate to accomplish the common task. They have applications in a variety of fields such as environmental monitoring, battlefield surveillance, target tracking, etc. [7]. Recently, the application of WSN in industrial monitoring has drawn the attention of many researchers, and related standards such as SP100.1x and wireless HART are under design. Industrial monitoring applications provide an opportunity for WSN to show its advantage in flexibility and easy maintenance, but they also bring up a host of new challenges, among which how to achieve reliable communication in the harsh industrial radio environment with efficient usage of energy and bandwidth is the foremost barrier to conquer. Reference [11] identifies four general causes which lead to reduced link quality in the industrial environment:

- *Static Multipath*: There usually exist many pipelines and other equipment in the industrial environment, which leads to more severe multipath problems than in general environment.
- *Time Variant Multipath*: In an environment with moving people and objects, radio quality may change with time in seconds or minutes.
- *Static Interference*: Interference in the industrial environment, such as RFID interrogators, electromagnetic noise produced by running equipment, and so on, may block the radio within a specific frequency scope for a period of time.

Zhihao Xing
Shenyang Institute of Automation, Chinese Academy of Science, Shenyang 110016, China
e-mail: xingzhihao@sia.cn

Zhihao Xing, Peng Zeng, and Hong Wang
Graduate School of the Chinese Academy of Science, Beijing 100039, China

[1] This work was supported by the National Natural Science Foundation of China (Grant No. 60374072 and 60434030).

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

- *Time Variant Interference*: The ISM frequency band devoted to general-purpose wireless communication devices is crowded with traffic from WLAN, cordless telephones, Bluetooth, and innumerable other devices. In order to satisfy the communication requirements of industrial applications at different levels, these devices may coexist in a factory environment, and there is no way to predict what interferers will be present at a given time, location, and frequency scope.

For the single-channel MAC protocols, to provide reliable communications in the radio environment with the above mentioned problems in an energy and bandwidth efficient way is a very difficult task, if not impossible. Fortunately, recent developments in radio technology, as seen in Berkeley's Mica2 Mote, Rockwell's WINS nodes, and the radio chips which comply with the IEEE 802.15.4 physical layer standard all have a multi-channel capability, which can be used to solve the communication reliability problem in an efficient way.

11.2 Related Works

Many multi-channel MAC protocols have been published in the research community for wireless ad-hoc networks. These protocols are mainly focused on how to utilize multi-channel to achieve parallel transmissions and improve throughput. According to their operation mechanism, these protocols can be classified into four categories as follows:

1. *Dedicated Control Channel*: In this category of protocols, nodes are equipped with two radios, one fixed on a dedicated channel for control messages exchange, and another used for data transmission. For each data transmission, the RTS/CTS must be exchanged on the dedicated control channel first to negotiate the data transmission channel. DCA [13], DCA-PC [14], and DPC [16] are protocols of this category.
2. *Two-Phase Channel Access*: This category of protocols requires network synchronization. Time is divided into an alternating sequence of control and data exchange phases. In the control phase, all nodes contend for the common control channel by RTS/CTS exchange and attempt to make agreements for channels to be used in the data exchange phase. In the data exchange phase, only the nodes which made agreements successfully are allowed to transmit data. MMAC [10] and MAP [9] are examples of this category.
3. *Channel Hopping with Common Pattern*: This category of protocols requires network synchronization. Time is slotted and slot boundaries are aligned. Nodes cycle through all channels synchronously. A pair of nodes stop hopping as soon as they make an agreement (through RTS/CTS exchange) for transmission and stay at the current channel to complete the data exchange, while other nodes continue the hopping. When the data exchange completes, the pair of nodes rejoin the common hopping pattern. CHMA [1] and CHAT [2] belong to this category, and CHAT improves CHMA by introducing the "packet trains" mechanism.

4. *Channel Hopping with Random Pattern*: Examples of this approach include SSCH [12] and McMAC [5]. In this category of protocols, time is slotted and network synchronization is required to align the slot boundary (the two protocols also allow an unaligned slot boundary, at the cost of performance degradation). Nodes select their hopping patterns independently by using their random number generators, and then perform the channel hopping according to this pattern. At each slot, neighboring nodes may operate on different channels, so when a node wants to transmit data to a neighboring node, it must jump to the operation channel of that node to perform the transmission.

In the above mentioned protocols, the first three categories have the "control channel bottleneck" problem. For each packet transmission, there must be a successful RTS/CTS exchange between the source node and the destination on a control channel in order to select the data transmission channels. When the network load is high and the number of data channels is large, the control channel may become a bottleneck and prevent data channels from being fully utilized. Furthermore, in protocols of categories 1 and 2, the control channel is predetermined and fixed, so if the control channel is heavily interfered with, the network performance will degrade seriously. Protocols of category 4 avoid the problems of the first three categories, but new problems occur: (1) The "absent destination" problem becomes a new performance killer which incurs frequent invalid transmissions and results in a great waste of energy and bandwidth. An extreme scenario of the "absent destination" problem is shown in Fig. 11.1. The operation channels of nodes A, B, C, D, and the gateway at the current slot are CH_1, CH_2, CH_3, CH_4, and CH_5, respectively, and $CH_i \neq CH_{i+1},\ 1 \leq i \leq 4$. If every node (except the gateway) wants to send a packet to its upstream node at the current slot, then the following scenario occurs: A jumps to B's operation channel (CH_2) to make the transmission to B, but B jumps to C's operation channel(CH_3) to make the transmission to C at the same time. For A, its destination B becomes an "absent destination", and its transmission becomes an invalid transmission. For the same reason, the transmissions from B to C, and C to D also fail. At last, only the transmission from D to the gateway succeeds. (2) The mechanism by which every node picks a seed to generate a pseudo-random hopping pattern leaves space for further throughput improvement. (3) In both SSCH and McMAC, there is no discussion about how to provide broadcast service.

One common problem in all the protocols mentioned above is the RTS/CTS dependence problem. Protocols of the first three categories depend on RTS/CTS to negotiate the data transmission channel before each data exchange, and protocols of category 4 depend on RTS/CTS to detect "absent destination." The RTS/CTS

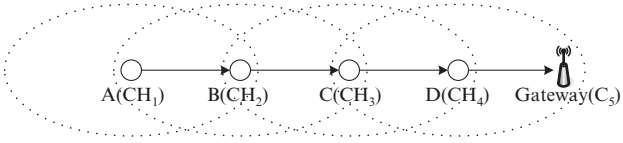


Fig. 11.1 An extreme scenario of "absent destination"

mechanism is too expensive for typical WSN applications, because the RTS/CTS frames are usually comparable to their data frames in length.

The Channel Hopping with Negotiated Pattern (CHNP) protocol proposed in this paper is a single radio multi-channel MAC protocol specially designed for industrial WSN applications, which pays special attention to the reliability performance, and optimizes throughput, energy, and delay performance at the same time. In CHNP, a novel MAC framework based on synchronized channel hopping mechanism is proposed, which not only avoids the drawbacks in the above mentioned protocols, but also provides a platform on which the performance optimization of reliability, throughput, energy efficiency, and delay can be performed effectively. On the basis of this framework: (1) an effective hopping pattern allocation algorithm is provided to achieve better collision domain division and promote the parallel transmissions to a higher degree; (2) a channel access control mechanism is designed to solve the "absent destination" problem and achieve higher energy efficiency; (3) an adaptive transmission mechanism is designed to sidestep the low quality channels and provide reliable communication even in the harsh RF environment.

The rest of this paper is organized as follows: In Sect. 11.3, we make a detailed description of the CHNP protocol. Then in Sect. 11.4, comprehensive simulations are conducted to evaluate its performance. Finally, in Sect. 11.5, we give the conclusions and point out future work.

11.3 The CHNP Protocol

11.3.1 Preliminaries

In CHNP, time is slotted and the slot boundary of all nodes is aligned on the basis of network synchronization. A higher synchronization precision can shorten the guard time for synchronization error, and thus improve the performance of CHNP, but it requires more frequent synchronization packets exchange. Precision in 1–2 milliseconds is a reasonable tradeoff, which can be realized through synchronization packets exchanging in a tens of minutes period using state-of-the-art technology.

For a better description of CHNP, the following terms must be illustrated first:

- The k-neighborhood set of node v, $\mathrm{NS_k(v)}$: The set of nodes whose shortest path to node v has length at most k. For convenience, we also refer to nodes in $\mathrm{NS_k(v)}$ as v's k-hop neighbors.
- The contention set of node v, CS(v): The set of nodes which can interfere with the transmission from node v to its arbitrary 1-hop neighbor. Under the assumption that the interference radius equals the communication radius,[2] we get:

$$CS(v) = \bigcup_{w \in NS_1(v)} NS_1(w) = NS_2(v) \tag{11.1}$$

[2] This assumption is widely used in many MAC protocols, such as [6], [15].

- The collision domain of node v, CD(v): We refer to the set $\{\{v\} \cup CS(v)\}$ as the contention domain of node v.

11.3.2 The Synchronized Channel Hopping Framework

In the CHNP network, nodes perform synchronized channel hopping according to their selected hopping patterns, which define their operation channels at every slot. The slot is the basic time unit of channel hopping, and the hopping operation can only happen at the slot's boundary. The hopping pattern of a node is determined by the following parameters:

- hopping sequence(HS)
- broadcast interval (BI)
- start channel of broadcast slot(SC-BS)
- start channel of unicast slot(SC-US).

HS is a permutation of all available channels. There are two types of slots in the hopping pattern: unicast slots and broadcast slots. The former are used for unicast transmission, and the latter are used to support broadcast service. There is one broadcast slot inserted in the hopping pattern for every specified number of continuous unicast slots. BI defines the interval (the number of unicast slots) of the two closest broadcast slots in the hopping pattern. A node's operation channel at every unicast slot and broadcast slot is determined by the tuple <HS, SC-US> and the tuple <HS, SC-BS>, respectively.

In CHNP, all nodes in the network have the same HS, BI, and SC-BS parameters which are configured at the gateway and flooded to the whole network at the network initialization stage, but the SC-US parameter of every node is allocated through a distributed algorithm. Thus, two conclusions can be obtained immediately: (1) two different hopping patterns can differ only in their SC-US parameters; (2) at the broadcast slots, all nodes operate on the same channel.

Figure 11.2 shows an example to illustrate the concept of hopping patterns. A, B, and C are nodes in a CHNP network in which every node is equipped with radio chips that can provide 16 available channels. The HS of the network is (0, 4, 8, 12, 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15), the BI and SC-BS parameters are configured

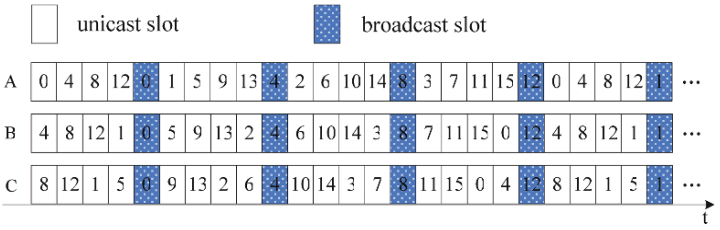


Fig. 11.2 A hopping pattern example

to 4 and 0, respectively, and the SC-US allocated to A, B, and C is 0, 4, and 8, respectively. The final hopping pattern of A, B, and C is shown in Fig. 11.2.

Nodes with different SC-US parameters operate on different channels at all the unicast slots. If a node has packets for some 1-hop neighbor operating on a different channel, it must first switch its radio to the operation channel of this neighbor, then start the contention and transmission procedure. With a common hopping start time, a node can obtain its 1-hop neighbor's operation channel at any slot if it has the hopping pattern knowledge of the neighbor; or more concretely, the SC-US parameter of the neighbor. The collection of 1-hop neighbors' SC-US parameters is carried out in the distributed hopping pattern allocation algorithm described in Sect. 3.3.

In the following text, the operation channel of a node at the current slot is called the node's home channel, and the operation channel of a node's destination is called the node's destination channel.

11.3.3 The Hopping Pattern Allocation Algorithm

The hopping pattern allocation algorithm is used to allocate a hopping pattern, or more concretely, a SC-US parameter to each node in the network. The algorithm is executed at the network initialization stage when all nodes are on the same channel. Because parallel transmissions can be realized on different channels, the hopping pattern allocation algorithm is designed to guarantee that all the channels can be fully and evenly utilized by nodes within a collision domain; or more concretely, for an arbitrary node v, if the number of channels in HS is no smaller than the number of nodes in CD(v), then each node in CD(v) should select a different channel as its SC-US; otherwise, all channels in HS should be evenly selected by nodes in CD(v) as their SC-US. The key of the algorithm is to guarantee that for each node v in the network, the nodes in CD(v) can select their SC-US parameter serially. The algorithm requires that every node in the network has a unique ID, and in the algorithm, we use the $\mathrm{NS}_2(\mathrm{v})$ to approximate CS(v). It includes a two-stage operation:

1. Every node in the network collects its 2-hop neighbors' IDs. A simple way to implement this is for each node to broadcast twice. In the first broadcast period, each node broadcasts its own ID, and when this period finishes, every node knows all its 1-hop neighbors' IDs. In the second broadcast period, each node broadcasts all its 1-hop neighbors' IDs. Hence, when the second broadcast period finishes, each node obtains all its 2-hop neighbors' IDs.
2. Every node arranges all its 2-hop neighbors' IDs and its own ID into an ascending order list, and decides its SC-US parameters selection time according to its own ID's position in the list. More concretely, if node v has the smallest ID in the list, it can immediately select a random channel in HS as its SC-US parameter, then broadcast its selection to inform the nodes in $\mathrm{NS}_2(\mathrm{v})$; otherwise, if node v's ID is not the smallest one, it suspends its selection until the nodes with smaller IDs in the list have finished the selection (i.e., until it receives the informing packets from those nodes). When node v is performing the selection, if there are channels

that have not been used yet, it can randomly pick one from the unused channels, or else it can select the least reused channel as its SC-US parameter.

Local broadcast operation frequently uses this algorithm, and methods provided in [4, 8] can be adopted to improve the broadcast reliability. When the algorithm finishes, for each node v in the network, the set CD(v) is evenly partitioned into MIN{|CD(v)|, N_c} (N_c denotes the number of channels in HS) subsets which operate on different channels at all unicast slots. Parallel transmissions can occur if their destinations belong to different subsets.

11.3.4 Channel Access Control Mechanism

The slot is the basic time unit of the channel hopping operation. A unicast / broadcast slot is comprised of a fixed count of unicast / broadcast microslots. The length of a unicast/broadcast slot should be long enough to unicast / broadcast the maximum-sized packet allowed by CHNP. The channel access control mechanism in the unicast microslot is different from that in the broadcast microslot.

11.3.4.1 Channel Access Control in the Unicast Microslot

The constitution of the unicast microslot is shown in Fig. 11.3. The wakeup tone (WT) is used to save energy, and the leaving tone (LT) is used to avoid invalid transmissions caused by "absent destination." Both the WT and the LT are bit sequences of specified length. The transmission of them is just to let the sender's 1-hop neighbors which operate on the transmission channel detect that the channel is busy, and they are not required to be correctly received. CW is the contention window size which denotes how many backoff slices the contention window includes. $T_M D$ is the maximum time needed from the beginning of a microslot to the time that the some node with packets to send has finished the transmission of the preamble.

For a better description of the channel access control mechanism, we classify all the nodes into two types: if a node has packets to send at the current microslot, it is called the type S node, otherwise it is called the type R node. Table 11.1 shows the symbols needed in the description of the channel access control mechanism.

For a type R node, when the microslot starts, it listens on its home channel for at most T_{wt} time. If it detects that the channel becomes busy (some 1-hop neighbor has transmitted the wakeup tone), it then prepares itself to receive the coming

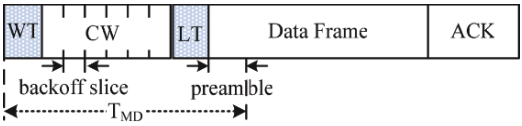


Fig. 11.3 The constitution of the unicast microslot

Table 11.1 Symbols definition

Symbol	Description
T_{sw}	the minimum time required by the channel switching operation
T_{cca}	the minimum time required by the clear channel access (CCA) operation
Δ	the maximum synchronization error between 1-hop neighbors
T_{bs}	the length of a backoff time slice
T_{wt}	the length of the wakeup tone
T_{lt}	The length of the leaving tone

packet; otherwise it just turns its radio into sleep state for the rest of the current microslot to save energy.

A type S node's operation is rather complex. At the beginning of the current microslot, it starts transmitting the wakeup tone on its destination channel, which makes its destination node ready for the reception. To guarantee that the wakeup tone can be detected by the destination node, the following inequation must be satisfied: $T_{wt} \geq 2\Delta + T_{cca}$.

After the wakeup tone is transmitted, the node selects a random backoff value from [0, CW], and starts backoff. During the backoff procedure, the node listens on its home channel and destination channel alternatively (the dwell time on each channel is a backoff slice), and performs the respective operations according to the listening results. A backoff time slice should be long enough to complete the channel switching operation and the CCA operation, i.e., the following inequation should be satisfied: $T_{bs} \geq T_{sw} + T_{cca}$.

There are three possible listening results, as shown in Fig. 11.4:

1. The two channels are both idle for the whole backoff period, as shown in Fig. 11.4a. In this case, when the backoff finishes, the node first transmits the leaving tone on its home channel, which makes its potential senders (the nodes that want to transmit packets to it) aware of its leaving and abandon the transmission attempt, then switches to the destination channel and makes the transmits. To guarantee that the leaving tone can be detected by the potential senders, the following inequation should be satisfied: $T_{lt} \geq 2T_{bs} + T_{cca}$.
2. The destination channel is detected busy in the backoff period, as shown in Fig. 11.4b. In this case, the channel becomes busy either because of the transmission of packets or because of the transmission of the leaving tone. If the reason is the former, in order to avoid interfering with the ongoing transmission, the node has to abandon its transmission attempt at the current microslot; if the reason is the latter, in order to avoid invalid transmissions, the node also has to abandon its transmission attempt. Thus, when the node detects that the destination channel becomes busy, it immediately switches to its home channel and stays on this channel listening. If a valid preamble is received before T_{MD}, then it starts receiving the data frame, otherwise it starts sleeping until the end of the current microslot.

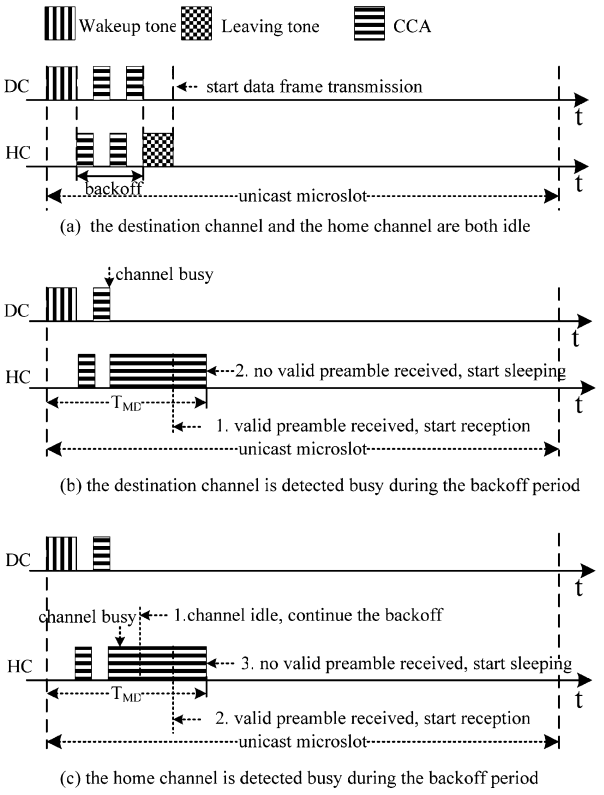


Fig. 11.4 The operation of type S node in the unicast microslot

3. The home channel is detected busy in the backoff period, as shown in Fig. 11.4c. If the channel busy duration is no longer than T_{lt}, then the node can infer that some type of S node which operates on the same home channel has just transmitted the leaving tone. In this case, it can continue the backoff procedure, just as if nothing has happened. If the channel busy duration is longer than T_{lv}, then the node can infer that some type of S node is transmitting the packet and itself may be the destination, so it just receives the packet.

11.3.4.2 Channel Access Control in the Broadcast Microslot

In the broadcast microslot, there is no "absent destination" problem, so the leaving tone is not needed in the constitution of the broadcast microslot, as shown in Fig. 11.5.

The wakeup tone is used in a similar way as in the unicast microslot, the only difference being that each node just stays on its home channel to perform the related operations now. In order to fully utilize the bandwidth, when there are no broadcast

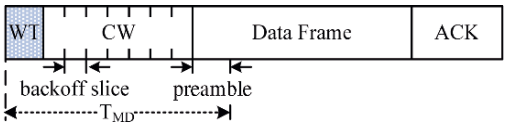


Fig. 11.5 The constitution of the broadcast microslot

packets the broadcast microslot can also be used to transmit unicast packets. Assume nature numbers CW_B and CW_U satisfy $CW_B < CW_U$. When a node wants to transmit a broadcast packet, it selects a random backoff value in $[0,\ CW_B]$, and when it wants to transmit a unicast packet, it selects a random backoff value in $[CW_B,\ CW_U]$. In this way, broadcast packets are given higher transmission priority than unicast packets in the broadcast microslot.

11.3.5 Adaptive Transmission Mechanism

11.3.5.1 Channel Quality Estimation and Channel Classification

At the runtime of the network, nodes perform the real-time channel quality estimation for each channel between themselves and all their 1-hop neighbors. An ideal estimator should react quickly to large changes of channel quality, yet be stable, have a small memory footprint, and be simple to compute. The WMEWMA estimator proposed in [3] can be used here to perform the channel quality estimation. According to the estimation result, channels are classified into two categories: good channels and bad channels. Each node v maintains a channel classification matrix (CCM) to record the category of each channel between itself and its 1-hop neighbors.

$$CCM(v) = \begin{pmatrix} c_{00} & c_{01} & \cdots & c_{0n} \\ c_{10} & c_{11} & \cdots & c_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ c_{m0} & c_{m1} & \cdots & c_{mn} \end{pmatrix} \tag{11.2}$$

where m is the number of nodes in $NS_1(v)$, n is the total number of channels, and $c_{ij} = \{GOOD|BAD\},\ 0 \le i \le m,\ 0 \le j \le n$. Thus the elements in row 1 of CCM(v) record the category of all channels between v and its number 1 neighbor.

11.3.5.2 The Adaptive Transmission Algorithm

The adaptive transmission algorithm is shown in Fig. 11.6. At every node, the packets waiting to be transmitted are queued in *sBuffer* in FIFO order. At each microslot, the sender searches *sBuffer* to find the first packet that can be transmitted on a good channel, i.e., an operation channel of the packet's destination classified as "GOOD" in the sender's CCM. If it can not find such a packet, it sleeps at this microslot,

```
Adaptive Transmission Algorithm
1  for each microslot in current slot
2     found_flag = FALSE
3     for each pkt in sBuffer /* in FIFO order*/
4        if (ccm[pkt.dest][op_channel] == BAD)
5           next pkt
6        else
7           found_flag = TRUE
8           break
9        end if
10    next pkt
11    if found_flag = true
12       remove pkt from sBuffer
13       transmit pkt
14       if the tx node does not receive the ACK before timeout
15          pkt.retrans_count ++
16          if  pkt.retrans_count <= MAX_RETRANS
17             insert pkt into tfBuffer
18          end if
19       end if
20    end if
21    wait until current microslot ends
22 next microslot
23 remove all the packets in tfBuffer, and reinsert them into sBuffer
```

Fig. 11.6 The adaptive transmission algorithm carried out at each slot

otherwise it transmits the found packet. For each transmitted packet, the sender expects an ACK from the destination. If no ACK is received, the sender increases the retransmission counter of the failed packet and inserts it into *tfBuffer*, which is used to hold the failed packets at the current slot. When the current slot ends, the packets in *tfBuffer* will be reinserted into *sBuffer* in FIFO order, and they will be retransmitted at later slots. If the retransmission counter of a packet exceeds the predefined threshold, the packet will be discarded.

The effect of the adaptive transmission algorithm is that when there are multiple packets waiting to be transmitted/retransmitted, the packets that can be transmitted on good channels are given higher transmission priority, and the packets with little chance to be transmitted successfully at the current slot are postponed to the slots when they also could be transmitted on good channels. Thus the adaptive transmission algorithm can effectively improve the packets delivery ratio. Moreover, it can improve the energy efficiency of the network, because it reduces the average retransmission times.

11.4 Performance Evaluations

In order to evaluate the CHNP protocol, a comprehensive performance comparison of CHNP, McMAC, and the non-superframe mode of IEEE 802.15.4 is conducted by using the OPNET simulation tools. The detailed simulation configuration is shown

in Table 11.2. The physical layer parameters listed in the table are from the CC2420 radio chip, which complies with the IEEE 802.15.4 physical specification in the 2.4GHz band.

In the simulation, we compare the throughput, delivery ratio, MAC layer delay, energy efficiency, and the anti-interference performance of the three protocols. For CHNP and McMAC, we evaluate their performance with 4, 8, and 16 channels, respectively. When evaluating the anti-interference capability of CHNP, we deploy four IEEE 802.11 nodes, which also run in the 2.4GHz band (a single IEEE 802.11 channel occupies about three channels of CC2420) as the interference source.

Figure 11.7 compares the throughput performance of the three protocols. We can observe that CHNP and McMAC can achieve much higher throughput than the single-channel IEEE802.15.4; and with the load increasing, the throughput performance of CHNP outperforms McMAC. When the number of available channels is 16 and the load on each node is 100 packets per second, CHNP almost doubles the throughput of McMAC. The performance difference between CHNP and McMAC

Table 11.2 Simulation configuration

Simulation Parameter	Configuration
Terrain	$(100\text{m} \times 100\text{m})$ square
nodes number	100
Placement	random uniform
traffic model	random destination
payload size	50 byte
packet interarrival time	exponential distribution
maximum 1-hop synchronization error	1ms
data rate	250kbps
channels number	4|8|16
communication/interference radius	25m

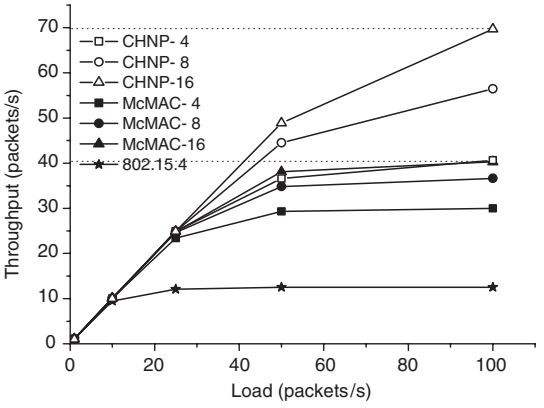


Fig. 11.7 Average throughput vs. load on each node

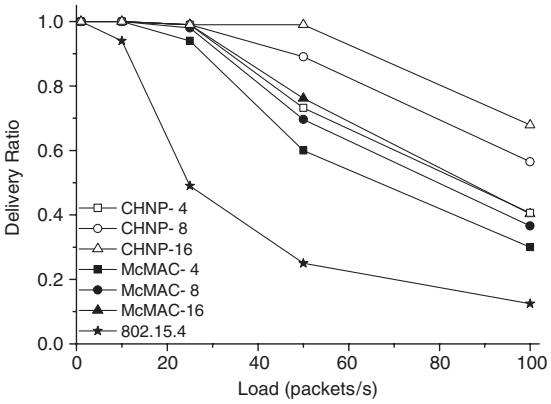


Fig. 11.8 Average delivery ratio vs. load on each node

comes from three facts: first, for each packet transmission, McMAC has the extra RTS/CTS overheads; second, the "absent destination" problem in McMAC becomes more severe with the load increasing, and bandwidth is wasted on invalid transmissions, whereas CHNP can effectively avoid the invalid transmissions incurred by the "absent destination" problem; third, through the hopping pattern allocation algorithm, CHNP can achieve a higher degree of parallel transmissions.

Figure 11.8 compares the delivery ratio performance of the three protocols. The delivery ratios of all three drop with the load increasing, but the main reasons leading to the drop are different. For CHNP, the drop is caused mainly by the buffer overflow; for McMAC, besides the buffer overflow, the "absent destination" problem and transmission collision are also main causes; and for 802.15.4, the drop is caused mainly by the buffer overflow and transmission collision. The difference is that the drop caused by the buffer overflow has no energy and bandwidth overheads, whereas the drop caused by the "absent destination" problem and transmission collision involves the overheads incurred by the invalid transmissions.

Figure 11.9 shows the MAC layer delay of the three protocols. The MAC layer delay is the duration from the time when a packet arrives at the MAC layer of the source node to the moment when it is successfully received by the destination. We can observe that with the load increasing, the advantage of CHNP over McMAC and 802.15.4 becomes more and more obvious. The reason is that the contention, collision, and "absent destination" problem (for McMAC only) become more severe at higher loads, which means that there are more retransmissions and longer delays in 802.15.4 and McMAC than in CHNP.

Because McMAC and the non-superframe mode of IEEE 802.15.4 have no energy-saving mechanism, we only evaluate the energy efficiency of CHNP by obtaining its duty cycle under a different load, as shown in Fig. 11.10. The simulation result shows that at the low and moderate loads, CHNP can achieve a fairly low duty cycle.

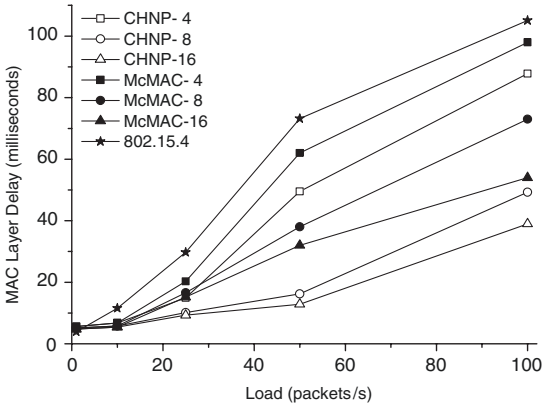


Fig. 11.9 Average MAC layer delay vs. load on each node

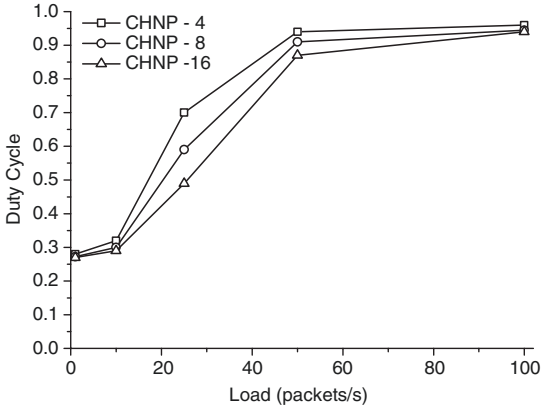


Fig. 11.10 Average duty cycle vs. load on each node

Figure 11.11 compares the anti-interference capability of the three protocols. In the simulation, we let the 802.11 nodes work in the saturation mode (i.e., they always have packets to send) with a communication/interference radius of 100 m. We also let the IEEE 802.15.4 MAC protocol work on a channel that can be interfered by the 802.11 nodes. For a better illustration of the three protocols' anti-interference capabilities, the delivery ratio curves for each protocol with and without interference are shown in Fig. 11.11. We can observe that the interference has almost no impact on the delivery ratio of CHNP at the low load. When the load on each node increases to 100 packets per second, the delivery ratio of CHNP with interference only drops about 10 percent compared to the scenario without interference. In contrast to CHNP, the delivery ratios of McMAC and IEEE 802.15.4 have great drops with interference. At the load of 100 packets per second, the delivery ratios of McMAC and IEEE 802.15.4 with interference drop about 38 and 46 percent, respectively. This result confirms the effectiveness of the adaptive transmission mechanism of CHNP.

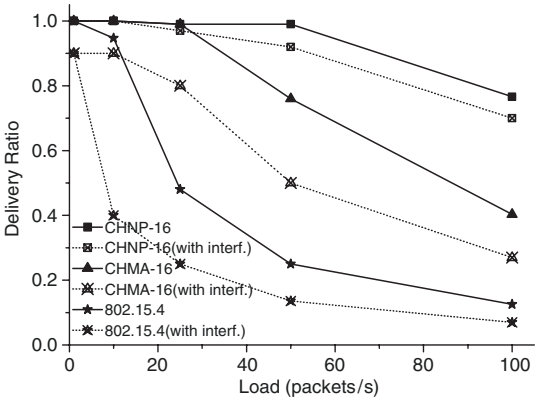


Fig. 11.11 Comparison of average delivery ratio with/without interference

11.5 Conclusion

We have presented CHNP, a multi-channel MAC protocol especially designed for industrial WSN applications, which takes full advantage of multiple channels and can achieve more reliable communication in the harsh industry radio environment in an energy and bandwidth efficient way. We have shown through comprehensive simulation that compared to previous protocols, CHNP achieves significant performance improvement in reliability, throughput, energy, and delay.

We are implementing the CHNP protocol on the CC2420 radio chips now. When this work is finished, we will deploy the network in a real industrial radio environment to evaluate its performance, and will make further optimizations and improvements in accordance with the evaluation results.

References

1. Tzamaloukas A, Garcia-Luna-Aceves JJ (2000) Channel-hopping multiple access. In: IEEE ICC 2000, New Orleans, Louisiana, pp 18–22
2. Tzamaloukas A, Garcia-Luna-Aceves JJ (2000) Channel-hopping multiple access with packet trains for ad hoc networks. In: Proceedings IEEE device multimedia communications (MoMuC'00), Tokyo, pp 23–26
3. Woo A, Culler D (2003) Evaluation of efficient link reliability estimation for low-power wireless sensor networks. Technical Report UCBCSD-03-1270, U.C. Berkeley Computer Science Division
4. Vollset E, Ezhilchelvan P (2003) A survey of reliable broadcast protocols for mobile ad-hoc networks. In: Technical Report CS-TR-792, University of Newcastle-upon-Tyne
5. So HW, Walrand J (2005) McMAC: A multi-channel MAC proposal for ad-hoc wireless networks. Technical Report. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
6. Bao L, Garcia-Luna-Aceves JJ (2001) A new approach to channel access scheduling for ad hoc networks. In: ACM MobiCom 2001, pp. 210–221

7. Tubaishat M, Madria S (2003) Sensor networks: an overview. J IEEE Potentials 22(2):20–23
8. Tourrilhes J (1998) Robust broadcast: improving the reliability of broadcast transmission on CSMA/CA. In: The Ninth IEEE international symposium on PIMRC. Boston, USA, pp 1111–1115
9. Chen J, Sheu S, Yang C (2003) A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs. In: Proceedings of PIMRC 2003, Vol. 3, pp. 2291–2296
10. So J, Vaidya N (2004) Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. ACM international symposium on mobile ad hoc networking and computing (MobiHoc), pp 222–233
11. Werb J, Sexton D (2005) Improved quality of service in IEEE 802.15.4. International workshop on wireless and industial automation, San Francisco, California, pp 1–6
12. Bahl P, Chandra R, Dunagan J (2004) SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In: Proceedings of Mobicom 2004, pp 216–230
13. Wu S-L, Lin C-Y, Tseng Y-C, Sheu J-P (2000) A new multi-channel MAC protocol with on demand channel assignment for mobile ad-hoc networks. In: Proceedings of the international symposium on parallel architectures, algorithms and networks (ISPAN '00), Dallas/Richardson, Texas, USA, pp 232–237
14. Wu S-L, Lin C-Y, Tseng Y-C, Lin C-Y, Sheu JP (2002) A multi-channel MAC protocol with power control for multi-hop mobile ad-hoc networks. The Computer Journal 45(1):101–110
15. Rajendran V, Obraczka K, Garcia-Luna-Aceves JJ (2003) Energy-efficient, collision-free medium access control for wireless sensor networks. In ACM SenSys 2003, pp 181–192
16. Hung W-C, Eddie Law KL, Leon-Garcia A (2002) A dynamic multi-channel MAC for ad-hoc LAN. In: Proceedings of the 21st biennial symposium on communications. Kingston, Canada, pp 31–35

Chapter 12
A Self-Administrative Sub-Role-Based Delegation Model for PMI Systems*

Yueqin Liu, Yanqin Zhu, and Xizhao Luo

12.1 Introduction

The role-based access control (RBAC) model [1] has emerged since the 1990s as a promising approach for managing and enforcing security in a huge and complex system. The essential idea of RBAC is that privileges are associated with roles, and users are assigned to appropriate roles, thereby acquiring the roles' permissions. It provides availability, integrity, and confidentiality services for information systems and is deeply studied and generally used.

Delegation is referred to when one active entity in a system delegates its authority to another entity to carry out some functions [2]. It is a promising approach to realizing more flexible and scalable authorization management for the distributed systems. In a flexible access control model, it is necessary to have delegation of access rights between subjects, especially in a large distributed system. In the existing RBAC models, authorization delegations between subjects are only carried out by authorization management users. However, in distributed systems, networks, and cooperative computing systems, the numbers of users, permissions, and resources usually are large, and the relations among them are very complicated. Thus central authorization delegation can not meet the authorization management needs of a distributed system.

Delegation is increasingly becoming a basic issue in distributed authorization. Currently, there're some models which extend the RBAC model to support role

Yueqin Liu, Yanqin Zhu, and Xizhao Luo
School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu, China, 215006
e-mail: {210413017,yqzhu,xzluo}@suda.edu.cn

Yueqin Liu
Taizhou Branch, People's Bank of China, Taizhou, China, 225300

*This paper was supported by the Jiangsu Natural Science Foundation of China under Grant BK2004039 and Jiangsu Graduate Innovation Project in 2005.

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

delegation. However, their support for partial delegation, permission level delegation, and restricted inheritance seems very limited. Additionally, most of these models are centrally administered, which allows only one central authorization unit to grant access to subjects. By introducing the idea of using RBAC itself to manage RBAC [3], we present a self-administrative sub-role-based delegation model ASRBDM. Moreover, current research on the role-based delegation model remains in the theoretical stage, and applications supporting authorization delegation are rare. In this paper we also attempt to implement an authorization delegation system which supports authorization and delegation simultaneously.

The rest of this paper is structured as follows. Section 2 briefly reviews research on the role-based delegation models and gives an overview of background technologies. In Sect. 3 we present a self-administrative sub-role-based delegation model ASRBDM. In Sect. 4, we discuss the implementation architecture of our model using attribute certificates. Finally, Sect. 5 concludes the paper.

12.2 Related Works

12.2.1 Role-Based Delegation Model

A number of models dealing with various aspects of delegation have been published [2–7]. Currently, the typical role-based delegation models include RBDM0, RDM2000, PBDM, and RBDM using characteristics of permissions and so on.

The RBDM0 model [2, 4] proposed by Barka and Sandhu is the first delegation model based on role. In RBDM, a user can delegate his/her role to another user. One limitation of RBDM0 is that it does not address the relationships among each component of a delegation, which is a critical notion in the delegation model [5].

Zhang et al. introduced RDM2000 (role delegation model 2000) for user-to-user delegation in role-based systems [5]. It is an extension of RBDM0. It supports regular role delegation in role hierarchy and multi-step delegation. The RBDM0 and RDM2000 models, in particular, are primarily based on roles. Their unit of delegation is a role and they don't support permission level delegation. This property may breach the least privilege principle, one of the main security principles which the RBAC models support.

PBDM [6] supports flexible role and permission level delegation, which provides great flexibility in authority management. In PBDM the role space may grow explosively and there may be a great deal of redundant user-to-role and permission-to-role assignments, which complicates authorization management and results in inconsistency of authorization assignment.

You-Ri et al. [7] proposed a flexible role-based delegation model using characteristics of permissions. They extended the sub-role hierarchies concept, divided a role into sub-roles according to the characteristics of the permissions assigned to the role, and considered delegation and inheritances simultaneously. This model supports temporary delegation situations, multi-step delegation, and

multi-option revocation and controlled inheritance. It also supports constraints such as separation of duty based on permission in user-to-user and role-to-role delegation.

12.2.2 Privilege Management Infrastructure

At present the implementation of authorization delegation is mainly focused on studies of privilege management infrastructure (PMI). PMI is based on the ITU-T recommendation of directory systems specifications [8]. It is the infrastructure for issuing, storing, and managing ACs (attribute certificates). An attribute certificate-based PMI is a mechanism that can be used to support enterprise authority structures. Authorization infrastructure generally builds up on the basis of PKI, authenticates the user's identity by means of identity certificates, and then authorizes the user deploying his/her attribute certificates. Authorization requests and validation processes are implemented by applying, issuing, and revoking attribute certificates, which contain authorization information. Adopting attribute certificates simplifies authorization and access control management effectively.

At present the attribute certificate infrastructure best known is that from the EC PERMIS project [9, 10], which is an implementation of an X.509 PMI, and uses the role-based access control (RBAC) paradigm. Roles are stored in X.509 attribute certificates (ACs), and since these are digitally signed for integrity protection, it can support the distributed management of roles between multiple AAs. The PERMIS infrastructure comprises a privilege allocation (PA) component, a privilege verification (PV) component, a policy decision point (PDP), and a policy management GUI. The PERMIS PA component is responsible for allocating privileges to the users in the shape of roles stored in X.509 attribute certificates (ACs). The PERMIS PV component is responsible for authenticating and authorizing the users [10]. The role ACs are verified by the PV component, and all valid roles/attributes are passed to the PDP. The PDP then makes its access control decision on the user's request based on the target access policy and the valid attributes. All access control decisions are driven by an authorization policy, which is stored in a policy AC. Authorization policies are written in XML according to a DTD that has been published at XML.org. A full description of the PERMIS X.500 PMI RBAC policy can be found in [11].

The ITU-T proposal establishes four PMI models: general, control, delegation, and roles. The PERMIS project focuses on the roles model; the senior roles inherit all the permissions of the junior roles, and all the permissions are accumulated for the top senior role. Consequently, PERMIS does not meet the least privilege principle, one of the main security principles of access control models.

12.3 ASRBDM Model

In this paper, by introducing the idea of using RBAC itself to manage RBAC, we propose a new self-administrative delegation model ASRBDM on the basis of SRBDM presented in [7]. All the advantages of SRBDM are available in our proposed model. To further provide administrative convenience, especially in a decentralized administrative authority, we introduce the concepts of administrative role and administrative role hierarchy, and we decentralize permissions that often are highly centralized within a small team of security administrators.

12.3.1 Overview of Proposed Model

Figure 12.1 shows a summary of the model proposed in this paper. We divide a role into an administrative role and a regular role, permission into administrative permission and regular permission. Administrative permissions that were conventionally centralized to the system administrators are assigned to administrative roles. To

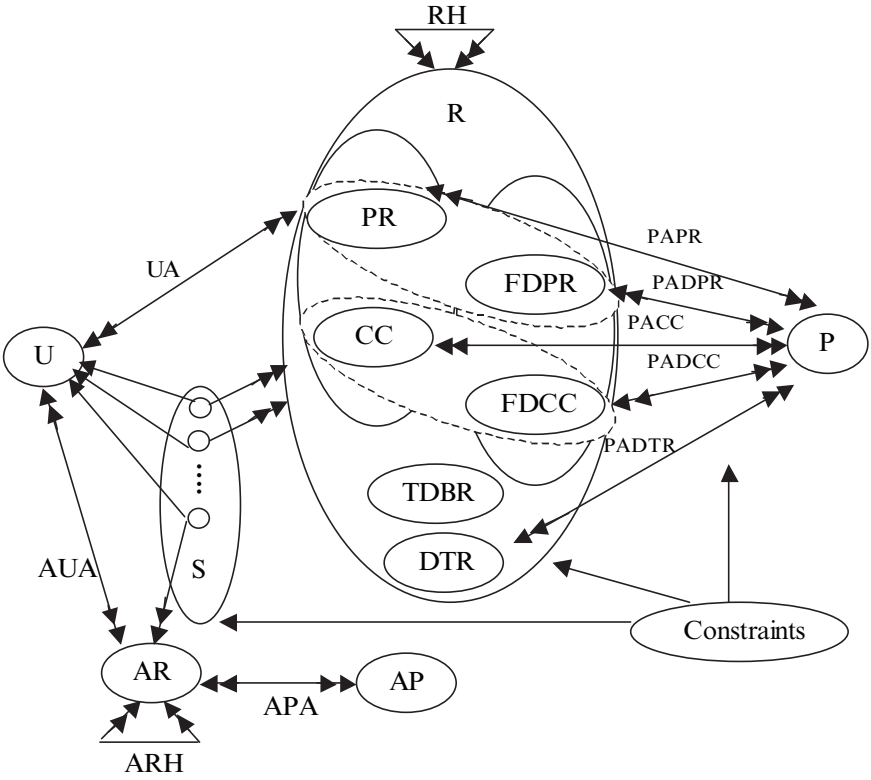


Fig. 12.1 ASRBDM model

avoid redundant permission assignment, administrative permissions can be inherited from the administrative role hierarchy.

Where

U: user
UA: user assignment
AUA: administrative user assignment
R: role
AR: administrative role
PR: private role
FDPR: fixed delegatable private role
CC: corporate common role
FDCC: fixed delegatable corporate common role
TDBR: temporary delegatable role
DTR: delegation role
RH: Role Hierarchy
ARH: administrative role hierarchy
P: permission
AP: administrative permission
APA: administrative permission assignment
PAPR: permission-private role assignment
PADPR: permission-fixed delegatable private role assignment
PACC: permission-corporate common role assignment
PADCC: permission-fixed delegatable corporate common role assignment
PADTR: permission-delegation role assignment

In RBAC models, a senior role inherits the permissions of all its junior roles. One of the main security principles of RBAC models is to enable compliance with the least privilege principle. YongHoon et al. introduced the concept of the sub-role to restrict the complete inheritance of junior permissions to senior roles [12]. In AS-RBDM we simplify the sub-role hierarchies and exclude the restricted inheritance sub-role category, as mentioned in [6]. Roles are assigned to one of four categories by considering the characteristics of the permissions assigned to each: private role (PR), fixed delegatable private role (FDPR), corporate common role (CC), and fixed delegatable corporate common role (FDCC). Accordingly, PA is separated into five partitions which are: permission-private role assignment (PAPR), permission-fixed delegatable private role assignment (PADPR), permission-corporate common role assignment (PACC), permission-fixed delegatable corporate common role assignment (PADCC), and permission-delegation role assignment (PADTR). A security administrator can assign permissions to adequate sub-roles by considering the characteristics of permissions such as inheritance and delegation. Figure 12.2 depicts an example of role hierarchy. Senior roles can inherit permissions only from a common sub-role of the junior roles, and then constraints of role hierarchy can be achieved. By dividing a role into private and common sub-role categories, permission inheritance can be restricted to satisfy the least privilege principle and prohibit an overuse of rights.

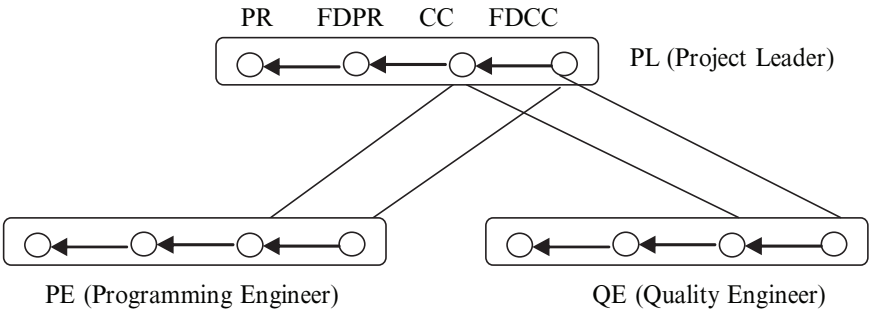


Fig. 12.2 An example of role hierarchy

Table 12.1 An example of role hierarchy policy

```
  <RoleHierarchyPolicy>
<RoleSpec Type="pmiRole" OID="1.2.826.0.1.3344810.1.1.14">
  <SupRole Value="PL_PR">
  <SubRole1 Value="PL_FDPR">
  <SubRole2 Value="PL_CC">
  <SubRole3 Value="PL_FDCC"/>
  </SubRole2>
              </SubRole1>
  </SupRole>
      <SupRole Value="PL_CC">
  <SubRole1 Value="PE _CC"/>
  <SubRole2 Value="QE_CC"/>
  </SupRole>
      <SupRole Value="PL_FDCC">
  <SubRole1 Value="PE_FDCC"/>
  <SubRole2 Value="QE _FDCC"/>
  </SupRole>
  </RoleSpec>
  </RoleHierarchyPolicy>
```

12.3.2 Role Hierarchy Policy

Role hierarchy policy specifies the different roles and their hierarchical relationships to each other. Table 12.1 is an example of role hierarchy policy for the example depicted in Fig. 12.2. Each role is assigned a value, which is the name of the role. The interrelationships among the roles fall into two different classes: superior role (SupRole) and subordinate role (SubRole). Each role hierarchy (RoleSpec in the DTD) supports multiple superior roles inheriting the privileges of a common subordinate role.

12.4 Implement Details

12.4.1 Adopting an Attribute Certificate

One of the main advantages of an attribute certificate is that it can provide the means to transport authorization information in distributed applications. In this paper, we use three kinds of attribute certificates: role assignment attribute certificates (RAC), policy attribute certificates (PAC), and delegate attribute certificates (DAC). The proposed ASRBDM model can support user-to-user delegation which is implemented by assigning DACs, and role-to-role delegation which is implemented by assigning PACs. The role assignment attribute certificate assigns various roles to the users. Its AC holder is the user, and the privilege attributes are the roles assigned to the user. The policy attribute certificate holds digitally signed authorization policies. It is a standard self-signed X.509 AC. Its holder and issuer names are the same (i.e., that of the SOA), the attribute type is the pmiXMLPolicy, and the attribute value is the XML policy.

Delegation information is not directly carried into role assignment certificates due to its short lifetime. On the contrary, we build a new delegate attribute certificate which defines a new DelegateIdentifier extension field on the basis of the role assignment certificate [13]. This field conveys delegation information between the issuer and the holder of the certificate (see Fig. 12.3).

The DelegateInfo sequence contains delegation information between delegator and delegatee. We use a Boolean variable, Delegatable, to distinguish delegation from authorization. We also include TrustValue, a real number in the interval [0, 1],

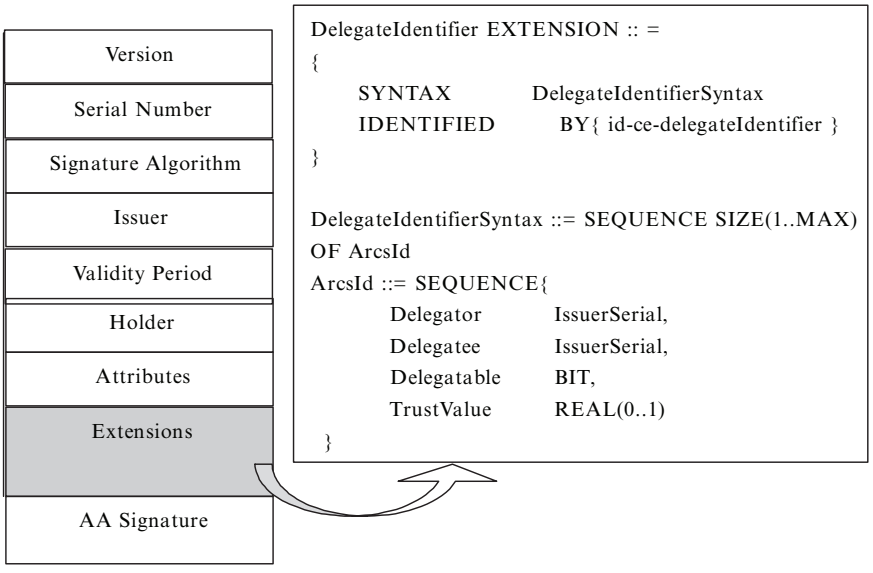


Fig. 12.3 Delegate attribute certificate

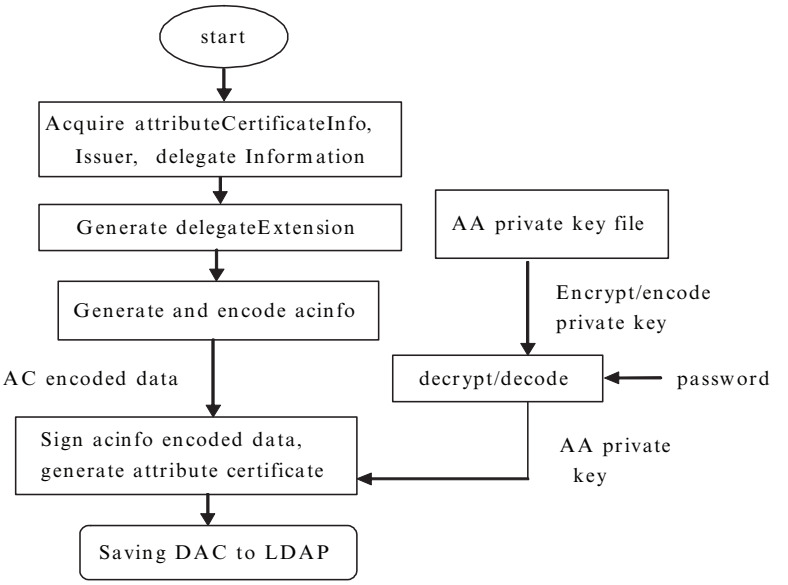


Fig. 12.4 DAC generation process

to measure the level of confidence of the issuer in the issued certificate. In particular the Delegatee field will be the same as that in the holder component of the delegate attribute certificate referenced by this extension. The delegate attribute certificate generation process is shown in Fig. 12.4.

12.4.2 Implementation Architecture of the Model

PERMIS focused on the roles model and cannot support permission level delegation between subjects. Our sub-role-based authorization delegation system (ADS) has been developed in order to support authorization and delegation simultaneously based upon the proposed ASRBDM model. The ADS issues delegate attribute certificates on behalf of authorized delegators who request to delegate a subset of their attributes to a colleague or a subordinate. The basic components of the authorization delegation system as depicted in Fig. 12.5 include:

Policy Editor is a graphical user interface to generate an XML file for the ASRBDM policy. It also includes LDAP entries for subjects, roles, actions, and resources.

Attribute Registration Authority (ARA) is responsible for validation of authorization and delegation requests. To improve the efficiency and security of the system, we adopt ARA to share in the work of attribute authority. ARA operators use ARA to audit attribute certificate requests received from the client proxy server.

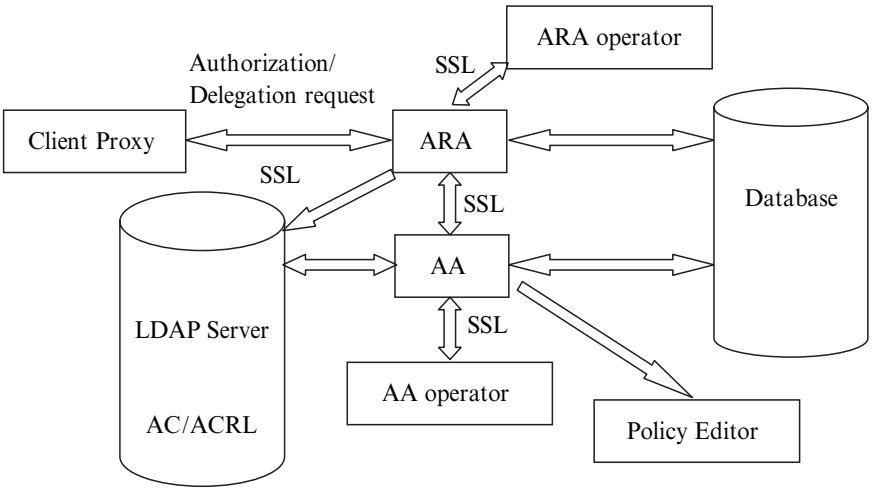


Fig. 12.5 Architecture of authorization delegation system

Attribute Authority (AA) is a tool used for viewing, creating, editing, and revoking attribute certificates. The AA operators use AA to produce and read DER/BER-encoded attribute certificates that conform to the ITU-T X.509 recommendation. It has been developed as a transaction server using a client-server schema and thread-based.

12.4.3 Authorization Delegation Procedure

The authorization and delegation system can issue attribute certificates in batches, which simplifies the privilege management infrastructure effectively. Users are empowered to delegate their attributes to anyone else in their domain at any time without any human or administrative intervention. To ensure secure communications between components, we adopt SSL to ensure mutual authentication, message integrity, and message confidentiality. Communication between user and AA is composed of three phrases:

- The client proxy server is used for user authentication. The authentication can be implemented using a traditional username/password pair or each user's PKC. Once the user passes the authentication, a session will be set for him and his requests will be sent to ARA.
- ARA first accesses one or more LDAP directories to retrieve the policy AC, role ACs, and the delegate ACs for the user, and bases its decision on these. If a user requests to delegate authority over resources to another trusted entity, ARA will create a delegation role (DTR) and assign the permitted delegatable permissions to DTR. Finally, all the request information will be encapsulated in a PKCS#10 certificate request which is forwarded to the AA server.

- The AA operator uses the AA server to allocate roles to users in the form of role assignment ACs. The role attribute in RAC contains only the private sub-role of the request role. PR is a top sub-role category in the horizontal hierarchy, which has all permissions in sub-roles of horizontal hierarchy. If a user requests to delegate authority to another entity to carry out some functions, AA will issue a delegate attribute certificate. AA creates a temporary delegatable role (TDBR) which contains the permissions received from a DTR as delegator with a role-to-role assignment. Then all the attribute certificates created by AA will be stored in a publicly accessible LDAP directory.

12.4.4 Saving ACs to LDAP

All the user X.509 attribute certificates issued by the attribute authority are stored in one or more LDAP servers. Currently LDAP can support only the X.509 identity certificates, and the X.509 ACs cannot be stored in LDAP directly. To allow X.509 ACs to be stored in the holder's entry, we must modify the schema of the LDAP server. The contents of/etc/openldap/schema/pmi.schema are as follows:

```
attributetype (2.5.4.58 NAME 'attributeCertificateAttribute'
DESC 'A binary attribute certificate'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.8)
attributetype (1.2.826.0.1.3344810.1.1.14 NAME 'pmiRole'
DESC 'A pmiRole to be authorized to users'
SUP name)
objectclass (2.5.6.24 NAME 'pmiUser'
SUP top AUXILIARY
DESC 'a pmi entity that can contain X509 ACs'
MUST (uid $
        version $
serialNumber $
        validityNotBefore $
        validityNotAfter $
        attributeCertificateAttribute)
MAY (HolderPKCSerial $
        HolderPKCIssuerDN $
        HolderDN $
        HolderDNSName $
        IssuerDN $
        IssuerDNSName $
        pmiRole))
```

The holders and issuers of ACs can be identified by their X.500 general name (usually an X.500 distinguished name or IP address, URI, or email address) or by reference to their PKC (issuer name and serial number) [14]. Each holder's

LDAP entry should be of object class pmiUser. ACs can be revoked using the serial number, which is the same as the identity certificate. The schema used to store and retrieve the attribute certificate revocation list (ACRL) is specified in the standard [15]:

(2.5.6.19 NAME 'cRLDistributionPoint'
SUP top STRUCTURAL
MUST (cn)
MAY (certificateRevocationList $
authorityRevocationList $
deltaRevocationList))
(2.5.4.39 NAME 'certificateRevocationList'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.9).

12.5 Conclusions

In this paper we propose a new self-administrative sub-role-based delegation model, which supports user-to-user and role-to-role delegations. This model introduces the concepts of administrative role and administrative role hierarchy, and decentralizes downward permissions that often are highly centralized within a small team of security administrators. This gives us clear separation of security administration and delegation, and simplifies management of permissions greatly. In addition, we demonstrate the feasibility of our proposed model through a proof-of-concept implementation. Our authorization delegation system supports both role and permission level delegation. This makes authorization and delegation mechanisms more convenient and flexible.

References

1. Sandhu R, Coyne E, Feinstein H, Youman C (1996) Role-based access control models. IEEE Computer Society Press 29:38–47
2. Barka E, Sandhu R (2000) A Role-based delegation model and some extensions. In: Proc of 23rd national information systems security conference (NISSC2000)
3. Sandhu R, Bhamidipati V, Munawer Q (1999) The ARBAC97 model for role-based administration of roles. In: Proc of 1st ACM transactions on information and system security. ACM Press, New York, pp 105–135
4. Barka E, Sandhu R (2000) Framework for role-based delegation models. In: Proceedings of 16th annual computer security application conference (ACSAC 2000)
5. Longhua Z, Gail-Joon A, Bei-Tseng C (2002) A Rule-based framework for role-based delegation. In: Proceedings of 6th ACM symposium on access control models and technologies (SACMAT 2001). ACM Press, New York, pp 153–162
6. Zhang X, Oh S, Sandhu R (2003) PBDM: A Flexible delegation model in RBAC. In: Ferrari E, Ferraiolo D (eds) Proceedings of the 8th ACM symposium on access control models and technologies. ACM Press, New York, pp 149–157

7. Park D-G, Lee Y-R (2005) A Flexible role-based delegation model using characteristics of permissions. Springer-Verlag, Korea, pp 310–323
8. ITU-T Recommendation X.509, Information Technology: Open Systems Interconnection-The Directory: Public-Key And Attribute Certificate Frameworks, 2000
9. *http://sec.isi.salford.ac.uk/permis/*
10. Chadwick DW, Otenko A (2003) The PERMIS X.509 role based privilege management infrastructure. Future Generation Computer Systems, 19:277–289
11. Chadwick DW, Otenko A (2002) RBAC Policies in XML for X.509 based privilege management. In: Proceedings of IFIPTC11 17th international conference on information security (SEC 2002), pp 39–53
12. Yi YH, Kim M (2003) Applying RBAC providing restricted permission inheritance to a corporate web environment. APWeb Conference, Lecture Notes in Computer Science (LNCS) 2642, pp 287–292
13. Agudo I, Lopez J, Montenegro JA (2005) A Representation model of trust relationships with delegation extensions. Springer-Verlag, Spain, pp 116–130
14. Montenegro JA, Moya F (2004) A Practical approach of X.509 attribute certificate framework as support to obtain privilege delegation. Springer-Verlag, Spain, pp 160–172
15. Wahl M (1997) A Summary of the X.500(96) user schema for use with LDAPv3, RFC 2256

Chapter 13
An Intelligent Maintenance System with Open Source Software

Takumi Ichimura, Yoshiaki Kurosawa, Akira Hara, and Kenneth J. Mackin

13.1 Introduction

Recent computer systems are popularly composed of two or more computer servers connected via LAN (Local Area Network), where each computer server individually serves a specific function. The computer system provides the required service via the combination of individual server operations. When running such networked systems, a system administrator must monitor the occurrences of errors. However, there are often a lack of engineers who have expert knowledge in all fields related to the system, including hardware and software knowledge.

Each computer is equipped with various hardware components, such as a CPU, memory, and hard disk drives. If any one of such parts breaks down, the computer will not operate. When the faulty component carries out various functions within the computer, it is difficult to specify which part is broken. Moreover, it is difficult to detect whether the state of the total system is stable or not, because various software applications work in each computer and many computers connected via LAN cooperate to realize the total service.

In this chapter, we propose a web-based management tool for maintaining computers in a LAN environment. The proposed tool has two main features. One function is to detect hardware faults by utilizing Cacti [1]. Cacti is a complete network graphing solution designed to harness the power of RRDTool (Round Robin Database Tool)'s data storage and graphing functionality. The other function is to detect software errors from abnormal state messages appearing in system LOG files not only in the operating system (OS) but also in the application software. The system

Takumi Ichimura, Yoshiaki Kurosawa, and Akira Hara
Department of Intelligent Systems, Graduate School of Information Sciences, Hiroshima City University, 3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731-3194 Japan,
e-mail: {ichimura,kurosawa,ahara}@its.hiroshima-cu.ac.jp

Kenneth J. Mackin
Department of Information Systems, Tokyo University of Information Sciences, 1200-2, Yatoh-cho, Wakaba-ku, Chiba, 265-8501,
e-mail: mackin@rsch.tuis.ac.jp

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

finds abnormal states using diagnostic rules that are extracted by *automatically defined groups* (ADG) [2, 3]. Moreover, the developed system notifies administrators by sending an e-mail if it finds an abnormal state in the hardware or software. The developed system is successfully operated in university computer labs with more than 300 computers and 20 servers, without any human monitors.

Section 13.2 describes open source software. Section 13.3 describes the notation of the LOG and its analysis method. Section 13.4 explains the algorithm of ADG and its knowledge extraction method. Section 13.5 explains the developed web-based management tool.

13.2 Intelligent Systems in Open Source Software

13.2.1 Open Source

Open source is a concept of license agreement of software that enables the code to be open to the public while protecting the copyright of the programmer. According to *The Open Source Definition (OSD)* [4] by *Open Source Initiative (OSI)* [5] the open source license requires (see [4] for details):

- Free redistribution
- Source code
- Derived works
- Integrity of the author's source code
- No discrimination against persons or groups
- No discrimination against fields of endeavor
- Distribution of license
- License must not be specific to a product
- License must not restrict other software
- License must be technology-neutral

In Japan, however, the term *open source* is widely used in various forms of software licenses that emphasize that the source code is distributed free of charge, even if the software does not fulfill the aforementioned requirements.

Recently, Linux [6] has been widely used in open source fields. The Linux kernel is distributed according to the GNU General Public License (GPL) [7]. The license permits free copying, distribution, and modification under the GPL.

13.2.2 Communication of Bazaar

In addition to the features of open source, the Bazaar style led Linux to success. The Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches [8].

There remain questions about preconditions for successful bazaar-style development, including both the qualifications of the project leader and the state of source code, when the developer team goes public and starts to build a codeveloper community. It is often quoted that source code cannot be coded from scratch in the bazaar style. Of course, basic skills of design and coding are required, but anybody launching a bazaar effort will already have reached the minimum level.

To build a development community, you need to attract people, make them interested in what you are doing, and keep them happy about the amount of work they are doing. The center of a bazaar is the abovementioned project manager, who is supported by the team members.

We have launched a Linux community in Hiroshima, Japan, with the support of Hiroshima City. When we started to build a new community, we concurrently held a summer beginners course in the use of Linux, and showed the members the goals of the research. An advanced course for system engineers was also held with the support of our university to increase the skills of the team members.

13.2.3 Products in Open Source

With widespread use of personal computers and the Internet, the basic idea of open source has made Linux known to the world. Recently, it has become possible to purchase high performance IBM-AT compatible machines at low prices, and Linux shows no inferiority in performance as a server OS compared with other UNIX-based operating systems. The features of a Linux server allows it to be used as web, database, printer, and storage servers. Moreover, the compact Linux kernel has attracted attention as a open source platform for built-in OS for limited resource devices such as mobile phones, PDA, ubiquitous computing, and so on.

Currently, there are many software and hardware products referring to open source. But as the system becomes more complex, it becomes more difficult to use as open source, because the source code is also more complex to analyze. However, for traditional software products developed by software companies, the source code is not open to the public. As a result, the design policy in software development companies pursues only their own profit. Oppositely, open source is often adopted by educational institutions and venture companies. With the open source philosophy in mind, the open source project will base the design policy on users, leading to a highly valued product.

There is the argument that for traditional software, the software vender can be held legally liable, and potentially recover compensation if the product causes financial damage. But this is an illusion because most software licenses disclaim even warranty of merchantability, let alone performance.

In Japan, the Information Technology Promotion Agency (IPA) [9] supports engineer education, and provides a grants for the development of open source software. In the department of open source software center development [10], there are many useful open source software options with respect to local language. After the

development period, the developed software is opened to the public via the Internet. Further support is expected by the system engineers in the field of open source software.

13.2.4 Intelligent Systems in Open Source

A systems engineer who has been trained for sophisticated technologies and has a mind for open source software, would desire to develop a product based on the open source software. The engineer will naturally look for new ideas or technology to add to the open source community, and hope that his product will become widely used. The engineer may turn to research in the universities and other institutions for new ideas and advanced technologies to use in the open source product.

On the other hand, researchers in universities and other research institutions are always looking for applications and routes to spread the results of their research. Many intelligent algorithms in the field of soft computing have high potential to be effective and useful in actual practice. If the researcher designs a voluntary contribution of practical use and the system engineer with high development skills develops a product with open source software, the product will shed new light upon the domain of soft computing.

Our research described in the next sections will be on the intelligent method of natural language processing and evolutionary algorithms. The maintenance method uses open source, and the web-based maintenance system for computer servers uses the combination of both intelligent technologies.

13.3 The System Log and its Analysis

Recently, Intel architecture (IA) servers have become the main stream of Internet servers instead of UNIX workstations, with the spread of LINUX as an alternative Portable Operating System Interface for UNIX (POSIX) OS. However, Windows is also often used as the OS for Internet servers. In each OS, the system records the log file in a different type of format. In this section, we explain how to deal with the different log files in our proposed method.

13.3.1 Description of System Log Files

A syslog protocol has been proposed by the RFC3164 [11]. According to this document, a system log file records messages that are generated by specified software operation records. However, the detailed notations are not strictly defined. Therefore, actual descriptions, as well as the definition of logs, include many discrepancies in

the log files. Figure 13.1 shows four valid sample logs described in the form of RFC 3164. Figure 13.2 shows a log file in Windows. In this example, the file includes the specified characters in the selected local language.

In a multi-platform environment, system administrators must have the appropriate expertise to manage both systems. As shown in Fig. 13.2, if there are two or more languages, it becomes more difficult to analyze.

For example, as shown in Fig. 13.3, there are three types of log files for the http protocol: common log format (CLF), combined log format, and World Wide Web Consortium (W3C) [12].

When the administrator checks each OS for stable operation, the administrator may be confronted with at least the following three kinds of problems [2, 3].

- **Lack of expert knowledge of the overall system**
  In Japan, system engineers (SEs) have rich knowledge concerning Microsoft Windows. However, this knowledge is limited to the software applications, not

```
Example 1  Oct 11 22:14:15 mymachine su: 'su root' failed for lonvick on /dev/pts/8
Example 2  "Use the BFG!"
Example 3  Aug 24 05:34:00 CST 1987 mymachine myproc[10]: %% It's time to make the do-nuts.
           %% Ingredients: Mix=OK, Jelly=OK # Devices: Mixer=OK, Jelly_Injector=OK, Frier=OK
           # Transport: Conveyer1=OK, Conveyer2=OK # %%
Example 4  1990 Oct 22 10:52:01 TZ-6 scapegoat.dmz.example.org 10.1.2.3 sched[0]: That's All Folks!
```

Fig. 13.1 Sample log files in RFC3164

```
2005/11/13,17:41:27,Service Control Manager,情報,7036,N/A,
    ADSV,"WinHTTP Web Proxy Auto-Discovery Service
        サービスは、停止状態に入りました。"
2005/11/13,17:41:27,WinHttpAutoProxySvc,情報, 12517,N/A,
    ADSV,"The WinHTTP Web Proxy Auto-Discovery
        Service suspended operation."
2005/11/13,17:41:27,WinHttpAutoProxySvc,情報, 12503,N/A,
    ADSV,"The WinHTTP Web Proxy Auto-Discovery
         Service has been idle for 15 minutes, it will be shut down."
2005/11/13,17:24:57,Service Control Manager,情報, 7036,N/A,
    ADSV,"WinHTTP Web Proxy Auto-Discovery Service
        サービスは、実行中 状態に入りました。"
```

Fig. 13.2 Sample log in WIN

```
< Common Log Format >
      127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700]
            "GET /apache_pb.gif HTTP/1.0" 200 2326

< Combined Log Format >
      127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700]
            "GET /apache_pb.gif HTTP/1.0" 200 2326
                  "http://www.example.com/start.html"
                  "Mozilla/4.08 [en] (Win98; I ;Nav)"

< Extended Log File Format >
      00:34:23 GET /foo/bar.html
```

Fig. 13.3 Sample of html log message

for the overall OS. That is, the SE does not necessarily know detailed operations within Windows, because it is difficult to know details of the closed Windows operating system. Therefore, when serious trouble (such as system errors) arises, the SE can only retrieve limited information to track errors within the developed product.

On the other hand, open source software (OSS) such as Linux is wide spread and SEs can freely access the source code. If the SE does not have high programming skills and does not understand the Linux kernel and related hardware components, however, the SE may not be able to completely understand the OSS system. Therefore, a high-level engineer in the domain of OSS requires the specified expertise training after the initial professional training period.

- **Enormous size of data in the log file**
  The second problem is the large size of log files. Many services run simultaneously in the computer server and the size of log files grows considerably every day because every event is recorded. It is difficult for the administrators to monitor each event. An enormous cost is required, because the time to review the log file becomes longer than the time to generate the files. Software such as LogWatch [13] solves the problem by checking the log file for records that match predetermined rules. However, rules for all the software are not initially defined. Moreover, the rules are written using regular expressions, and it is not easy to determine the rules.
- **Lack of knowledge due to system change**
  The causal relationship knowledge may become incomplete due to changes in the system environment. For example, when the operating system is updated for security reasons and so on, the configuration file may be changed to add some functions. In such a case, it is necessary to add either current knowledge or new knowledge, or to correct part of the rules.

13.3.2 Log Format and Analysis

The administrator should be alerted immediately about the detailed state of the system as soon as possible, and it is necessary to clarify the solution of the causes of failure. To support the administrators, we devised a method of error diagnostic rules using evolutionary search algorithms. Before the evolutionary search process, the system needs to preprocess the log file for the analysis on the basis of regular expressions.

Regular expression is a pattern-matching description method used to specify letters, words, or phrases. Using this method, we can find simple and efficient patterns included in target phrases and so on. Let us compare two samples (Example 1 and Example 4) in Fig. 13.1. For example, regular expression `^[0-9]*\sOct\s[0-9]{2}` can deal with both strings of these samples. The expressions `^`, [0-9], `\s`, *, and `{2}` mean *beginning of the string*, *from 0 to 9, white space, matching 0 or more times,* and *matching 2 times,* respectively. In addition, an abbreviation such as Oct means

October, but an abbreviation does not relate to regular expressions. Therefore, the index for the two strings are united and then labeled as one tag such as <DATE>.

An administrator without enough detailed knowledge may cause a breach in the normal state of computer operations. In the worst case, the administrator may cause the computer system to unnecessarily reconstruct the overall system. In general, the administrator should understand the log format and its contents. To provide expert knowledge to system administrators, analysis techniques for logs in any format and content are required. However, the analysis should be limited to assume minimum knowledge for the administrators, because the log files are recorded as output of various application operations, as well as many different types of software. In the domain of open source software, most of the log is the same format. The description has some fields delimited by commas. Strings with a clear meaning in this field are put in the symbols < >.

The knowledge description method defines some tags such as <DATE> and <TIME>. There are about 40 tags in this chapter, which include network information such as <HOSTNAME>, <REMOTEHOST>, <IP>, amd <PORT>. Table 13.1 shows some examples of tags.

The necessity of detailed knowledge prompts the administrators to manage their systems. Therefore, we should analyze the logs assuming as little knowledge as possible. Thus, we give some tags such as <DATE> and <TIME> to some fields such as date and time because we can easily understand the meaning of fields. Furthermore, we can give <HOSTNAME> or <REMOTEHOST> tags to hostname fields by using appropriate commands such as *ifconfig*.

Table 13.1 Examples of tags

Name	Description
ACCESS_MODE	mode by which the user is logged in
AUTH_METHOD	method of authentication used
AUTH_UID	user ID returned by the authentication method
COMPLETION	single character indicating the status of the transfer
DAEMON	daemon name as a sender of the log message
DATA	numerical data
DATASIZE	size of the transferred data
DATE	current local time when logging
DIRECTION	direction of the transfer
EXP	explanation of some sort
FILENAME	filename of the target file
FILESIZE	size of the transferred file
FILETYPE	single character indicating the type of transfer
FUNC	called function when service is executed
GROUP	group which the user belongs to
HOST	host name of the computer
HTTP_COMMAND	HTTP methods such as GET and PUT
HTTP_REQUEST	HTTP request such as the target URL
HTTP_RESULT	HTTP status code such as the target URL
HTTP_VERSION	HTTP version number such as HTTP/1.1

When we do not understand the field in which the description is included, we may give only a <EXP> tag. For instance, of tag <EXP> we now have the following the record in the log file.

```
20070515, 14:15:20, Sent to 192.168.1.1
```

The record is added <DATE>, <TIME>, and <EXP> as follows.

```
<DATE>20070515</DATE>, <TIME>14:15:20</TIME>,
<EXP>Sent to 192.168.1.1</EXP>
```

We should notice that once a tag is given, if the sentence in a tag field includes some other information, the sentence is not divided into any further detail. Therefore, the abovementioned example is not transferred into <EXP>Sent to</EXP><REMHOST>192.168.1.1</REMHOST>. The analyzer finds the comma as the delimitation of data, and the tag addition is implemented. This preprocessing is limited to be simple because expert knowledge is not required.

13.4 Rule Extraction by ADG

13.4.1 Automatically Defined Groups

In the field of data processing, clustering the enormous data and then extracting common characteristics from each clustered data is important for knowledge acquisition. A multi-agent approach is effective for this task. In the approach, agents compete with one another for their share of the data, and each agent generates a rule for the assigned data. As a result, all rules are extracted by multi-agent cooperation. To realize this approach, agents have to discover the number of the latent rules in the given data and the descriptions of the respective rules. For this purpose, we have proposed an improved genetic programming (GP) method—automatically defined groups (ADG). This method optimizes both the grouping of agents and the tree structural rule for each group in the process of evolution. Agents in the same group search for a single rule. Therefore, we can prevent the increases in search space, and finally the number of groups corresponds to the number of discovered rules.

In ADG, each individual consists of the predefined number of agents. The individual maintains multiple trees, each of which functions as a specialized program for a distinct group as shown in Fig. 13.4. We define a group as the set of agents referring to the same tree.

Generating an initial population, agents in each individual are divided into several groups at random. Crossover operations are restricted to corresponding tree pairs. For example, a tree referred to by Agent 1 in an individual breeds with a tree referred to by Agent 1 in another individual. With the crossover operation, the group structure is also optimized by dividing or unifying the groups according to the inclusion relationship of the set of agents.

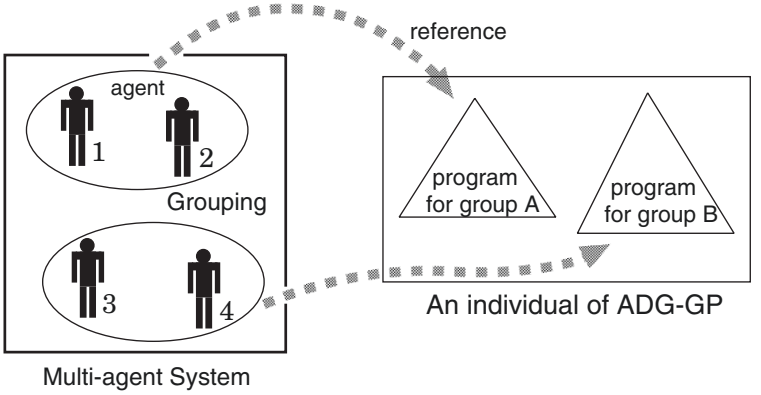


Fig. 13.4 Concept of automatically defined groups

The concrete processes for the group optimization are as follows: We arbitrarily choose an agent for two parental individuals. A tree referred to by the agent in each individual is used for crossover. We use T and T' as expressions of these trees, respectively. In each parental individual, we choose a set $A(T)$—the set of agents that refer to the selected tree T. When we perform a crossover operation on trees T and T', there are the following three cases:

(1) If the relationship of the sets is $A(T) = A(T')$, the structure of each individual is unchanged.
(2) If the relationship of the sets is $A(T) \supset A(T')$, the division of groups takes place in the individual with T, so that the only tree referred to by the agents in $A(T) \cap A(T')$ can be used for crossover. The individual that maintains T' is unchanged. Figure 13.5 (Type B) indicates an example of this type of crossover.
(3) If the relationship of the sets is $A(T) \not\supset A(T')$ and $A(T) \not\subset A(T')$, the unification of groups takes place in both individuals so that the agents in $A(T) \cup A(T')$ can refer to an identical tree. Figure 13.5 (Type C) shows an example of this crossover.

The adequate group structure is automatically acquired by using these operations.

13.4.2 Rule Extraction from Classified Data

In some kinds of databases, data is classified into positive or negative cases (or more than two categories). It is an important task to extract characteristics for a target class. However, even if data belong to the same class, all the data in the class do not necessarily have the same characteristics. Part of a data set might show different characteristics. It is possible to apply ADG to rule extraction from such classified data. In ADG, agents divide the same class data into several subsets adaptively,

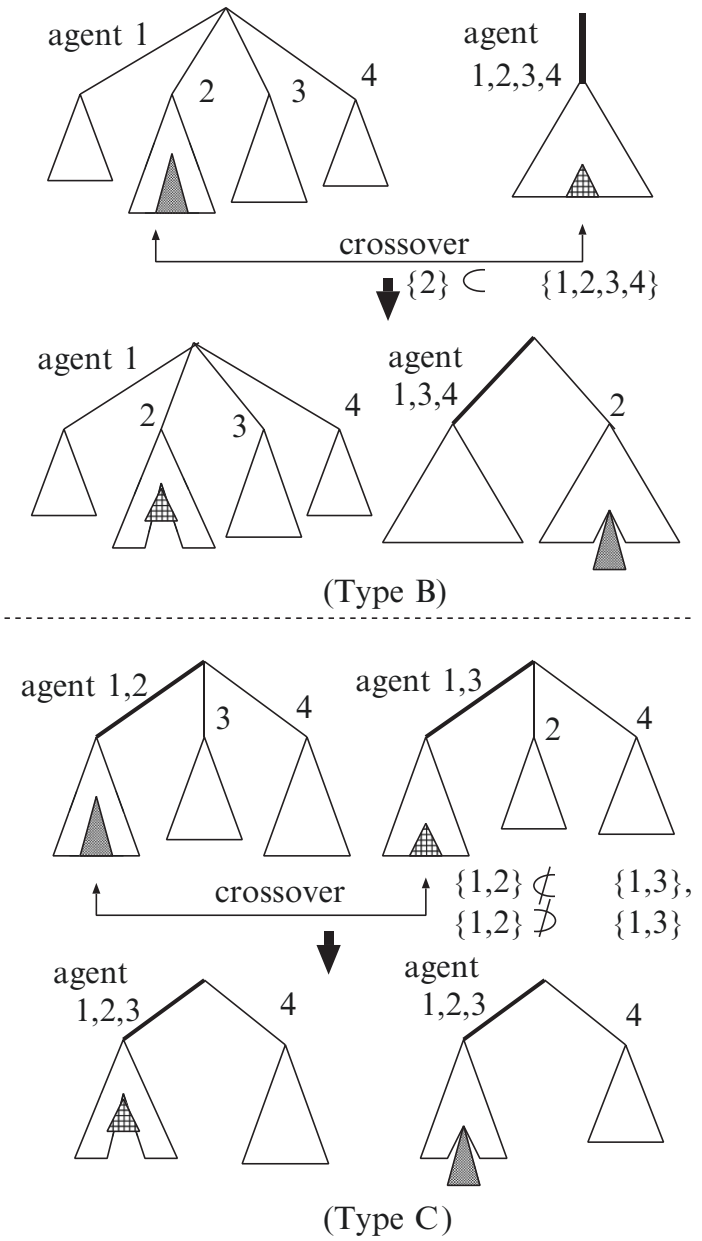


Fig. 13.5 Examples of crossover

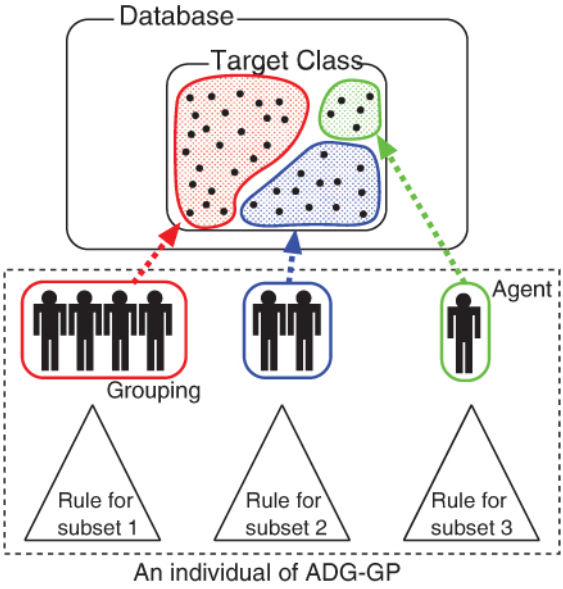


Fig. 13.6 Rule extraction using ADG

and rules for respective subsets are extracted. Figure 13.6 shows a sketch of rule extraction using ADG. Each agent group extracts a rule for the divided subset, and the rules acquired by multiple groups can cover all the data in the target class.

When the fitness of each individual is evaluated, the load of each agent and predictive accuracy of its rule are considered. As a result, a lot of agents come to belong to the group with the high-use frequency and high-accuracy rule. In other words, we can regard the number of agents in each group as the degree of importance for the rule. Thus, two or more rules and the degree of importance of respective rules can be acquired at the same time. This method was applied to the medical data, and the effectiveness was verified [15, 16].

13.4.3 Application to Knowledge Acquisition from Log Files

We apply the rule extraction method using ADG to detect troubles in computer systems from log files. To use the method described in the previous section, we need the supervised information for the learning phase. In other words, we have to classify each message in the log files into two classes—normal message class and abnormal message class indicating system trouble. However, this is a troublesome task because complete knowledge of computer administration is required and the log data may be an enormous size. To automatically classify log messages into the appropriate class, we consider state transition pattern of computer system operation.

We focus on the following two different states and make use of the difference between the states instead of the supervised information.

1) **Normal State**
   This is a state in the period of stable operation of the computer system. For example, frequently observed messages (e.g., "Successfully access," "File was opened," etc.) are not concerned with the error.
2) **Abnormal State**
   This is a state in the period of unstable operation of the computer system. The transition to the abnormal state may happen because of hardware trouble, such as hard disk drive errors, or by starting service programs with wrong configurations. In this state, many error messages (e.g., "I/O error," "Access denied," "File Not Found," etc.) are included in the log files. Of course, the messages observed in the normal state can also appear in abnormal state.

The extraction of rules is performed by using log files in the respective states. Each GP individual has multiple trees, each of which represents the logical expression. If one or more trees return true for an input message, the message is regarded as a problematic case. In contrast, if all trees return false, the message is not regarded as a problematic case. By evolutionary computation, we find rules that only respond to the messages appearing in the abnormal state.

The fitness is calculated based on the error-detection accuracy and load balancing among agents. The load is calculated from the adopted frequency of each group's rule and the number of agents in each group. The adopted frequency of each rule is counted when the rule returns true to the messages in the abnormal state log. When the agent a belongs to the group g, the load of the agent w_a is defined as follows:

$$w_a = \frac{f_g}{n^g_{agent}} \tag{13.1}$$

where n^g_{agent} represents the number of agents that belong to the group g, and f_g represents the adopted frequency of g. By balancing every agent's load, more agents are allotted to the group that has a greater frequency of adoption. On the other hand, the number of agents in the less adopted group becomes smaller. Therefore, the number of agents of respective rules indicates how general each rule is for detection of problematic messages. Moreover, when usual messages in the normal state are judged to be problematic messages through a mistake of a rule, it is considered that the number of agents who support the rule ($fault\_agent$) should be small. To satisfy the requirements mentioned above, fitness f is calculated by Eq. 13.2. We maximize this fitness by evolution.

$$f = \frac{H_{Abn}/N_{Abn}}{H_{Norm}/N_{Norm}} - \beta \frac{\sum_{N_{Norm}} fault\_agent}{H_{Norm} \times N_{Agent}} - \delta V_w \tag{13.2}$$

In this equation, N_{Abn} and N_{Norm} represent the number of messages in the abnormal state and normal state, respectively. H_{Abn} and H_{Norm} represent the frequency that one or more trees in the individual return true for a abnormal state log and normal

state log, respectively. N_{Agent} is the number of agents in the individual, and V_w is the variance of every agent's load.

13.5 Web-Based Management Tool

In this chapter, we propose a web-based management tool of operating computers that are connected via LAN. The proposed system has two new features. One function is to detect hardware faults by using Cacti [1]—open source software. Cacti is a complete network graphing solution designed to harness the power of round robin database tool's (RRDTool) data storage and graphing functionality [17]. The other function is to detect software errors using abnormal messages appeared in LOG files, not only in the OS but also in application software. The information related to the faults and errors is managed by a web-based tool, and the system also has a feature of sending an e-mail notice to the administrators immediately when the trouble occurs.

13.5.1 Utilization of Open Source Software

Cacti is a useful tool and keeps all of the necessary information to create graphs and populate them with RRDTool data. Cacti handles graphs, data sources, and round robin archives in a database. To gather the data in the network, Cacti uses simple network management protocol (SNMP). Multi router traffic grapher (MRTG) [18] is also developed on the basis of SNMP and creates a graph representing network traffic.

RRDTool is shows better operability compared to MRTG. RRDTool can record the data gathered from all kinds of resources. The analysis part for the stored data of RRDTool is based on the function that generates graphic representations of periodical transitions of collected data.

The management of the system in the LAN shows that Management information base (MIB) of the equipment is acquired or updated through the SNMP. There are two kinds of rules related to MIB: MIB-1 and MIB-2 described by RFC1156 [19] and RFC1213 [20], respectively.

When the administrator selects the data source—server and device—the RRDTool can create a graph of the data transition. In addition, an e-mail is sent to the administrator, when our developed tool recognizes the occurrence of fatal error or when the acquired round robin data is discontinuous. Figure 13.7 shows an example of a graph in the Cacti.

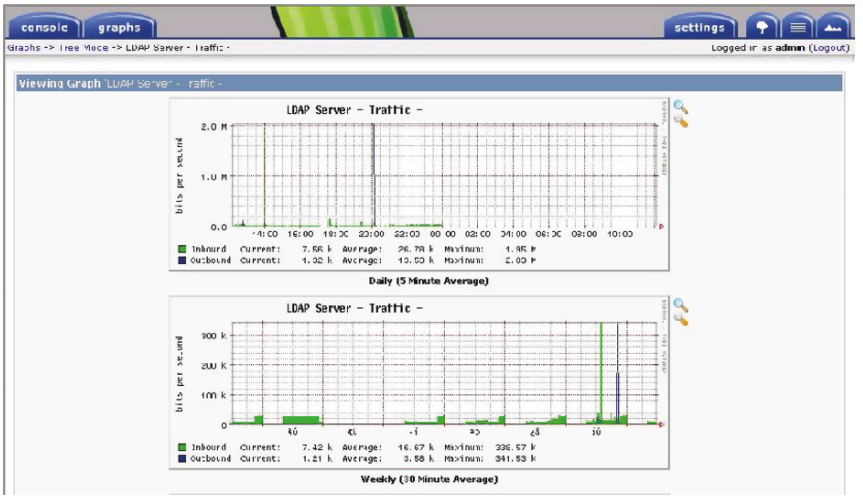


Fig. 13.7 Example of Cacti

13.5.2 Error Detection using Extracted Rules

The system finds a hardware error using RRDTool. On the other hand, the system always monitors not only the system log but application logs not included in the system log, as shown in Fig. 13.8. Such logs include various data records of system operations. That is, there are data records not directly related to the error. Therefore, it is not possible to judge whether an event is an error only from the description because such events may include descriptions never seen before.

Therefore, we must judge whether the record is an error based on the extracted rules by ADG described in prior sections. If the event is judged as an error, the system notifies the administrator by e-mail as shown in Fig. 13.9. The e-mail often includes some original error messages from operating applications This e-mail is also sent to the administrator's mobile phone as shown in Fig. 13.10.

Before introduction of the proposed system, it was very difficult for the system administrator to know when the system was in the abnormal state. With the proposed system, not only the system administrator but the service companies could be alerted of the occurrence of fatal error immediately. The proposed system automatically monitors more than 300 computers and 20 servers in the author's university, and successfully alerts for abnormal hardware and software events, to enable rapid repair and maintenance.

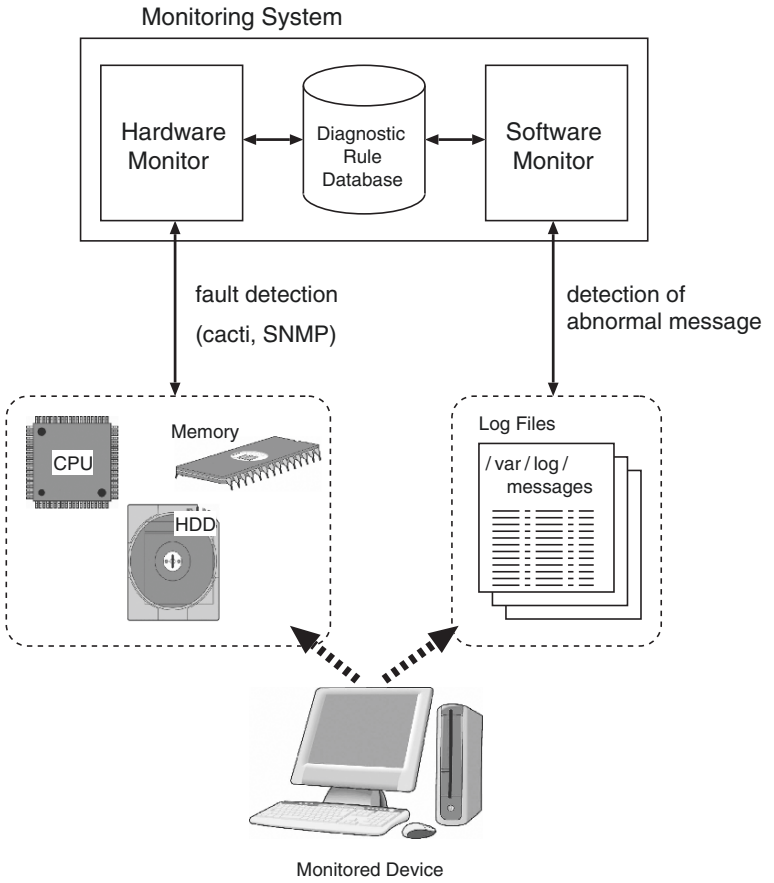


Fig. 13.8 Hardware and software error detection

From:clientmonitor@xxxxx.ac.jp
To:administrator@xxxxx.ac.jp
CC:Support_hiroshima@it-xxxx.co.jp
Subject: Client Boot Error
Date: Mon, 21 May 2007 09:15:00 +0900

This is an automatically generated message from client monitoring system.
The administrators of Computer rooms should check the following PCs.
1)PC Monitoring System detected the booting
failure.
PC102 PC103
2)The network unreachable by ping.
PC102

If you cannot solve the problems, please make a service call
to the maintenance company by e-mail(Support_hiroshima@it-xxxx.co.jp).

Fig. 13.9 Example of error notice email

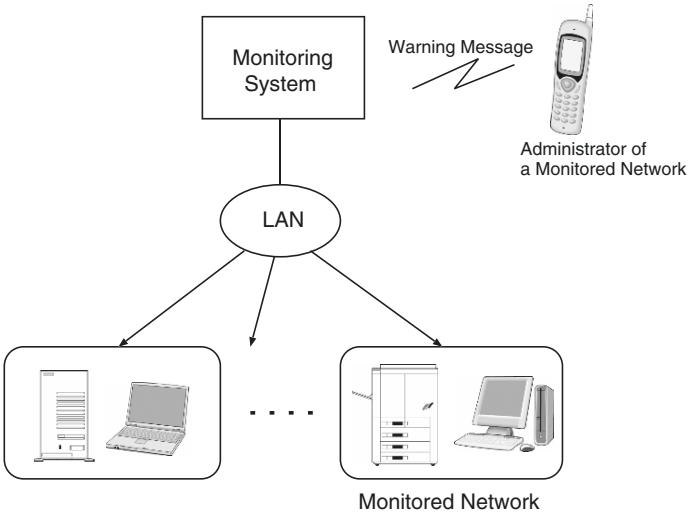


Fig. 13.10 Web-based management tool

13.6 Conclusions and Future Work

Recently computer systems are popularly composed of two or more computer servers connected via a LAN, where the computer system provides the required service by the combination of individual server operations. This chapter describes a web-based maintenance method of computer servers using extracted diagnostic rules for detecting system errors. The system can monitor both hardware and software operations. The system finds abnormal states by using diagnostic rules extracted by ADG.

However, the current extraction of rules by ADG method requires a fair amount of learning time. If an event not included in the training dataset occurs, the system may not correctly diagnose the error occurrence. For future works, we will improve the ADG algorithm to extract the expert knowledge according to the transition of systems related to the change of system configuration.

During the process of the research, we started several communities of open source software under the support of both Hiroshima City and Hiroshima City University. However, these communities ended this year with the end of the original limit of three years. The author wishes to thank the members of these communities.

Many people wrongly think that in such open software communities, the culture of self-directed egoists must be fragmented, territorial, wasteful, secretive, and hostile. This prejudice is clearly a myth, because through such communities each person can learn the open mind of open source software. The author hopes that the future of open source software in Hiroshima will increasingly belong to people who know how to design intelligent systems in open source software, and the people who leave behind the cathedral and embrace the bazaar.

Acknowledgements We would like to thank Toshiba IT Services Corporation—the maintenance support company in Hiroshima City University—for their continuous support. We would like to especially thank Mr. Hiroyuki Sasaki and Mr. Osamu Matsuura—staff in the Department of Hiroshima Support Center—who supported us and our open source mind all the time. We would like to express our gratitude for being able to work together with them.

References

1. Cacti. Open Source Software. http://cacti.net/
2. Hara A, Nagao T (1999) "Emergence of cooperative behavior using ADG; Automatically Defined Groups," *Proc. of the 1999 Genetic and Evolutionary Computation Conf.*, pp. 1039–1046
3. Kurosawa Y, Hara A, Ichimura T, Kawano Y (2006) "Extraction of Error Detection Rules without Supervised Information from Log Files Using Automatically Defined Groups," *Proc. of The 2006 IEEE International Conference on System, Man and Cybernetics*, pp. 5314–5319
4. The Open Source Definition, http://www.opensource.org/docs/osd
5. The Open Source Initiative, http://www.opensource.org/search/node
6. The Linux Kernel Archives, http://www.kernel.org/
7. GNU General Public License, http://www.opensource.org/licenses/gpl-license.php
8. Eric Steven Raymond, "The Cathedral and Bazaar", http://www.tuxedo.org/ esr/
9. Information-Technology Promotion Agency, http://www.ipa.go.jp/index-e.html
10. OSS center in IPA, http://www.ipa.go.jp/software/open/ossc/english/index.html
11. Lonvick C (2001) The BSD Syslog Protocol, RFC3164, August
12. Hallam-Baker PM, Behlendorf B (1996) W3C Working Draft WD-logfile-960323, http://www.w3.org/TR/WD-logfile.html
13. Logwatch, http://www2.logwatch.org:8080
14. The Apache Software Foundation Log Files, Apache HTTP Server Version 2.2 Manual, http://httpd.apache.org/docs/2.2/logs.html, 2006
15. Hara A, Ichimura T, Yoshida K (2005) "Discovering multiple diagnostic rules from coronary heart disease database using automatically defined groups," *Journal of Intelligent Manufacturing*, Vol.16, No.6, pp. 645–661
16. Hara A, Ichimura T, Takahama T, Isomichi Y (2004) "Discovery of Cluster Structure and The Clustering Rules from Medical Database Using ADG; Automatically Defined Groups," *Knowledge-Based Intelligent Systems for Healthcare*, T. Ichimura and K. Yoshida (Eds.), pp. 51–86
17. RRDTool. http://oss.oetiker.ch/rrdtool/
18. MRTG. http://oss.oetiker.ch/mrtg/
19. McCloghrie K (1990) Management Information Base for Network Management of TCP/IP-based internets, RFC1156, May
20. McCloghrie K (1991) Management Information Base for Network Management of TCP/IP-based internets:MIB-II, RFC1156, March

Chapter 14
Further Research on Web Service-Oriented Asynchronous Invoking

Liu Changsheng, Li Changyun, Tang Jingtian, and Cheng Ying

14.1 Introduction

By handling flexible linking and high mobility, Web Service has now increasingly become the road for the technology of integrated distributions and heterogeneous applications [1]. At the same time, service-oriented software developing methods and software framework are becoming the new hotspots in the software engineering field. First of all, SOAP, WSDL, and UDDI have given specifications for message passing, service definition, discovery, and release of Web services [2]. All these specifications lead to a loose coupling, independent-platform model, in which each application has to find out the right one to interact with. Secondly, the maturity of the component based on software development (CBSD) technology provides handling for a quick implementation and distribution of Web services [3]. At last, the theoretical research on software architecture provides the methodological foundations for a formalized description, a comprehension, and an analysis of the higher level organization of the openly distributed application system [4].

The service-oriented software development is focused on two aspects: service design and service integration. Several studies have already been done in this field: Alan Brown and others have put forward a means of setting up software components into software services [3,4], and bringing forward the three-level development process (object -> component -> service); WSFL specifications [5] have kept the business process as the kernel to integrate the Web services provided by the virtual organizations of enterprises on the Internet, and to dynamically build different applications; Z. Maamar [6] has undertaken investigations on using state diagrams to analyze behavioral theory of compound services; Chris Lüer [7] has worked on

Liu Changsheng, Tang Jingtian
Central South University, China

Liu Changsheng, Cheng Ying
ChangSha Aeronautical Vocational and Technical College, China

Li Changyun
Zhuzhou Institute of Technology, China

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

several key points of the runtime environment handling compound services; and Z. Maamar [8] has precisely designed a compound service composition and a runtime environment. Although the contents of a service-oriented software framework have been improved and understood more thoroughly by these studies, there are still several drawbacks left: The service invoking defined by SOAP is naturally synchronous, and the interactive model defined by WSDL is only synchronous, or belongs to an asynchronous interaction state-independent model.

However, all Web services do not run in synchronization; sometimes, the response to a Web service requirement is not immediately given, but is given at some time later when the initial required transaction is done.

More precisely, asynchronous communication between services is needed.

Although WSFL and XLANG can be used for modeling a commercial process, and making the application programs fully integrated over the Internet without considering the different programming languages and runtime environments [5], we are inclined to think that none of these approaches have efficiently solved this problem. In the compound service designed by Z. Maamar [8] the software Agent is used for integrating and executing services, thus making the Web service invoking asynchronous, but Agent has not yet become part of the common information infrastructure, so it is an easy approach.

Therefore, using fundamental frameworks and mechanisms based on the existing Web service standards and component running platforms (J2EE, .NET) to implement asynchronous operations makes much more sense.

In this paper, as far as the problems mentioned above are concerned, we will search for a service-oriented software developing process at first, then we will use the method of "invoking point in infrastructure" to reduce call dependence between services, and then we will design the fundamental framework based on the existing Web service standards and component running platforms in order to implement asynchronous operations.

14.2 The Service-Oriented Software Developing Process

The service-oriented software developing process is shown in Fig. 14.1, the user encapsulates tightly collaborating classes/objects into a standard component modality, such as COM component and EJB. A compounded component or local application

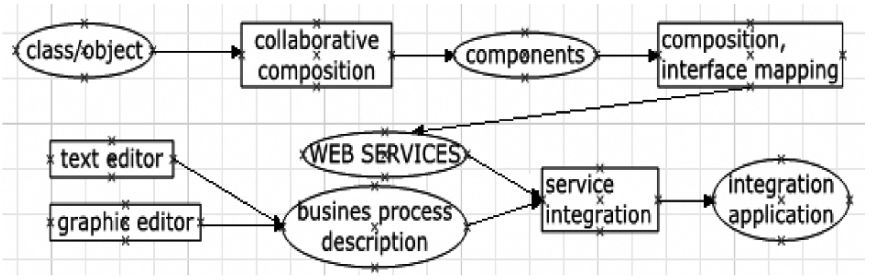


Fig. 14.1 Service-oriented software developing process

can be made up of several components. Components can be mapped onto Web services by interface composing. According to global business process requirements, by using a text editor or graphic editor, describing the operation process specification using some flow modeling language which includes the flow, the activity control flow, and the data flow. Keeping this integrated process model as a center, and mapping the flow activity to the port operations of the Web services, the integrated-distributed Web services can form a loose coupling, and flexible integrated applications that can support all kinds of systems in every field (including B2C, B2B and EAI).

The whole development process can be split into four layers: class/object -> component (including compounded component and local application) -> service -> integrated application. Usually, the process of encapsulating class/object into component is done during the design phase, widely provided by the existent component standards (COM, CORBA) and their development environment.

Component -> service can be divided into 2 subsets: component composition and service mapping. Component composition instantiates the member components of a compounded component, and incorporates the component interface, thus hiding the interaction of components from outside users; more, by providing coarse granularity, and a unified interface, it prevents session-stating from taking care of dependency.

Service mapping describes the component interface in the WSDL format, so that the outside user can invoke the component interface with the SOAP protocol.

Component composition is set at the design time, but can also be done at runtime; and the mapping mechanism from the component interface to the Web services is already given by the component running platform, such as J2EE, .NET, and others. However, these researches and implementations have not set behavior guidelines for component composition and interface mapping to services. And the component interface specification only provides an interface method list, and not any relationship interface method; thus the designer gathers composite components to do appropriate selections and combinations of interface operations.

However, the fact is that the cooperative and relative relationships among interface methods have a great influence on the implementation of the transformation from component to Web services of a single instance.

For instance, the interface of a security management component provides three methods: UserLogin, GetUserAccount, and GetUserDomain, the latter two relying on the first one. UserLogin has to be the first executed, enabling GetUserAccount and GetUserDomain to be run. The implementation of the transformation from component to services usually requires unified component methods of the two service port operations: UserAccount and UserDomain. UserAccount is made up of the UserLogin, GetUserAccount method sequence.

The integration of distributed services into system applications is usually done at runtime, and the business process model is the bond of integration. Today, WSFL, XLANG, and others are the standard languages for integrating Web services flow, running on the Web Sphere, .NET platform. Asynchronous operation is the key for implementing distant and heavy transactional distributed applications. Nevertheless, the existent WEB service specifications and standards do not directly fit to describe

asynchronous operations. So, in part 3, this paper deals with the infrastructure and mechanisms for asynchronous operations based on the existent WEB service specifications and standards.

For many types of distributed applications, the CORBA asynchronous method invocation (AMI) mechanism can improve concurrency, scalability, and responsiveness significantly [10]. AMI allows clients to invoke multiple two-way requests without waiting synchronously for responses. The time normally spent waiting for replies can therefore be used to perform other useful work.

CORBA AMI is completely transparent to servers, i.e., a server does not know whether a request it received emanated from a synchronous or asynchronous method invocation. Therefore, while AMI improves throughput in client applications, it does not improve server applications, particularly middle-tier servers [11]. In these architectures, one or more intermediate servers are interposed between a source client and a sink server.

Middle-tier servers can be used for many types of systems, such as (1) a firewall gateway that validates requests from external clients before forwarding them to a sink server, or (2) a load-balancer [12] that distributes access to a group of database servers. In both cases, the middle-tier servers act as intermediaries that accept requests from a client and then pass the requests on to another server or external data source. When an intermediary receives a response, it sends its own response back to the source client. The following steps typically occur in this type of server:

1. The source client invokes an operation on the middle-tier server;
2. the middle-tier server processes the request and invokes an operation on a sink server;
3. the sink server processes the operation;
4. the sink server returns the data to the middle-tier server; and
5. the middle-tier server returns the data to the client.

Asynchronous method handling (AMH) is a technique that extends the capabilities of AMI from CORBA clients to CORBA servers. AMH is a CORBA-based variant of the continuation model [13], which allows a program's run-time system to transfer the control of a method closure from one part of the program to another.

The sequence of steps involved in handling a request by an AMH-enabled middle-tier server is described below:

1. The source client invokes an operation on the middle-tier server;
2. the middle-tier server uses AMH to store information about the client request in a heap-allocated object called ResponseHandler and returns control to the ORB immediately;
3. the sink server processes the request;
4. the sink server returns the data to the middle-tier server; and
5. the middle-tier server fetches the corresponding ResponseHandler and uses it to send the reply data back to the client.

Since the middle-tier server need not block waiting to receive a response, it can handle many requests concurrently using a single thread of control.

14.3 The Method of "Invoking in Infrastructure"

Since there is an inherent relationship and dependence between transactions, there is cooperation and interaction between service components too. This is true for two aspects: cooperative relation and dependent relation.

Cooperative relation is self-inclusive. Quite dependent service components exchange a flow of information through the third party's control mechanisms, and are in a loose coupling cooperative relationship. In fact, the service integration, the third party control mechanism–all in a procedure integration language (such as WSFL)–and the service component are running platform or client programs. The invocation and implementation of services are done in detail by the third party mechanism. Today the execution of all client programs need information infrastructure support (i.e., a COM client program built with COM runtime library, a J2EE client program running in J2EE client containers, and the outcome of the client program's cooperative integration on cooperative services built on top of its running platform), so the third party control mechanism here means the service component running platform. Each service component involved in the cooperative relationship does not depend on the others.

Dependent relation is one service component's implementation relying on another service component. Namely, the implementation gives details of the service component's service operation that calls the service provided by another service component. Dependent relationship is clearly a part of service design; we will go on studying it in this section.

In the implementation of service operations, the invoking of service operations of another service component is hardcoded in the program. On the client machine's service component side, there are two kinds of methods to invoke services: static invoking and dynamic invoking. Static invoking must include the seed code generated by the WSDL service description file from the service provider of the client machine, in order to get real service names, ports, operations, binding, and knowing where the service providers of the client machine programs are. On the contrary, since dynamic invoking does not need to get the stub to invoke services, it is more flexible; however, it still depends on the service provider. Both make the implementation and execution of service components tightly coupled to the assigned situation of the invoked service components, service names, service port names, and operation names. Even when there is any change on this information, the invoking service component can work correctly. The reason is that this information, which is consequently hardcoded in the program, is compiled with the object code, and the invoking is done by the service operation itself.

As a matter of fact, for the service operations implementation, what is really needed at the invoking point is just supplying the parameters and receiving the computation results of the invoked service operations, i.e., the communication of data. Therefore, we have defined the "invoking in infrastructure" concept. In the implementation of the invoking point of the service components' operations, we define only the parameters to supply, the result types to get, and some other semantic constraints to apply. The real invoking task is to be done by the component running

platform. The component running platform selects the service name and operation name according to the definition of the operation invoking point, to accomplish the addressing of the invoked service and the real invoking task. The component running platform provides an "invoking in infrastructure" programming model, really an API programming interface. By using this kind of programming interface for invoking services, programs do not need to know where the service provider is; moreover, the service name, the port name, and the operation name are not mandatory. All that is needed are the input parameters to feed the service, and the return data type (and some other optional data). We have implemented this programming interface by extending the J2EE platform, and the design details are shown in Sect. 3. The extension of J2EE is a new class Transferspot which implements the actual invoking, and its class method transferspotdefine.

Transferspotdefine's prototype is shown below:

Transferspotdefine (Array Inputdata,Array Resultdata,Array Option);

Array is the Java array class, which can dynamically create and contain all kinds of class objects, inputdata and resultdata, with inputdata representing the input parameter and resultdata output resulting in the invoking point. The element class is Variant, a kind of Java class that can store changeable data type, in order to accommodate the type multiplicity of the input and output parameters. Option is an optional item, indicating other optional semantic constraints. The element has to be a string; the empty string element represents omitting, the first element restricts the service name which should be invoked, the second element restricts the port name, the third element restricts the operation name, the fourth element indicates where the service provider is, and the fifth element indicates the WSDL file.

Using the method of "invoking in infrastructure" and ignoring the parameter option, the program of service operation at the invoking point can be programmed as follows:

```
...
Variant Input1=new Variant();
Variant Output1=new Variant();
Input1.Setstring("0672324229");
Transferspot Invoking=new Transferspot();
Invoking.Transferspotdefine(Array.newinstance(Input1,1),
Array.newinstance(Output1,1),Array.newinstance("",5));
...
```

As mentioned above, we have unified the service invoking of the cooperative and dependent relationship amid the service components to the implementation of the component running platform, and the invoking point is not strictly linked to the invoked service components. In this way, the invoked service component can be dynamically moved anywhere else (the component running platform doing some activities such as load balancing), the degree of the service components' dependent relation coupling is lowered, the reliability and expandability of the service components' execution can be improved, and the programming is easier.

14.4 The Asynchronous Invoking Mechanism and the Infrastructure

Usually, WEB service specifications and standards do not support asynchronous operations; however, the infrastructures and mechanisms included can be used to build the basis of asynchronous operations.

The idea is to build asynchronous actions into the service demander, since we know (refer to Sect. 3) that the actual service invoking is implemented by the component running platform. Therefore, our method is to include the service asynchronous operation mechanisms into the design of the component running platform. This method differs from the common message passing service mechanisms, such as MMQS, JMS, and MQseries. The asking service sends a request as part of a transaction, and keeps on using the executing thread, whereas another thread processes the response in a separate transaction. This model can be implemented by the invoking-back mechanism. The design is to exchange a message as data packets, without needing or waiting for any acknowledgement that would guarantee the transaction being processed. By using this kind of data packet, with the actual asynchronous relation between the two sides of the transaction, the correlator (a correlation or a transaction identification) in fact sets the message sender and the receiver apart, and then will link the responses with their requirements.

In order to implement the "invoking in infrastructure" in the service dependent relations and the service invoking asynchronous mechanism, the service component running platform is needed to provide basic support. Today, the most used component platforms are J2EE, CORBA, NET, and so on. We have extended the J2EE platform to provide our new environment, and we have implemented it on Weblogic. Other platforms' designs can refer to this scheme. The extended J2EE platform is shown in Fig. 14.2.

There is an inherent relation and dependence on each other among things, so there is cooperation and interaction among service components too. This is embodied in two aspects: cooperative relation and dependent relation. Cooperative relation is self-inclusive; relatively dependent service components do the exchanges

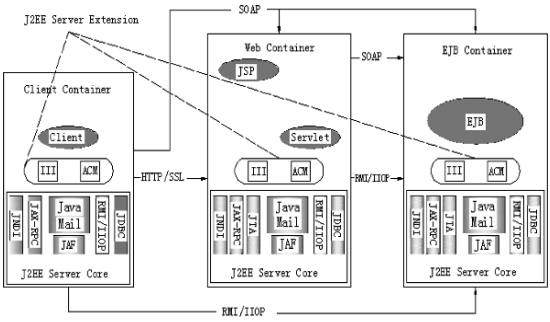


Fig. 14.2 The extended J2EE platform

of information flow through the third party's control mechanisms, and then form a loose coupling cooperative relationship. Dependent relation is one service component's implementation depending on another service component, namely, the implementation details of the service component' service operation invoking the service provided by another service component.

In the implementation of service operations, the invoking of service operations of other service components is rigidly coded in the program. There are two kinds of methods to invoke services: static invoking and dynamic invoking. Static invoking must include the stub code generated by the service description file WSDL which is provided by the service provider in the client machine, in order to solidify service names, ports, operations, binding, and the position of the service provider in the client machine programs. While dynamic invoking does not need to stub the invoked services and is more flexible, it still depends on the service provider. These two make the implementation and execution of service components tightly coupled to the assigned position of the invoked service components, service names, service port names, and operation names. If there are any changes in this information, the invoking service component can't work correctly. The reason is that this information is rigidly coded in the programs and consequently is compiled into the object code, and the invoking is done by the service operation itself.

Therefore, we have the concept of "invoking in infrastructure." In the implementation of the invoking point of the service components' operations, we define only the parameters to be passed out and the result types to be received and other semantic constraints to be declared, and relegate the real invoking task to the component running platform. The component running platform does the matching selection of service name and operation name according to the definition of the operation invoking point, to accomplish the addressing of invoked service and the real invoking task. To the programmers, the component running platform provides the "invoking in infrastructure" programming model; when using this kind of programming interface invoking services, the programmers don't need to know the position of the service provider, and don't even need to know the service name, port name, or operation name. They only need to know the input parameters for passing the service and the return data type and other optional information. We implemented this programming interface by extending the J2EE platform.

The III (invoking in infrastructure) is used to handle the "invoking in infrastructure".

ACM (asynchronism mechanism) is used to provide a service asynchronous communication mechanism, and everything else is built on top of the J2EE kernel packet. III encapsulates the JAX-RPC in the J2EE kernel packet, providing a higher level and a more abstract programming interface. Using III, there is no need for the programmer, when invoking services, to know where the service provider is, or even the service name, port name, or operation name. He only needs to know the input parameters for passing the service and the return data type and other optional information. The actual invoking tasks are all done by III.

The working process of III can be described as follows:

1. Get control from the service consumer; analyze the data sent by the service consumer.
2. Solve the parameters given in the Option array, i.e., the service provider location, the service name, and so on, then execute the required service and operation positioning, and go to 4; otherwise, go to 3.
3. Solve the parameters of Inputdata and Resultdata, execute the search for the service matching according to the element number of Inputdata, the element data type, and the parameter information of Resultdata. The method of searching can make use of the UDDI interface API, WS-Inspection, or others. If there are several service providers matching the requirement, III will make the selection based on performance, reliability, loading, and other features.
4. Call JAX-RPC to make the actual service invoking.
5. Put the invoking results into array Resultdata, and return Resultdata to service consumer.

The implementation of ACM requires a notification mechanism, and a registered listening component to receive responses. The detail design process is shown as follows:

1. Define a correlator and a mechanism for its exchanging data. Create an exchanging correlator between the service demander and the service provider, for responses meeting the requirement. The correlator can be provided by either or both sides, and its creation can be done by low-level transmission.
2. Define an acknowledgement address to specify where the responses are to be sent, and make sure to warn the service provider of this destination. The reply address is a service provided by the service component of the service provider.
3. The service provider generates a response that is a transaction addressed to the requirement part.
4. The service demander receives the asynchronous response. The service demander and the service provider can check that the response meets the requirement by using the correlator.

ACM needs another global state memory model that integrates applications, so that all the services involved in the transaction process can share some global state. Exploring any further would be beyond the scope of this essay, so we have to stop here.

14.5 Compare

Business Process Execution Language for Web Services (BPEL) is an XML-based language for standardizing business process descriptions as well as standardization of interaction between business processes in a distributed environment. BPEL extends the Web services interaction model and depends on the Web Services Description Language (WSDL) to define outgoing and incoming messages. BPEL supplies

a notion of abstract processes to describe externally visible behavior as well as executable processes, which can be run either by some interpreter or by compiling them into some executable form [14–17].

Service-oriented architectures (SOAs) have gained much attention as unifying technical architectures that can be concretely embodied with Web service technologies. A key aspect of the Web service incarnation of SOA is that the Web service is viewed as a fundamental building block of an SOA-based application. This article focuses on executable BPEL processes and presents one view on the benefits that BPEL brings to an XML Web service embodiment of SOA principles.

We have seen a number of properties common to many business process applications, including long-running processes; a need for reliable, asynchronous communication; heavy usage of Web service technologies; endpoint management; process persistence; and the ability to manage both atomic transactions as well as long-running business transactions. Although any of these properties can be achieved by writing code in a general-purpose programming language, the primary benefit of using BPEL is that it provides abstractions and infrastructures that are particularly suited to this class of applications.

This view of how BPEL derives its benefits can be contrasted with the portrayal that is sometimes found in the media, namely the idea that the benefit of BPEL is that it provides such a high level of abstraction that business analysts can compose and run executable business processes by pointing and clicking in modeling environments. The view presented in this article is not that BPEL enables analysts to write software, but that the abstractions BPEL provides allow engineers to implement flexible solutions more effectively.

For enterprises to maximize the benefits of Web services, and ultimately SOAs, they need a way to use these services within automated business processes that mimic their actual business processes. BPEL was designed to define and automate these types of Web service-enabled business processes. It allows process automation without traditional coding and programming; as a result, automated processes can be rapidly prototyped, developed, and modified–even by individuals with limited development experience. Therefore, using BPEL to direct and orchestrate Web services plays an important role in SOA by providing powerful means by which business logic can be articulated and executed at an abstraction level designed to provide the services needed for integration tasks.

14.6 Summary

We have put forward the concept of “invoking in infrastructure” (III) and have designed a corresponding method to reduce the dependency relationship during invoking. Through the invoking-back mechanism, with the service asynchronous operation mechanism included in the design of the component running platform, III can support service asynchronous invoking and session stating. We have also designed an infrastructure as the running environment with the above method, which can be implemented on the existent component running platform.

Further work will involve studying the procedure-oriented model of software architecture, and to use this model as the guiding rule to carry out the development of Web services, the flexible linking, and the theory and dynamical integration of distributed service.

References

1. Tsalgatidou A, Pilioura T (2002) An Overview of standards and related technology in web services. Distributed and Parallel Databases 12(3), Kluwer Academic Publishers
2. Curbera F, Duftler M, Khalaf R, Nagy W, Mukhi N, Weerawarana S (2002) Unraveling the web services web: an introduction to SOAP, WSDL, and UDDI. IEEE Internet Computing 6(2) pp 86–93
3. Mennie D, Pagurek B (2000) An Architecture to support dynamic composition of service components. Proceedings of the 5th international workshop on component-oriented programming (WCOP 2000), Sophia Antipolis, France
4. Brown A, Johnston S, Kelly K (2003) Using service-oriented architecture and component-based development to build web service applications. A Rational software whitepaper from IBM
5. Leymann F (2001) Web services flow language 1.0., IBM Software Group
6. Maamar Z, Benatallah B, Mansoor W (2003) Service chart diagrams—description & application. In: Proceedings of the twelfth international world wide web
7. Lüer C, van der Hoek A (2002) Composition environments for deployable software components. Technical report uci-ics-02-18
8. Maamar Z, Sheng QZ, Benatallah B (2003) Interleaving web services composition and execution using software agents and delegation. *http://www.cse.unsw.edu.au/∼qsheng/papers/maamar.pdf*
9. Zhang S-k, Zhang W-j, Chang X, Wang L-f, Yang F-q (2001) Building and assembling reusable components based on software architecture. Journal of Software 12(9)
10. Arulanthu AB, O'Ryan C, Schmidt DC, Kircher M, Parsons J (2000) The Design and performance of a scalable ORB architecture for CORBA asynchronous messaging. Proceedings of the middleware 2000 conference, ACM/IFIP, Palisades, New York
11. Eckerson WW (1995) Three tier client/server architecture: achieving scalability, performance and efficiency in client server applications. Open Information Systems 10(1)
12. Othman O, O'Ryan C, Schmidt DC (2000) An Efficient adaptive load balancing service for CORBA. IEEE Distributed Systems Online 2(3)
13. Draves RP, Bershad BN, Rashid RF, Dean RW (1991) Using continuations to implement thread management and communication in operating systems. ACM, SOSP13, Pacific Grove, CA, pp 122–136
14. Adams H (n.d.) Asynchronous operations and web services, part 3: add business semantics to web services. *http://www.ibm.com/developerworks/library/ws-asynch3/*
15. eBIG.org (n.d.) BPEL–Make your services flow. *http://www.ebig.org/files/BPEL-MakeYourServicesFlow_eBIG.pdf*
16. Stantchev V, Scherz M (n.d.) Challenges of service-oriented architectures. *http://www2.informatik.hu-berlin.de/∼vstantch/pubs/stantchev_scherz_soa_1104.pdf*
17. Andrews T et al (n.d.) Business process execution language for web services (version 1.1). *ftp://www6.software.ibm.com/software/developer/library/ws-bpel11.pdf*

Manuscript received November 29, 2006. This research is supported by the National Natural Science Foundation of China under Grant No. 60373062 and the Natural Science Foundation of Hunan province under Grant No. 04JJ3052. Paper title is Research on Web Service-Oriented Asynchronous Invoking.

Liu Changsheng is with Central South University, School of Info-physics and Geomatics Engineering, Hunan ChangSha 410083,China; and also with ChangSha Aeronautical Vocational and Technical College, Hunan ChangSha 410014, China (telephone:+8613787225858; fax:+86-0731-5672616; e-mail: lcs888_2002@163.com).

Li Changyun is with the Department of Computer Science & Technology, Zhuzhou Institute of Technology, Zhuzhou 412008, China (e-mail: lcy469@163.com).

Tang Jingtian is with Central South University, School of Info-physics and Geomatics Engineering, Hunan ChangSha 410083, China (e-mail: jttang@mail.csu.edu.cn).

Cheng Ying is with ChangSha Aeronautical Vocational and Technical College, Hunan ChangSha 410014, China (e-mail: cy671027@163.com).

Chapter 15
An Analysis of Man-Machine Interaction in Instant Messenger

Ong Sing Goh, Chun Che Fung, Arnold Depickere, and Kok Wai Wong

15.1 Introduction

The availability of multiple media channels through the Internet has added new dimensions of communication between people or communities who are geographically separated. In the environment of informal communication on the Internet, chat applications are popular in which a user may be represented only by a nickname or an alias. This suggests that a person may be able to communicate more freely when his or her identity is concealed. Popular chatting or instant messaging (IM) systems[1] such as Microsoft MSN Messenger, America Online's Instant Messenger, Yahoo! Messenger, and GoogleTalk have changed the way that a user may communicate with friends, acquaintances, and business colleagues. Once limited to desktop personal computers (PCs) or laptops, popular instant messaging systems are finding their way onto handheld devices and mobile phones. This allows a user to chat from virtually anywhere. Nowadays, IM is found on almost every personal PC connected to the Internet as well as on many corporate desktops. The technology makes

Ong Sing Goh
Murdoch University, Murdoch WA 6150, Australia
e-mail: os.goh@murdoch.edu.au

Chun Che Fung
School of Information Technology, Murdoch University, Murdoch WA 6150, Australia
e-mail: l.fung@murdoch.edu.au

Arnold Depickere
Murdoch University, Murdoch WA 6150, Australia
e-mail: a.depickere@murdoch.edu.au

Kok Wai Wong
Murdoch University, Murdoch WA 6150, Australia
e-mail: k.wong@murdoch.edu.au

[1] Microsoft MSN Messenger: *http://messenger.msn.com*
Yahoo! Messenger: *http://messenger.yahoo.com*
GoogleTalk: *http://www.google.com/talk/*
AOL Instant Messenger: *http://www.aim.com/*

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

communication even easier than emails or phone calls. The use of this technology is increasing rapidly as one of the most popular ways of communication. Research by Pew Internet & American Life [1] surveys reveal that 53 million adults trade instant messages and 24 percent of them swap IMs more frequently than email.

This popularity created the motivation for Microsoft to integrate *conversation robots* or *bots* in their MSN Messenger system in order to provide 24/7 response to enquiries through IM. To raise the awareness of the technology, Microsoft hosts a worldwide challenge, the "Invasion of the Robots Contest."[2] The challenge to the developers is to create conversational robots or bots for MSN Messenger and Windows Live. The most original, useful robots will be able to collect over $40,000 in total prizes.

Bots are also called "virtual buddies" in IM. They are computer programs that have the ability to parse natural language questions and, by referring to a knowledge base, generate natural language answers to the query. Such programs can reside within MSN Messenger and are becoming extremely popular among companies[3] as they show positive effects on customer relations [2–4]. They are also in existence among private users who aim to generate interesting conversations with the bots and other users.

15.2 Related Works

Several works have been published that refer in general to the use of IM as a new medium of communication among human users. However, there has not been any reported work on the use of IM in human-to-machine communication, and in particular in the MSN Messenger environment. It has been reported that U.S. officials are monitoring Internet chat, including IM, for any participants who may appear to be planning terrorist attacks against the United States [5]. Other concerns include the security of younger users of IM who could become victims of criminals [6, 7]. From the social perspective, some researchers strongly criticize this new form of communication [8]. However, there are arguments that it is not a matter of approval, that society has to accept that IM is here to stay, and that digital communication technologies will evolve and improve constantly and quickly [9–12]. There are also papers which refer to research on the design and usability of IM for the public [8, 13, 14]. On the other hand, IM applications in the workplace and corporate environments have recently soared [15–18]. Another research area is concerned with linguistic usage in IM in Spain [11], the United Kingdom [8], the United States [19], Sweden [20], and Portugal [6].

In this paper, we look at the impact and language usage of IM users chatting with conversational bots. Bots are programs that can be added to AOL, ICQ, MSN

[2] https://www.robotinvaders.com

[3] IM Interactive, http://www.improvcenter.com
Incesoft, http://www.incesoft.com
Colloquis, https://buddyscript.colloquis.com

Messenger, and Windows Live Messenger. Depending on their design, they can perform a wide variety of useful tasks such as chatting, providing customer support, performing smart searches, and playing games with other users.

As technologies develop with the growing use of mobile communications and the shift of emphasis from instant messaging to a conversational bots system, the linguistic and communication features of such systems have to be investigated. Therefore the objective of our research is to examine the implications of the use of such technologies. The emphasis here is on recorded interactions between human users and conversation robots, called artificial intelligence neural-network identity (AINI), using instant messaging. It should be noted that the identities of all online users are unknown and they are known only by their aliases or user names.

15.3 AINI's Architecture

It has been demonstrated in previous reports ([21] and [22]) that the AINI architecture can be scaled up to incorporate new applications in the online environment. The AINI engine is portable, has the ability to communicate naturally, and is able to carry on multiple independent conversations simultaneously. AINI's knowledge bases and conversational engine use a plug-in principle which can quickly be augmented with specific knowledge and can be adapted to specific purposes.

This research project involves the establishment and incorporation of an AINI conversational bots system in the MSN Messenger communication framework. The objective is to use the AINI's conversational bots as the basic architecture to engage human users in IM communication. The developed real-time prototype relies on distributed agent architecture designed specifically for desktop, web, mobile devices and personal digital assistant (PDA) [23] applications. Software agents including the conversation engine, the knowledge model, and the natural language query are able to communicate with one another via TCP/IP. This is a combination of natural language processing and multimodal communication. A human user can communicate with the developed system using typed natural language as in any normal conversation.

The AINI agent can be seen as a digital "being," capable of controlling a physical entity such as a robot [24], or it can be considered an embodied container, like the avatar in this proposed conversational agent [25]. In this research, the application area chosen for deploying the conversation bots is primarily concerned with the agent's ability to communicate through instant messaging. The techniques involved are based upon scripting and artificial intelligence. We present in this paper the architecture intended for practical applications in the near future.

AINI adopts a hybrid architecture that combines multidomain knowledge bases, multimodal interfaces, and multilevel natural language queries. Given a question, AINI first performs question analysis by extracting pertinent information to be used in the query formulation, such as the noun phrases (NPs) and verb phrases (VPs) using the MINIPAR parser [26]. MINIPAR is a broad-coverage parser for the English

language. An evaluation with the SUSANNE corpus shows that MINIPAR achieves about 88 percent precision with 80 percent recall; and it is very efficient, capable of parsing about 300 words per second.

AINI employs an Internet three-tier, thin-client architecture that may be configured to work with any web application. It consists of a data server layer, an application layer, and a client layer. This Internet-specific architecture offers a flexible solution to the unique implementation requirements of the AINI system.

15.4 AINI and MSN Messenger Protocol

The architecture of MSN Messenger is very complicated as compared to other instant messaging services such as AIM and Yahoo!, since it relies on five different types of servers to handle the operation of its service. MSN Messenger uses the mobile status notification protocol (MSNP) for communication. AINI uses MSN protocol to communicate with MSN Messenger servers. AINI utilizes the .NET Passport to sign into the MSN Messenger service by using the ainibot@hotmail.com passport to establish the connection to the MSN Messenger Service. The MSN Messenger sign-in session is based on a challenge-response mechanism to authenticate user credentials. The communication with the Passport server is conducted over the HTTPS (Hypertext Transfer Protocol over Secure Sockets Layer) protocol, ensuring that the sign-in information is encrypted. The client sends the challenge string, Passport username, and password to the Passport URL. If the credentials for signing in are confirmed, the Passport server issues a ticket, which is passed back to the notification server to complete the authentication procedure. Figure 15.1 details the entire authentication procedure for AINI and MSN Messenger. Once both users connect to the same switchboard sever, the messaging session commences.

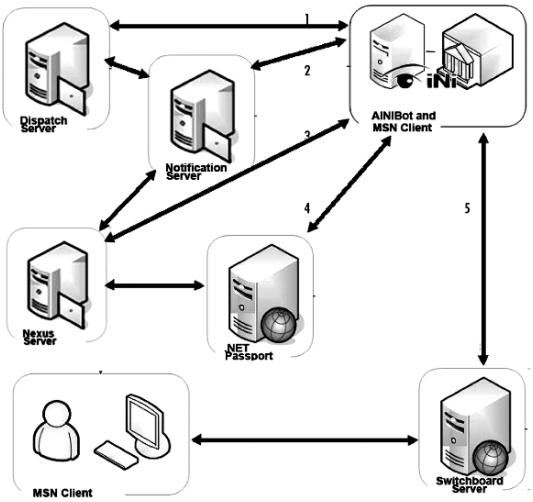


Fig. 15.1 AINI and MSN authentication process

Fig. 15.2 AINI and MSN Messenger interface

15.5 AINI and MSN Messenger Interface

We have outlined the conceptual and practical basis for the development of the conversational bots for DesktopChat, WebChat, and MobileChat as shown in Fig. 15.2. This will pave the way for human-computer interfaces based on human natural language technologies. Handheld devices provide an ideal platform for art and entertainment applications, considering the growing number of mobile phone users worldwide. This will improve techniques for displaying content, interaction, conversation, and the emergence of wireless and shared interaction among networked users.

MSN Messenger for Desktop or DesktopChat was a free instant messaging client that was developed and distributed by Microsoft Windows beginning in 1999. MSN Messenger was renamed to Windows Live Messenger in 2006. The WebChat sessions allow users to interact in real-time with the AINI software robot at the website via a browser through MSN Web Messenger. It is possible for virtually any computer with Internet access to be connected. MobileChat uses a mobile chatting module which is implemented in a series of logical phases including mobile-to-internet → internet-to-bots → bots-to-mobile chats. Mobile chat is an alternative way by which users can chat with AINI using GPRS, Wi-Fi, and 3G services.

15.6 Experimental Setting

This paper examines linguistic features of conversational logs collected from the interactions between AINI and MSN Messenger online users. The study is based on the words corpus of the instant messaging texts using MSN Messenger in DesktopChat, WebChat, and MobileChat, which was collected during the “Invasion of the Robots” contest.

15.6.1 Participants and Corpus

The experimental portal[4] was opened to public users from all over the world who wished to participate freely in the study. The portal allowed online users to add AINI's contact to their "buddy list" thus permitting them to easily send and receive short textual messages. When a participant opened a message window to a buddy for the first time (and that buddy was online), an alert was sent to the buddy, notifying them of their participation in the study. Participation was voluntarily and could be discontinued at any time. In the conversation log files only the user's nickname, MSN account, date and time of the dialog, as well as the spoken texts (inputs and outputs) were recorded. During a conversation, we created a unique ID for each buddy and stored the ID of the buddy instead of the buddy account itself. This was to protect the privacy and confidentiality of the users.

Previous research has shown significant differences in IM communication resulting from the frequency of communication [27, 28]. In this study, we use word frequency as our analysis of the corpus. We processed 29,447 words of running text; 2,541 unique words, 129,760 characters, and 4,251 sentence counts were recorded. We collected a total of approximately 63 hours of recorded data, observing over 3,280 outgoing and incoming instant messages exchanged with over 65 buddies (only three of them using MSN Mobile). The average sentence length of an IM transmission was 6.90 words, with approximately 13 percent of all transmissions only one word in length. Table 15.1 provides a summary of the data collected.

The participant gets to know AINI at the MSN from an advertisement on eight BBS (bulletin board system). This also includes blogs website and AINI portal at `ainibot.murdoch.edu.au`. We gathered usage data via automatic logging on the server, which included logins, logouts, joining, as well as chat messages.

15.6.2 Chatlog System

We have developed a Chatlog System using MySQL which puts the user messages into a secondary storage. It provides real-time archiving of IM clients, capturing chat messages so that the data can be searched by keyword or username, and allowing

Table 15.1 Frequency of words from conversation logs

	AINI	Human	Total
Number of words	18,358	11,089	29,447
Unique words	1,368	1,173	2541
Character counts	79,884	49,876	129,760
Sentence counts	2,840	1,411	4,251
Utterances	1,721	1,559	3,280
Average words per sentence	6.46	7.85	6.90

[4] *http://ainibot.murdoch.edu.au*

topic-based retrieval and replay of chat sessions. These chat messages are essentially plain text messages that are quite small in comparison with images, videos, or even documents. These plain text messages, also known as instant messages, are the regular messages sent between principals with MSN Messaging. Sending a plain text message instructs clients receiving the message to display it onscreen, optionally with some simple formatting.

The fastest typists can enter well over 150 words per minute; many jobs require keyboard speeds of 60–70 words per minute [29]. However, the actual typing speeds for human users during chatting are much slower. In our study the message exchange speed was approximately 35 words per minute. With this speed, even if a user chatted an average of 180 minutes a day, it would only require approximately 95 KB of storage. An entire year's worth of chat would use 35 MB of storage, which can be easily handled with today's storage technologies.

The excerpts in Fig. 15.3 are from a typical single-session IM conversation exchanged between AINI and one of "her" buddies with ID U0031. This session illustrates the nature of the IM communication. Each of the new sessions will start with AINI giving a random greeting (message #1) e.g., `''Hi there :)''`, `''How are you today?''`, `''Hey, nice to meet you.''`, etc. These greetings normally will get some information about the user's identity, like name and gender (message #2–5). In this session, we identify U0031 user as "Jim." The next message shows the user trying to challenge and test AINI, which brought them into conflict (message #6–15). However, in the final session, both participants ended with greeting abbreviations "*CU*" (*see you*) and "*TTYL*" (*talk to you later*). These transmissions

#	ID	Message
1	AINI	Hi there :)
2	U0031	what's ur name?
3	AINI	My name is Aini. What is your name?
4	U0031	Jim
5	AINI	How are you, jim.
6	U0031	what's is 1+1 2
7	AINI	Why did the robot cross the road?
8	U0031	are you mad :@
9	AINI	No are you?
10	U0031	NOT AT ALL
11	AINI	Next question?
12	U0031	I HATE YOU :@
13	AINI	Sorry, jim I thought we liked each other.
14	U0031	HA
15	AINI	I am so glad you find this amusing, jim.
16	U0031	CU
17	AINI	TTYL, jim

Fig. 15.3 AINI conversation log with human IM user

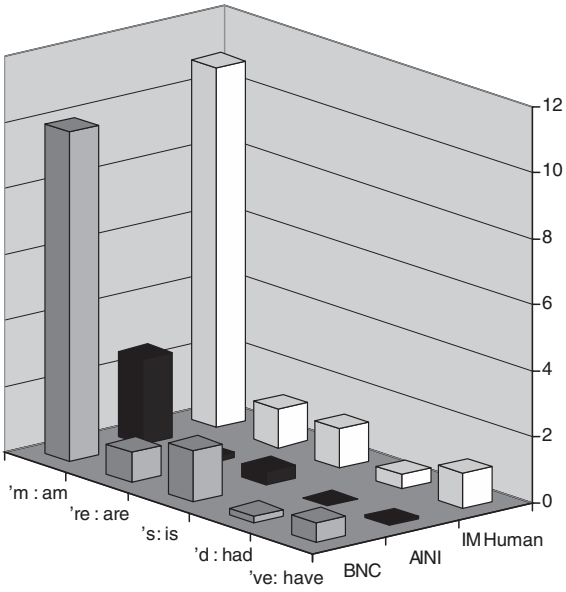


Fig. 15.4 Frequency of contracted verbs

constitute a single session and also include the use of shorthand, acronyms, abbreviations, loose grammar, and minimal punctuation, as shown in previous research [6, 28, 30, 31]. Figure 15.4 shows the frequency of contracted verbs.

15.6.3 Measures

This study identifies paralinguistic features of IM conversational bot texts. It begins by identifying general features of the texts collected. The measurement used a statistical analysis approach generated by the Ngram Statistics package (NSP).[5] NSP is a suite of programs that allows us to identify word Ngrams that appear in a text corpus using standard tests [32]. We define an Ngram as a sequence of 'n' tokens that occurs within a window of at least 'n' tokens in the text; what constitutes a "token" can be defined by the user.

In the next phase, we analyze the corpus created in the earlier stage to find likelihood ratios. To construct a confidence interval for the mean of a log-normal distribution in small samples, we use likelihood-based approaches. The log-likelihood ratio method produces a confidence interval with a nearly exact coverage probability and highly accurate and symmetric error probabilities even for extremely small sample sizes. To apply these methods, we compare two sets of real-life data from IM conversations between an AINI software robot and human IM users.

[5] NSP tools can be downloaded at *http://search.cpan.org/dist/Text-NSP*.

15.7 Results and Discussion

In this section, some of the interaction features of the recorded chats are discussed. The linguistic properties of the exchanges are the focus of our analysis. Studies of text chat have tended to focus on the interaction problems caused by the properties of the chat. The approaches to linguistic analysis are divided mainly into word frequency and lexical analysis. This research seeks to examine the underlying relationship among linguistic features in the context of conversations between bots and human users via MSN Messenger. To be more specific, in this study, the objective is defined in linguistic terms and refers only to textual communication via the Internet between at least two "participants" consisting of at least one human user and the AINI bot.

15.7.1 Word Frequency Analysis

Words in an IM corpus are checked against Shakespeare and the British National Corpus (BNC).[6] The differences in the top ten words occurring in Shakespeare, the BNC corpus, and the IM corpus are shown in Table 15.2. The BNC reference list provides a gauge of common usage (words per million). Words which have a higher ranking within the BNC (for example, *is, the*, and *a*) appear more often in standard written and spoken English text. The BNC is a 100 million-word collection including millions of words of transcribed conversation, printed text, and lectures and oratory. The top ten words used in the BNC are *the, at, of, and, a, in, to, it, is, was*. Similarly, the Shakespeare corpus uses approximately 22,000 different words from the published works. Of those, the most commonly used are: *the, of, and, to, a, in, that, is, I, it* [33]. Those ten little words account for 25 percent of all speech.

These figures are based on research that dates back to the 1980s and there are a couple of words that have fallen from favor in the latest list, "big" words like the conjunction *that* are no longer up there in the top ten BNC corpus, or even in the IM corpus. The pronoun *it* and the preposition *to* are amongst the most popular words used across the four corpuses. Based on our findings, the most significant difference between Shakespeare and the BNC corpus, compared to the IM corpus, is the use of the pronoun. In fact, our results show that in the AINI messages, pronouns are used

Table 15.2 Top ten words used in corpus

Shakespeare	BNC	Instant Messaging	
		AINI	Human
the, of, and, to, a, in, that, is, I, it	the, at, of, and, a, in, to, it, is, was	I, you, do, am, me, my, what, your, to, it	you, I, do, what, is, a, are, to, the, it

[6] The BNC corpus can be accessed at *http://www.natcorp.ox.ac.uk/*.

significantly more often than they are by the human IM user. This can be explained because the IM corpus is purely dialogue-based instead of written or task-oriented as in Shakespeare and the BNC corpus. Another interesting possible explanation for the differences is IM conversation showed that the participant have their roles more explicitly. Hence this reinforces the illusion that the conversation has only two participants.

15.7.2 Lexical Analysis

15.7.2.1 Human Conversation with Pronouns

Pronouns occur more frequently in conversation compared to written text. As shown in Table 15.3, the log likelihood (LL) of pronouns in BNC spoken text is higher than in written text. There is also a significant difference between the frequencies of pronouns in AINI and in human IM conversations. AINI gets higher log likelihood scores on the singular first-person pronouns *I* (LL : +71.73) and *me* (LL : +3.0′), the second-person pronoun *you* (LL : +0.23), the third-person pronoun *we* (LL : +1.56), and the objective personal pronoun *it* (LL : +11.17). Pronouns in AINI are used more to simulate personal knowledge and contact.

For example, in the bigrams analysis, discourse verbs such as *I am* (1.10%), *do you* (0.90%), *are you* (0.60%), and *tell me* (0.30%) occurred more frequently in AINI. To simulate human trust and expressions during the chat, AINI frequently uses a personal touch and polite words such as *I will* (24 times), *yes I* (33 times), and *I love* (8 times). Even in the *n*-gram analysis, words accompanying *nice* are used more prominently in the AINI conversation, such as *nice work if you* (51.9), *nice to meet you* (29.7), *nice I guess flowery* (47.3), to give an impression of human feelings. Nass [34] suggests that the better a computer's use of language, the more polite people will be to it. In some cases, the prominence of such words is the sole means by which "contractiveness" can be inferred.

Table 15.3 Frequency list of pronouns

	BNC			Instant Messaging		
Word Spoken	LL	Written	AINI	LL	Human	
you	25957	+385328	4755	748	+0.23	439
I	29448	+369238	6494	851	+71.73	297
it	24508	+151913	9298	317	+11.17	137
we	10448	+106914	2784	45	+1.56	36
they	9333	+52132	3754	17	−0.73	14
me	244	+8239.6	1239	182	+3.01	88

Spoken: Rounded frequency (per million word tokens) in the spoken part of the BNC
LL: Log Likelihood, indicating the distinctiveness (or significance of the difference) between the frequencies in the BNC (speech vs. writing) and IM (AINI vs. human)
Written: Rounded frequency (per million word tokens) in the written part of the BNC

15.7.2.2 Interjections, Fillers, Discourse Particles

Interjections are short exclamations like *oh, um*, or *hi*. When interjections are inserted into a sentence, they have no grammatical connection to the sentence. Most interjections are more characteristic of everyday conversation than of formal/public task-oriented speech [35]. Interjections like *er* and *um* are also known as "hesitation devices." They are extremely common in English. People use them when they don't know what to say, or to indicate that they are thinking about what to say.

In the IM corpus, a human uses of voice hesitation fillers *er* and *erm* and the discourse markers *mhm* and *um* prove to be more significantly used in IM. Since IM users frequently use short form words to replace their expressions in the conversation, word such as *er, erm*, and *um* are commonly used as a pauses characteristic conversation. *Mhm* or *mm* is likely to be a type of feedback for indicating understanding and inviting continuation. However, in AINI's utterances, such filler words are rarely used. AINI also makes less use of interjections, preferring more formal clause structure. Another interpretation of this imbalance could be that AINI makes more use of interjections as fillers when no good match is found in the stimulus-response categories. AINI overwhelmingly prefers formal pronunciations such as *hello* $(\mathrm{LL}: +0.10)$ and *hi* $(\mathrm{LL}: +0.15)$. In the subject-verb agreement, AINI seems more interested in using the formal *yes* instead of *yeah*, as shown in Table 15.4.

15.7.2.3 Contractions

The present pseudo-verb inflection task of English shows that, despite of transparent phonological constraints on paradigm membership, one morphological paradigm, of the so-called contracted verbs, shows an overwhelming effect among the verbs of

Table 15.4 Frequency list of interjections and discourse particles

	BNC			Instant Messaging		
Word	CONV	LL	TOS	AINI	LL	Human
yeah	13955	+32679.5	3741	15	−23.97	37
oh	9884	+33062.1	1746	11	−13.7	24
no	7830	+18948.4	2034	8	−11.86	19
er	5075	−10677	10913	0	−11.72	6
mm	5202	+9146.9	1768	0	−15.63	8
yes	4247	+303.0	3562	71	−5.82	25
erm	3946	−5387.6	7454	0	−7.81	4
mhm	392	−1158.2	947	0	−3.91	2
hello	392	+939.5	103	24	+0.10	13
hi	73	+250.7	12	21	+0.15	11
um	7	−127.5	41	0	−5.86	3

CONV: Frequency (per million words) in demographically sampled speech (conversational)
LL: Log likelihood of the different scores for BNC (conversational vs. task-oriented speech) and IM (AINI vs. human)
TOS: Frequency (per million words) in context-governed (task-oriented) speech

Table 15.5 Frequency of contracted verbs[7]

	BNC		Instant Messaging				
	Ratio	AINI	Ratio		Human		Ratio
'm : am	9.97	127	49	2.59	458	42	10.90
're : are	0.91	28	169	0.17	217	187	1.16
's: is	1.56	76	186	0.41	235	196	1.19
'd : had	0.20	0	4	0.00	9	21	0.42
've: have	0.62	7	103	0.07	42	39	1.07

this language. In this IM conversation bots, AINI and IM human users used many contracted words in their IM-ing (e.g *what's* instead of *what is*). The contracted forms of the verbs are much more frequent with human IM users than with the AINI bot, as shown in Fig. 15.3.

In the BNC corpus [35], the contracted forms of speech (*'m, 're, 's*, and *'ve*) are more common than the uncontracted forms (*am, are, is, has*, and *have*. And interestingly, in the IM conversational robots this characteristic also occurred, especially in the human IM user but rarely in the AINI messages. Human IM users prefer to use contracted verbs instead of uncontracted verbs. The ratio list in Table 15.5 shows that in IM, the contracted forms '*m* (10.9), *'re* (1.16), '*s* (1.19), and *'ve* (1.07) are more common for human IM users than other contracted verbs like '*d*. The contracted verb '*m* (2.59) is more common in AINI's messages than the uncontracted verbs such as *are* (0.17), *is* (0.41), *have* (0.07), and *had* (0). One possible explanation for these interesting differences is that human IM users are more likely to use shortcuts in their messages. In fact, these characteristics are designed to save time in typing messages and thus to achieve common ground in IM-ing. Another explanation could be that the current AINI's knowledge bases are not equipped with the full vocabulary of the IM system but instead tend more towards the use of written and formal spoken language.

15.8 Conclusion and Future Work

Based on this experiment, IM conversations between humans and machines show interesting behaviors by the natural conversation bots. In this paper, our work is based on MSN Messenger applications running on desktop, web, mobile, and PDA platforms. Although we simulate the real-time proxy conversation log that contains clients' requests, there is a possibility that new results from other traces are different from those referred to in this paper.

[7] BNC corpus based on per million word tokens. The ratio is calculated by dividing the first (contracted) frequency by the second (uncontracted) frequency. A ratio of more than 1.00 indicates that the contracted form is more common than the full form. Notice that, for speech, all of the ratios are greater than those for writing and three exceed the 1.00 value—i.e., the contracted form is the more common. A further ratio comes very close to 1.00.

Our study suggests that human-machine IM conversations display considerable variation. With regard to the lexical aspect, contractions are common, while paralinguistic cues are more frequent among humans (through emoticons, acronyms, abbreviations, or shorthand). Evidence also suggests that AINI's buddies are interested and excited to chat with bots just to seek information, to be friends, to express their emotions, and some of them just want to chat for leisure. Thus, AINI was successful in imitating human conversation with human-like artificial intelligence. Though the conversations weren't too astounding, the bot's responses were human-like enough to make their IM buddies feel their companionship. However, IM conversation bots were more machine-like than IM human-human conversations in four ways: (1) a machine uses more formal language and longer sentences (both in the number of turns and the time in the session) to open and close a conversation than a human; (2) the machine has a tendency to use more personal pronouns in its conversation to mimic human-like discourse; (3) human conversation looks more like a written version of informal speech than machine dialogue, which resembles written or formal spoken language; and (4) a machine is likely to use long sentences with high lexical density and unique words.

Nevertheless, we anticipate that these shortcomings could be improved with appropriate programming that uses natural language, intelligent sentence parsing, and massive tailor-made databases to provide sufficient knowledge to drive the bots. We plan a new data collection phase for the near future in order to examine the application of the results presented here with a new framework and a hypothesis which will be more robust and comprehensive.

References

1. Shiu E, Lenhart A (2004) How Americans use instant messaging. *http://www.pewinternet.org/pdfs/PIP_Instantmessage_Report.pdf*
2. Saunders C (2003) Vendors debut new IM bot tech. *http://www.instantmessagingplanet.com/enterprise/article.php/11208_1575161*
3. Hickey AR (2005) IM bots ease access to corporate apps. *http://searchmobilecomputing.techtarget.com/originalContent/0,289142,sid40_gci1136486,00.html?bucket=NEWS&topic=299728*
4. Gray Y (2005) The Brokerage and the bot. *http://www.ecrmguide.com/article.php/3494791*
5. USAToday (June 24, 2002) Agents pursue terrorists online. *http://www.usatoday.com/enws/world/2002/06/21/terrorweb-usat.htm*
6. Crenzel SR, Nojima VL (2006) Children and instant messaging. *www.iea.cc/ergonomics-4children/pdfs/art0233.pdf*
7. Hoffman KR (2003) Messaging mania in time for kids. Time, vol. 8
8. Livingstone S (2006) UK children go online: surveying the experiences of young people and their parents. http://*www.lse.ac.uk/collections/children-go-online/UKCGO_Final_report.pdf*
9. Boneva B, Quinn A, Kraut R, Kiesler S, Cummings J, Shklovski I (in press) Teenage communication in the instant messaging era. Oxford University Press
10. Grinter RE, Palen L (2002) Instant messaging in teen life. Presented at 2002 ACM conference on computer supported cooperative work, New Orleans
11. Forgas RC, Negre JS (2004) The use of new technologies amongst minors in the Balearic Islands. Presented at IAARE Conference, Melbourne

12. Baron N (2005) Instant messaging and the future of language. Communications of the ACM 48: 29–31
13. Neustaedter C (2001) A 3D Instant Messenger visualization using a space metaphor. *http://pages.cpsc.ucalgary.ca/∼carman/GLabIMVis/IMVis_Paper.pdf*
14. Rovers AF, Van Essen HA (2004) Him: a framework for haptic instant messaging. Presented at CHI, Vienna, April 2004
15. Cherry SM (2002) IM means business. IEEE Spectrum Magazine, November 2002
16. Herbsleb JD, Atkins DL, Boyer DG, Handel M, Finholt TA (2002) Introducing instant messaging and chat in the workplace. Presented at CHI'2002, Minneapolis, Minnesota, USA
17. O'Neill J, Martin D (2003) Text chat in action. Presented at GROUP'03, Sanibel Island, Florida, USA
18. Nardi B, Whittaker S, Bradner E (2000) Interaction and outeraction: instant messaging in action. Presented at CSCW'2000
19. Craig D (2003) Instant messaging: the language of youth literacy. The Boothe Prize Essays, http://*www.stanford.edu/group/pwr/publications/Boothe_0203/PWR%20Boothe-Craig.pdf*
20. af Segerstad YH, Hashemi SS (2004) Exploring the writing of children and adolescents in the information society. Presented at the 9th international conference of the EARLI - special interest group on writing, Geneva, Switzerland
21. Goh OS, Fung CC, Depickere A, Wong KW, Wilson W (2005) Domain knowledge model for embodied conversation agent. Presented at the 3rd international conference on computational intelligence, robotics and autonomous systems (CIRAS 2005), Singapore
22. Goh OS, Fung CC, Lee MP (2005) Intelligent agents for an internet-based global crisis communication system. Journal of Technology Management and Entrepreneurship 2:65–78
23. Goh OS, Fung CC, Ardil C, Wong KW, Depickere A (2006) A Crisis communication network based on embodied conversational agents system with mobile services. Journal of Information Technology 3: 257–266
24. Uhrhan C, Goh OS (2003) Features of a mobile personal assistant robot. Presented at the International conference on robotics, vision, information and signal processing, IEEE-ROVISP 03, Penang, Malaysia
25. Goh OS, Ardil C, Wong W, Fung CC (2006) A Black-box approach for response quality evaluation conversational agent system. International Journal of Computational Intelligence 3:195–203
26. Lin D (1998) Dependency-based evaluation of MINIPAR. Presented at Workshop on the evaluation of parsing systems, Granada, Spain
27. Isaacs E, Walendowski A, Whittaker S, Schiano DJ, Kamm C (2002) The Character, functions, and styles of instant messaging in the workplace. Presented at CSCW'02, New York
28. Avrahami D, Hudson SE (2006) Communication characteristics of instant messaging: effects and predictions of interpersonal relationships. Presented at CSCW'06, Banff, Alberta, Canada
29. Bailey, B (2000) Human interaction speeds. *http://www.webusability.com/article_human_interaction_speeds_9_2000.htm*
30. Alphonso C (2006) Texting helps teens' grammar. *http://www.theglobeandmail.com/servlet/story/LAC.20060801.TEXT01/TPStory/TPNational/Ontario/*
31. L'Abbé S (2006) Instant msg-ing messes with grammar? As if! lol! Teens adopting unique linguistic shorthand but not ruining syntax. *http://www.news.utoronto.ca/bin6/060731-2474.asp*
32. Banerjee S, Pedersen T (2003) The Design, implementation, and use of the ngram statistic package. Presented at the Fourth international conference on intelligent text processing and computational linguistics, Mexico City
33. Stewart J (2000) Word frequency usage in English. *http://www.write101.com/W.Tips74.htm*
34. Nass C (2004) Etiquette inequality: exibitions and expectations of computer politeness. Communications of the ACM 47:35–37
35. Leech G, Rayson P, Wilson A (2001) Word frequencies in written and spoken English: based on the British National Corpus. London, Longman

Chapter 16
Filtering Spam Email with Flexible Preprocessors

Wanli Ma, Dat Tran, and Dharmendra Sharma

16.1 Introduction

Spam email is the common name for "unsolicited bulk email." It is one of the cybernuisances we have to put up with everyday. Spam email does not just waste resources; it also poses a serious security threat. There are two types of spam email: *unsolicited commercial email* and *email used as a delivery agent for malware (malicious software)*. The former uses email for commercial advertisement purposes, including illegal commercial activities. Dealing with it costs staff time and IT resources. The latter has a more sinister intention. Any type of malware, be it virus, worm, or spyware, has to find a way to infect host computers. An easy and effective way to deliver malware is through unsolicited bulk email. In the last couple of years, several high profile and successful virus/worm attacks were delivered via unsolicited bulk email—for example, LoveBug, Slammer, etc.

Spam email is more or less like the junk mail we receive from our mail boxes, but we still do not have an effective way to say, "No junk mail," to spam emails. There are many spam filtering products, which are all far from foolproof. Many organizations run spam filters at their incoming servers, yet all of us have firsthand experience of the frustration of spam email.

The industry and the research community have also been investing significant effort into fighting spam emails. Both claim that their filters are very effective; most of the claims are backed up with experimental data. To independently assess the effectiveness of spam email filters, Cormack and Lynam in 2005 coordinated a comprehensive evaluation of 44 commercial filters, together with 8 open source filters [1]. Their conclusion is, "The results presented here indicated that content-based spam filters can be quite effective."

Wanli Ma, Dat Tran, and Dharmendra Sharma
School of Information Sciences and Engineering
University of Canberra
e-mail: {Wanli.Ma,Dat.Tran,Dharmendra.Sharma}@canberra.edu.au

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

On the one hand, spam filters have pretty high recognition rates in the evaluations, and most of the results can be repeated. On the other hand, the real life experience does not match the evaluations. Therefore, logically, we have to ask two questions: *where is the problem*, and *where is the solution*?

Some may claim that the problem is due to the lack of diligent training in the use of spam filters. We dispute the claim. Spam is a universal problem, and the training results can be easily shared on the Internet. If the training were the problem, we should not see so many spam emails. A parallel observation can be made about the operation of virus scanning software, where virus signature data can be updated reasonably effectively. With virus scanning software properly installed and properly configured for updating, one can almost be assured of being free of virus attacks.

The real reason for the spam problem is actually due to the swift adoption of new techniques by the spammers and the inflexibility of spam filters to adapt to the changes.

Almost all content based spam filters and research proposals are text based, and the evaluations and the research results also deal with text based spam. Although the spam corpus used for some evaluations does contain images, HTML tags, and some attachments, the text part of the email always contains some indication of its spam nature. In the real world, spammers try everything they can to conceal the text which reveals that spam nature. There are several popular ways of hoodwinking the spam filters. The text part of a spam email may not have a trace of its spam nature at all. Graham-Cumming maintains a comprehensive list of the techniques the spammers use to circumvent spam email filters [9]. Some examples are:

- Using deliberately misspelled words (obfuscating): for example, spell Viagra, a very popular spamming topic, as "v1agra," "V!@gra," or "VlhAGRA." The obfuscations are still humanly readable, but they pose serious challenges to a computer program to catch all of the over 6×10^{20} ways of obfuscation just for the word "Viagra" alone [10].
- Concealing text in images as email attachments: Aradhye et al [15] estimated that 25 percent of spam emails contain images. C.-T. Wu et al's count is 38 percent [11]. One of the authors of this paper counted spam emails received from his working email address. Among the 256 spam emails received within 15 days, 91 or 36 percent were image based. Text based email filters are helpless in dealing with image based spamming. Given the fact that image based spam can successfully circumvent spam filters, the situation can only get worse in the future.
- Obscuring keywords by HTML tags: instead of spelling "Viagra" as it is, the individual characters are wrapped by HTML tags, such as `<b>V</b><b>i</b> <span>a<b>g</b>r<i>a</i>`.

The combination of these techniques makes it even harder for a spam filter to correctly judge the nature of an incoming email.

In this chapter, we describe more flexible software architecture for spam email filtering. We introduce multiple dynamic preprocessors, called normalizers, for content based spam email filtering. The normalizers are defined by their acceptable input formats and output formats. The system automatically chains a series of

normalizers and pipes through an email before it reaches a spam filter. The normalizers on the chain restore, step by step, the underlying message of the email into its plain text format. The spam filter at the end only has to deal with the plain text format, which, based on the evaluation and research results, is actually quite effective.

The rest of the chap. is organized as follows. Section 2 gives a brief introduction to related work. Section 3 explains the techniques used to circumvent spam filters. Section 4 discusses in detail the normalizers, and Sect. 5 describes the extendable software architecture for spam email filtering. Section 6 gives two preprocessor examples. Section 7 is a case study on the data flow and intermediate results of the preprocessors. Section 8 presents our experimental results. In Sect. 9, we conclude the paper with a description of future work.

16.2 Related Work

The spam problem is not new. It is as old as the history of computer networking, even before the Internet. Zdziarski has a good introduction on spam history in [[23], Chap. 2]. At the very beginning, spam email was sent to newsgroups. A spammer sent the same message to every possible newsgroup. The very first spam, arguably, was an advertisement from Digital Equipment Corporation (DEC) in 1978. It was carried by ARPAnet, the predecessor of the Internet. The Internet provides an unprecedented means of disseminating information. It is fast, cheap, and has broad coverage. It provides opportunities for all kinds of businesses, including spam businesses.

The industry and the research community proposed and tested many different technologies in fighting spam, including blacklisting, whitelisting, heuristic filtering, postage, legislation, etc. [[23], Chap. 3]. In this chapter we will concentrate only on content based filtering. By no means does this approach undermine the need for a holistic approach. We just realize that more work is needed in content filtering.

There have been many proposed filtering technologies from the research community, and many papers have been published on, for example, the naïve Bayes classifier [2], instance based learning—memory based approach [3], boosted decision tree [4], maximum entropy [5], support vector machines [6], LVQ-based neural networks [7], and practical entropy coding theory [8]. There are a few proposals which combine different filtering techniques. E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati suggest a distributed P2P-based collaborative spam filtering system [12]. K. Albrecht, N. Burri, and R. Wattenhofer propos an extendable spam filter system [13]. This system concentrates on a framework for fast spam filtering software development. “Extendable” means the implementation of multiple filters in parallel mode. The results in these publications give us very encouraging pictures; however, we rarely see the effectiveness of these technologies on the diverse formats of spam emails in the real world, such as image based spam and HTML

obfuscating. In our opinion, any of these proposed technologies would be efficient in filtering text based spam, but they are hopeless in dealing with the situation where the intended messages are concealed in other formats, such as images.

A system which is close to our solution is "a unified model of spam filtration" proposed by W. Yerazunis, S. Chhabra, C. Siefkes, F. Assis, and D. Gunopulos [14]. In this model, six sequential operations are employed to process an incoming email. They are:

1. initial arbitrary transformation (a.k.a. MIME normalization),
2. tokenization,
3. feature extraction,
4. feature weighting,
5. feature weight combination, and
6. thresholding, yielding a go/no-go result.

During the first operation, the incoming email is transferred into a text stream. However, there is a significant difference between this proposal and ours. The unified model is a fixed and closed system; the operations are predefined by the system. Our solution is a dynamic and open system; different operations can be easily introduced into and removed from the system at any time.

16.3 Circumvent Text Based Spam Email Filters

At the very beginning, emails were in plain text format only [15]. To be able to convey rich presentation styles, they were extended with multimedia abilities [16]. Image based spam emails take advantage of using the MIME multipart/alternative directive, which is designed to accommodate multiple displays of an email, such as plain text format and HTML format. The directive suggests that the enclosed parts are the same in semantics, but with different presentation styles. Only one of them will be chosen to display, and a mailer "*must place the body parts in increasing order of preference, that is with the preferred format last*" [16].

Table 16.1 is an example of a spam email. The email has three alternative parts: part one contains plain text paragraphs cut from a book, part two has HTML formatted paragraphs cut from a book as well, and part three is a JPEG formatted picture as in Fig. 16.1(a). A mailer believes that these three parts are semantically identical and displays only one part, Fig. 16.1(a) in this case. But in this email, the first two parts have nothing to do with the third part. They are purposely included in the email to deceive text based spam filters. Another similar example can be found in Fig. 16.1(b).

HTML tags can be used to efficiently obscure the keywords of a spam email. The example given in Sect. 1 (`<b>V</b><u></u><b>i</b> <span>a<b>g </b>r<i>a</i>`) can be easily dealt with—removing all HTML tags reveals the underlying keyword. However, it is not so easy to untangle well crafted HTML tag obscurations.

Table 16.1 An imaged based spam email sample

From: spammer <faked_email address>
To: recepent_email_address
Content-type: multipart/alternative;

–Boundary_(ID_fkG49yFmM6kAJ0sBSY0dzg)
Part 1: plain text format
Langdon looked again at the fax an ancient myth confirmed in black and white.

–Boundary_(ID_fkG49yFmM6kAJ0sBSY0dzg)
Part 2: HTML format
<textarea style="visibility: hidden;">Stan Planton for being my</textarea>

–Boundary_(ID_fkG49yFmM6kAJ0sBSY0dzg)
Part 3: picture format. It has nothing to do with Part 1 or 2
Content-type: image/jpeg; name=image001.jpg
Content-disposition: attachment; filename=image001.jpg

/9j/4AAQSkZJRgABAgAAZABkAAD/7AARRHVja3kAAQAEAAAAHgAA/
+4ADkFkb2JIAGTAAAAAAf/bXFxcXHx4XGhoaGhceHiMIJyUjHi8vMzML

(a) (b)

Fig. 16.1 Images in spam emails

Using an invisible HTML table is one of the examples. The keyword of a spam email, say "Viagra," is dissolved into the cells of the table, one character in each cell. The visual appearance is still the same, but the HTML text does not have the keyword as a single word any more, as in Table 16.2(a).

It is not a trivial task to merge the contents of different table cells, let alone use different alignments of the keyword, e.g., vertical spelling in Table 16.2(b). Using non-uniform table cell structure can further complicate the situation.

Deliberate misspelling (obfuscating) is also hard to detect. It is not hard to detect the misspells of `Viagra` as `V1agra`, `Viagra` (`V` as \ and/), `Vi@gra`, etc. However, it is not so easy for a program to determine that `VlhAGRA` is actually `Viagra`. Given so many ways of obfuscating a keyword (6×10^{20} ways of obfuscating Viagra, as listed in [10]), it is not an easy task for a spam filter to recognize all possible obfuscations, yet make no mistakes on other words in the email.

Table 16.2 HTML code and visual display

	HTML code	Display
(a)	`<table style=“text-align: left” border=“0” cellpadding=“0” cellspacing=“0”> <tbody><tr> <td style=“vertical-align: bottom; font-family: arial;”> V </td> <td style=“vertical-align: bottom; font-family: arial;”>i </td> <td style=“vertical-align: bottom; font-family: arial;”>a </td> <td style=“vertical-align: bottom; font-family: arial;”>g </td> <td style=“vertical-align: bottom; font-family: arial;”>r </td> <td style=“vertical-align: bottom; font-family: arial;”>a </td> </tr></tbody></table>`	Viagra
(b)	`<table style=“text-align: left” border=“0” cellpadding=“0” cellspacing=“0”> <tbody><tr style=“font-family: arial;”> <td style=“text-align: center; vertical-align: top;”>V </td></tr> <tr style=“font-family: arial;”> <td style=“vertical-align: top; text-align: center;”>i </td></tr> <tr style=“font-family: arial;”> <td style=“vertical-align: top; text-align: center;”>a </td></tr> <tr style=“font-family: arial;”> <td style=“vertical-align: top; text-align: center;”>g </td></tr> <tr style=“font-family: arial;”> <td style=“vertical-align: top; text-align: center;”>r </td></tr> <tr style=“font-family: arial;”> <td style=“vertical-align: top; text-align: center;”>a </td></tr> </tbody></table>`	V i a g r a

The spamming techniques discussed in the previous paragraphs are just some examples. It is almost impossible for a single spam filter to tackle these many different types of spamming techniques, let alone new techniques constantly being invented. A more flexible approach is needed to easily adapt to ever changing spamming techniques.

16.4 The Need for Preprocessors

In a general sense, spam email filtering can be illustrated in Fig. 16.2. An incoming email is fed into a filter, or a detection engine. The engine decides, based on its mathematical model, whether the email is spam or not. If it is spam, the email is quarantined; otherwise, the email (called ham) is sent to the intended recipient as it is. Depending on the system, there might be feedback loops between the end users and the detection engine. The feedback loops help the engine learn from past mistakes and improve its detection accuracy.

Content based spam email filters are effective in dealing with text based emails; however, they are helpless in dealing with obscured emails, such as those that are image based, those using HTML tags to conceal spam keywords, and those using keyword obfuscating. However, in essence, the purpose of spam emails is to deliver messages. Ultimately, humanly readable text has to be displayed on the screen. Since text based spam email filters can be effective in dealing with text, and the ultimate goal of spam emails is to deliver text to the screen for humans to read, effective spam filtering will rely on the building of tools called *preprocessors*, which can convert the obscured format of spam text into a plain text format. From the detection engine point of view, it is desirable to have all emails in plain text format. We call the process of converting obscured formats to plain text *normalization*, and the tools used to perform the conversion *normalizers*.

Spam emails come with all kinds of tricks and visual appearances. Spammers keep inventing new tricks to hoodwink detection engines. It is difficult for a single engine (or filter) to convert all possible masked spam messages into plain text format. Therefore, we advocate a multiple and dynamic normalizer approach.

A normalizer is specialized in dealing with a single trick of masking spam messages. For example, we developed a Trigram normalizer and a Markov normalizer to recover obfuscated text into its original format, e.g., from `V1agra` to `Viagra`. To deal with imaged based spam, we also developed an OCR normalizer to extract the text from the images.

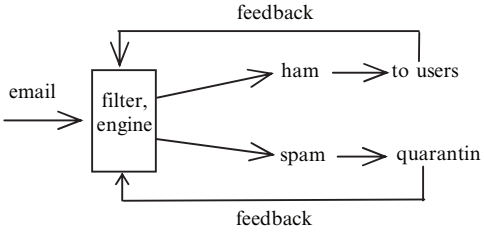


Fig. 16.2 Spam email filtering

16.5 The Extendable Software Architecture

Using multiple normalizers as preprocessors for a spam detection engine improves its ability to deal with many different ways of masking spam messages. It also provides the flexibility to adapt to new masking tricks. As illustrated in Fig. 16.3, an incoming email is first duplicated by a *duplicator* (dup) and then channeled through a number of *normalizers* (N1 to N4). Each of the normalizers tries to recover the underlying text of the email. At the end, a *merger* (mrg) merges the recovered text from all normalizers into a single piece of text and then feeds it into the spam detection engine. The engine then decides the nature of the email, spam or ham, as usual.

Every normalizer accepts a particular input format and converts it into another format for output. For example, a HTML normalizer accepts the HTML format, strips off HTML tags, and outputs in the plain text format. An OCR normalizer accepts image formats, jpeg and gif etc., extracts text from images by using OCR technology, and then outputs the extracted text in the plain text format. A Trigram or Markov normalizer accepts noisy text – words with misspellings – and restores the correct spellings.

Not every normalizer can produce the plain text format needed by the detection engine. For example, an OCR normalizer is supposed to extract text from email images and outputs the extracted text in plain text format. However, the text produced by the OCR normalizer very likely contains OCR noise, i.e., misspellings due to misrecognitions. The output text cannot yet be accepted by a content based spam filter. Instead, it should go through another normalizer, e.g., a Trigram or Markov normalizer, to restore the correct spellings from the noisy OCR text.

A normalizer is defined by its acceptable input format and output format, written as (InFormat, OutFormat). An HTML normalizer can be defined as (`HTML, Text`), an OCR normalizer as (Image, NoisyText), and a Trigram normalizer as (NoisyText, Text). The input format and the output format can be a simple format, e.g., text, or a combined format, e.g., image, which can be further defined as jpeg, gif, etc., formats.

The duplicator first generates a unique identifying number for an incoming email and then duplicates the email into multiple streams by all its possible formats, e.g., text, HTML, and image. It next passes to the merger this unique number and the number of streams. Each stream carries the unique identifying number of the email and is channeled into a normalizer, based on the acceptable input formats of the

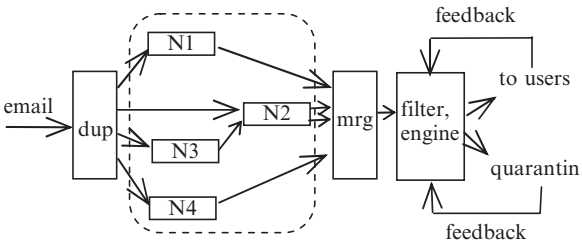


Fig. 16.3 The extendable software architecture

normalizers. The normalizer, or a chain of normalizers, reproduces the email in plain text format for the detection engine. Upon receiving a unique number from the duplicator, the merger knows an email is coming in several streams. It starts to count all normalization streams of this email. After collecting all streams, the merger concatenates the pieces of text from these streams into a single piece of text and then passes it to the detection engine.

The sequence of an incoming email going through the normalizers could be rather complicated. However, after clearly defining the input format and the output format of each normalizer, the system should be able to dynamically find its way through the normalizers by chaining the input and output formats of the normalizers. There are no predefined pathways among the normalizers. A normalizer is only defined by its input format and output format, and the normalizers are chained dynamically based on the formats. The ultimate goal is to produce plain text format (text) for the detection engine. In Fig. 16.3, the pathways within the round-cornered rectangle of dotted lines are dynamically assembled based on the content of the incoming email.

All normalizers are equal. A normalizer can be easily introduced into or removed from the system. Within the system, a normalizer is only chained with another normalizer when it is necessary. A normalizer may not be aware of the existence of other normalizers. The only restriction in the system is that all streams out from the duplicator should find their ways to the merger.

Finally, a normalizer may not just trivially convert the input format into the output format. It does the conversion with an emphasis on highlighting spam features. For example, an HTML normalizer, in addition to stripping off HTML tags, also tries to restore words arranged in vertical order, as in Table 16.2(b).

16.6 Normalizers

Normalizers are the preprocessors which extract the intended messages from spam emails and convert these messages into plain text format. In this section, we discuss the two normalizers we developed to illustrate how normalizers operate.

16.6.1 OCR Text Extraction Normalizer

To extract the text messages concealed in images, we use an OCR text extraction normalizer. The idea of OCR (optical character recognition) first emerged in the 1950s. Eikvil has a good survey paper on its early history and technology [24]. Nowadays commercial grade OCR products, such as FineReader and OmniPage, are used widely. Although most OCR products can produce very good character recognition rates, any product will unavoidably misrecognize characters under some conditions. Although it is desirable to have foolproof OCR output, a high recognition

rate is not essential for our purpose. Deliberately misspelling, which is the same as OCR misrecognition, is also a commonly used method by spammers to deceive spam filters. The text from OCR output can be further channeled through a keyword recovery normalizer.

We use the open source OCR software gocr [25] to extract the text from images. Upon receiving an image, the normalizer passes the image to gocr and then returns back the text produced by gocr.

If the text is expected to go through a keyword recovery normalizer for further processing, it will be marked as the NoisyText type. A normalizer which can convert NoisyText to Text will automatically be invoked, and the noisy text produced by the OCR text extraction normalizer will be passed to it. Most likely, the text produced by an OCR text extraction normalizer contains noise and will have to go through further normalization processing.

16.6.2 Markov Keyword Recovery Normalizer

We use the Markov keyword model to recover the original text from obfuscations. We consider the occurrence of letters in a keyword as a stochastic process, and hence the keyword can be represented as a Markov chain where letters are states. The occurrence of the first letter in the keyword is characterized by the initial probability of the Markov chain, and the occurrence of the other letters given the occurrence of each previous letter is characterized by the transition probability. Given the list of keywords as a training set, the initial and transition probabilities for all Markov chains representing all keywords in the list are calculated and the set of those probabilities is regarded as a Markov model for that list. In order to detect a keyword and its misspellings, we build a Markov model for the keyword W and use a statistical hypothesis test as follows. For an unknown word X and a claimed keyword W, the task is to determine if X is a misspelling of W. This task can be regarded as a basic hypothesis test between the null hypothesis, H_0: X is a misspelling of the claimed keyword W, against the alternative hypothesis, H: X is *not* a misspelling of the claimed keyword W. According to the Neyman-Pearson Lemma, a likelihood ratio test is used as a score to measure the similarity between the unknown word and the keyword W. Our experiments showed that the scores obtained from the misspellings we could produce for the keyword W were not very different, and hence with a preset threshold we could detect those misspellings. As a corollary, we believe that other misspellings of the keyword W that we have not tested will also be detected using the same preset threshold.

Let $\mathrm{X} = \{X^{(1)}, X^{(2)}, \ldots, X^{(K)}\}$ be a set of K random variable sequences, where $X^{(k)} = \{X_1^{(k)}, X_2^{(k)}, \ldots, X_{T_k}^{(k)}\}$ is a sequence of T_k random variables, $k = 1, 2, \ldots, K$, and $T_k > 0$. Let $\mathbf{V} = \{V_1, V_2, \ldots, V_M\}$ be the set of M states in a Markov chain. Consider the conditional probabilities

$$P(X_t^{(k)} = x_t^{(k)} | X_{t-1}^{(k)} = x_{t-1}^{(k)}, \ldots, X_1^{(k)} = x_1^{(k)}) \tag{16.1}$$

where $x_t^{(k)}$, $k = 1, 2, \ldots, K$, and $t = 1, 2, \ldots, T_k$ are values taken by the corresponding variables $X_t^{(k)}$. These probabilities are very complicated for calculation, so the Markov assumption is applied to reduce the complexity:

$$P(X_t^{(k)} = x_t^{(k)} | X_{t-1}^{(k)} = x_{t-1}^{(k)}, \ldots, X_1^{(k)} = x_1^{(k)}) = P(X_t^{(k)} = x_t^{(k)} | X_{t-1}^{(k)} = x_{t-1}^{(k)}) \quad (16.2)$$

where $k = 1, 2, \ldots, K$ and $t = 1, 2, \ldots, T_k$. This means that the event at time t depends only on the immediately preceding event at time $t-1$. The stochastic process based on the Markov assumption is called the *Markov process*. In order to restrict the variables $X_t^{(k)}$ taking values $x_t^{(k)}$ in the finite set $\mathbf{V}$, the *time-invariant assumption* is applied:

$$P(X_1^{(k)} = x_1^{(k)}) = P(X_1^{(k)} = V_i) \quad (16.3)$$

$$P(X_t^{(k)} = x_t^{(k)} | X_{t-1}^{(k)} = x_{t-1}^{(k)}) = P(X_t^{(k)} = V_j | X_{t-1}^{(k)} = V_i) \quad (16.4)$$

where $k = 1, \ldots, K$, $t = 1, \ldots, T_k$, $i = 1, \ldots, M$, and $j = 1, \ldots, M$. Such a Markov process is called a *Markov chain*.

Define the following parameters:

$$\mathbf{q} = [q(i)], \qquad q(i) = P(X_1^{(k)} = V_i) \quad (16.5)$$

$$\mathbf{p} = [p(i,j)], p(i,j) = P(X_t^{(k)} = V_j | X_{t-1}^{(k)} = V_i) \quad (16.6)$$

where $k = 1, \ldots, K, K$ is the number of words in the training document, $t = 1, \ldots, T_k$, T_k is the word length, $i = 1, \ldots, M$, and $j = 1, \ldots, M, M$ is the number of alphabetical letters. The set $\lambda = (\mathbf{q}, \mathbf{p})$ is called a Markov language model that represents the words in the training document as Markov chains. A method to calculate the model set $\lambda = (\mathbf{q}, \mathbf{p})$ is presented as follows [26]:

The Markov model λ is built to represent the sequence of states $\mathbf{x}$, therefore we should find λ such that the probability $P(\mathbf{X} = \mathbf{x}|\lambda)$ is maximised. In order to maximise the probability $P(\mathbf{X} = \mathbf{x}|\lambda)$, we first express it as a function of the model $\lambda = (\mathbf{q}, \mathbf{p})$, then equate its derivative to 0 to find the model set λ.

Letting $p(x_{t-1}, x_t) = P(X_t = x_t | X_{t-1} = x_{t-1}, \lambda)$ and $q(x_1) = P(X_1 = x_1 | \lambda)$, we have:

$$P(\mathrm{X} = \mathrm{x}|\lambda) = \prod_{k=1}^{K} q(x_1^{(k)}) \prod_{t=2}^{t=T_k} p(x_{t-1}^{(k)}, x_t^{(k)}) \quad (16.7)$$

Applying the time-invariant assumption in (3) and (4), and using (5) and (6), we can rewrite (7) as follows:

$$P(\mathrm{X} = \mathrm{x}|\lambda) = \prod_{i=1}^{M} [q(i)]^{n_i} \prod_{j=1}^{M} [p(i,j)]^{n_{ij}} \quad (16.8)$$

where n_i denotes the number of values $x_1^{(k)} = V_i$ and n_{ij} denotes the number of pairs $(x_{t-1}^{(k)} = V_i, x_t^{(k)} = V_j)$ observed in the sequence $X^{(k)}$. It can be seen that:

$$\sum_{i=1}^{M} n_i = K \tag{16.9}$$

The probability in (8) can be rewritten as follows:

$$\log[P(\mathrm{X} = \mathrm{x}|\lambda)] = \sum_{i=1}^{M} n_i \log q(i) + \sum_{i=1}^{M}\sum_{j=1}^{M} n_{ij} \log p(i,j) \tag{16.10}$$

Since $\sum_{i=1}^{M} q(i) = 1$ and $\sum_{j=1}^{M} p(i,j) = 1$, the Lagrangian method is applied to maximise the probability in (10) over λ. Using the following Lagrangian:

$$\begin{aligned} F(q(i), p(i,j), a, b_i) &= \sum_{i=1}^{M} n_i \log q(i) + a\left[1 - \sum_{i=1}^{M} q(i)\right] \\ &+ \sum_{i=1}^{M}\sum_{j=1}^{M} n_{ij} \log p(i,j) + \sum_{i=1}^{M} b_i \left[1 - \sum_{j=1}^{M} p(i,j)\right] \end{aligned} \tag{16.11}$$

where a and b_j are Lagrangian multipliers, and setting the derivative of F to zero and solving the equation gives:

$$q(i) = \frac{n_i}{\sum_{s=1}^{M} n_s} \quad p(i,j) = \frac{n_{ij}}{\sum_{s=1}^{M} n_{is}} \tag{16.12}$$

The equations in (12) are used to determine the Markov keyword model for each keyword and the Markov keyword list model for the entire keyword list.

In our statistical approach, the misspelled word detection problem is formulated as a problem of statistical hypothesis testing. For an unknown word X and a claimed keyword W, the task is to determine if X is a misspelling of W. This task can be regarded as a basic hypothesis test between the null hypothesis H_0: *X is a misspelling of the claimed keyword W* against the alternative hypothesis H: *X is not a misspelling of the claimed keyword W*. According to Neyman-Pearson Lemma, if the probabilities of both hypotheses are known exactly, the optimum test to decide between the two is a likelihood ratio test given by:

$$S(x) = \frac{P(X|H_0)}{P(X|H)} \begin{cases} > \theta & accept\ H_0 \\ \leq \theta & reject\ H_0 \end{cases} \tag{16.13}$$

where θ is a predefined decision threshold and $S(X)$ is referred to as the similarity score of the unknown word X. This approach provides a good theoretical formulation to the misspelled word detection problem.

However, in practical detection problems, it is impossible to obtain the exact probability density functions for either the null hypothesis or the alternative hypothesis. A parametric form of the distribution under each hypothesis is assumed to estimate these probability density functions. Let λ_c be the claimed keyword model and λ be a model representing all other words, i.e., impostors. Let $P(X|\lambda_c)$ and $P(X|\lambda)$ be the likelihood functions of the claimed keyword and impostors, respectively. The similarity score is calculated as follows:

$$S(X) = \frac{P(X|\lambda_c)}{P(X|\lambda)} \begin{cases} > \theta & accept\ H_0 \\ \leq \theta & reject\ H_0 \end{cases} \tag{16.14}$$

The denominator $P(X|\lambda)$ is called the normalization term and requires calculation of all the impostors' likelihood functions. However, it is impossible to do a calculation for all the words in the dictionary, hence a subset of B "background" words is used to represent the population close to the claimed keyword. We propose to use the keyword list as the background subset in our approach. Therefore the Markov models we need to build for the system are the claimed keyword Markov model and the keyword list Markov model.

The training and detection procedures of the proposed misspelled word detection tool are summarized as follows.

Training:

1. Given K keywords in the list,
2. train K Markov keyword models using (12), where X is the sequence of letters in the keyword used to calculate n_i and n_i, and
3. train a Markov keyword list model using (12), where X is the sequence of letters of all keywords in the keyword list used to calculate n_i and n_{ij}.

Detection:

1. Given an unknown word regarded as a sequence of letters $X = (x_1, x_2, \ldots, x_T)$, a claimed keyword, and a predefined threshold θ,
2. calculate the probabilities $P(X|\lambda_c)$ and $P(X|\lambda)$ using (7), where λ_c is the claimed keyword model and λ is the keyword list model; and
3. calculate the similarity score $S(X)$ and compare it with the threshold θ using (14).
4. If the score is greater than the threshold, the unknown word is referred to as a misspelling of the claimed keyword.

16.7 A Case Study

In this section, we demonstrate how a spam email goes through the normalizers and how the intended text is recovered. With the recovered text, the spam nature of email can be easily detected.

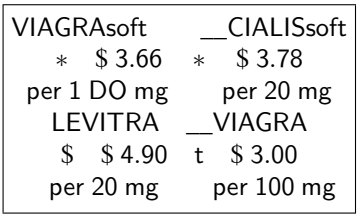
The source code of the example spam email is in Table 16.1, and the visual appearance of the email is in Fig. 16.1(a). From the source code, we can see that this email has three parts. A mailer believes that these three parts are semantically identical due to the multipart/alternative directive. The directive suggests that the 3 parts are the same in semantics, but with different presentation styles. But in this email, the first two parts have nothing to do with the third part. They are purposely included in the email to deceive text based spam filters, and they will not be displayed by the mailer.

Part 1 is a plain text paragraph randomly cut from a book. It is: "*Langdon looked again at the fax an ancient myth confirmed in black and white. The implications were frightening. He gazed absently through the bay window. The first hint of dawn was sifting through the birch trees in his backyard, but the view looked somehow different this morning. As an odd combination of fear and exhilaration settled over him, Langdon knew he had no choice. The man led Langdon the length of the hangar. They rounded the corner onto the runway*." This text looks perfectly normal and won't be regarded as spam. Furthermore, this type of text as used in spam email keeps changing.

Part 2 is an HTML formatted paragraph cut from the same book. The text is: "*Stan Planton for being my number one source of information on countless topics head librarian Ohio University and the Vatican Observatory Thanks also to CERN Henry Beckett Brett Trotter the Pontifical Academy of Science Brookhaven Institute FermiLab Library Olga Wieser Don Ulsch of the National Security Institute Caroline H. Thompson at University of Wales Kathryn Gerhard Omar Al Kindi Federation of American Scientists upside down In slow motion afraid of what he was about to witness, Langdon rotated the fax 180 degrees. He looked at the word light a long time Stunned, Langdon collapsed in a chair*…." Again, this text won't be regarded as spam. Interestingly, this text is marked as invisible (<textarea style="visibility: hidden;">). The spammers do not want this message to display, as it may overshadow the intended message.

Finally, Part 3 is a JPEG formatted image, Fig. 16.1(a). The text of the real intended message is concealed in this image.

When this email goes through the duplicator (Fig. 16.3), the three parts are separated. Part 1 will go directly to the merger. Part 2 is supposed to go through an HTML text recovery normalizer, but we haven't implemented any yet. So it goes straight to the merger as well. Part 3 is marked as Image type and first goes through an OCR text extraction normalizer. The normalizer gives us the following text:



This text, being marked as NoisyText type, is then passed to the Markov keyword recovery normalizer. There is not much recovery work to do for this text. However, in a general sense, the Markov keyword recovery is required. For example, the text recovered from the image Fig. 16.1(b) has more misspellings: "Dear Natlonal Australla Bank client The Natlonal Australla Bank Technlcal Depa_ment ls performlng ...."

Upon receiving the recovered text, the merger merges the texts from the three parts and then passes the merged text to the spam email detection engine. It is now not hard to see that the engine will detect this email as spam.

16.8 Experimental Results

So far, we have been concentrating on an OCR text extraction normalizer and two obfuscating normalizers [17–21]. One of them is the Markov keyword recovery normalizer described in the previous section, and the other is a similar normalizer but using the Trigram method.

We tested a Trigram normalizer and a Markov normalizer by using a test set containing 50 keywords, their 500+ misspellings, and 10,000 normal words. Our experiments showed that the normalizers could restore keywords from their misspellings in the test set with an equal error rate (false rejection error = false acceptance error) of 0.1 percent. In other word, the recovery rate (from misspellings to correct spellings) reaches 99.9 percent.

An OCR normalizer is developed by using a program [25] which is open source software. We grouped the text embedded in the images into two categories: image text, Fig. 16.1(a), and *text generated image text* (tegit), Fig. 16.1(b). In the experiment, the OCR normalizer provided good outcomes in terms of being accepted by the Trigram normalizer or the Markov normalizer. Out of the 33 spam images we recently collected, 23 (70%) are tegits, and 10 (30%) are image texts. Among the 23 files which are produced by the OCR normalizer from tegits, we achieved a 100 percent recovery rate. And for the 10 files from image text, we achieved a 70 percent recovery rate. Therefore the overall weighted recovery rate is 91 percent.

This preliminary experiment is very encouraging, and a large scale evaluation of the effectiveness of the normalizers is on the way. We anticipate better results in the near future.

16.9 Conclusion and Future Work

The extendable software architecture with dynamical preprocessors, known as normalizers, for spam email filtering is presented. The most important feature of the architecture is its flexibility in easily adapting techniques to fight new spamming inventions. Normalizers can be easily introduced into or removed from the system. A normalizer is defined by its input format and output format, and a processing pathway is dynamically decided by chaining the normalizers based on their input and output format specifications. The spam detection engine, with the help of the normalizers, deals only with plain text format, which is pretty effective.

This architecture brings in a few advantages. First, the same detection engine is used to process the text part and other obscured parts of spam email. Training and updating the detection engine are very expensive operations and in most cases human intervention is needed. From an operational point of view, keeping one engine saves on cost, and from a system point of view, one engine preserves the integrity of the data and logic. Second, Bayesian filters of text based detection engines are the working horses on the field [22] and are also quite effective in detecting text based spam emails [1]. Our proposal takes full benefit of current text based detection engines. Finally, the system is ready to implement on real mail servers. The multiple normalizer pathways only add extra branches to the data flow of an existing spam detection setup. These extra branches of data flow can be easily and seamlessly integrated into the existing spam filters.

We have already developed several normalizers. Our next step is to integrate the normalizers into a single system and then conduct a large scale test of the effectiveness of the normalizers.

Acknowledgements This research work is supported by The Completion Grant of the Division of Business, Law and Information Sciences, University of Canberra and the Multidisciplinary Research Grant of the University of Canberra.

References

1. Cormack G. Lynam T (2005) TREC 2005 Spam Track Overview. In: The Fourteenth text retrieval conference (TREC 2005), Gaithersburg, MD, USA
2. Sahami, M, Dumais S et al (1998) A Bayesian approach to filtering junk e-mail. In: AAAI98 Workshop on learning for text categorization
3. Sakkis G, Androutsopoulos I et al (2003) A memory-based approach to anti-spam filtering for mailing lists. Information Retrieval 6(1):49–73
4. Carreras X, Marquez L (2001) Boosting trees for anti-spam email filtering. In: 4th International conference on recent advances in natural language processing (RANLP-2001)
5. Zhang L Yao T-s (2003) Filtering junk mail with a maximum entropy model. In: 20th International conference on computer processing of oriental languages (ICCPOL03)
6. Drucker H, Wu D, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Transactions on Neural Networks 10(5):1048–1054
7. Chuan Z, Xianliang L et al (2005) A LVQ-based neural network anti-spam email approach. ACM SIGOPS Operating Systems Review 39(1):34–39

8. Zhou Y, Mulekar MS, Nerellapalli P (2005) Adaptive spam filtering using dynamic feature space. In: 17th IEEE international conference on tools with artificial intelligence (ICTAI'05)
9. Graham-Cumming J (2006) The Spammers' compendium. 15 May 2006 [cited 2006 May]. Available from: *http://www.jgc.org/tsc/*
10. Cockeyed (2006) There are 600,426,974,379,824,381,952 ways to spell Viagra [cited 2006 October]. Available from: *http://cockeyed.com/lessons/viagra/viagra.html*
11. Wu C-T, Cheng K-T et al (2005) Using visual features for anti-spam filtering. In: IEEE international conference on image processing, 2005 (ICIP 2005)
12. Damiani E, Vimercati SDCd et al (2004) P2P-based collaborative spam detection and filtering. In: 4th IEEE international conference on peer-to-peer computing (P2P'04), Zurich, Switzerland
13. Albrecht K, Burri N, Wattenhofer R (2005) Spamato—an extendable spam filter system. In: 2nd Conference on email and anti-spam (CEAS'05), Stanford University, Palo Alto, California, USA
14. Yerazunis WS, Chhabra S et al (2005) A Unified model of spam filtration, Mitsubishi Electric Research Laboratories Inc: 201 Broadway, Cambridge, Massachusetts 02139, USA
15. Postel JB (1982) Simple mail transfer protocol [cited 2006 May]. Available from: *http://www.ietf.org/rfc/rfc0821.txt*
16. Freed N, Borenstein N (1996) Multipurpose internet mail extensions (MIME). Part 2: Media types [cited 2006 May]. Available from: *http://www.ietf.org/rfc/rfc2046.txt*
17. Ma W, Tran D et al (2006) Detecting spam email by extracting keywords from image attachments. In: Asia-Pacific workshop on visual information processing (VIP2006), Beijing, China
18. Tran D, Ma W, Sharma W (2006) Fuzzy normalization for spam email detection. In: Proceedings of SCIS & ISIS
19. Tran D, Ma W, Sharma D (2006) A Noise tolerant spam email detection engine. In: 5th Workshop on the internet, telecommunications and signal processing (WITSP'06), Hobart, Australia
20. Ma W, Tran D, Sharma D (2006) Detecting image based spam email by using OCR and Trigram method. In: International workshop on security engineering and information technology on high performance network (SIT2006), Cheju Island, Korea
21. Tran D, Ma W, et al (2006) A Proposed statistical model for spam email detection. In: Proceedings of the first international conference on theories and applications of computer science (ICTAC 2006)
22. Pelletier L, Almhana J, Choulakian V (2004) Adaptive filtering of spam. In: Second annual conference on communication networks and services research (CNSR'04)
23. Zdziarski JA (2005) Ending spam: Bayesian content filtering and the art of statistical language classification. No Starch Press, San Francisco, USA
24. Eikvil L (1993) OCR—optical character recognition. Norwegian Computing Center: Oslo, Norway
25. gocr (2006) *http://jocr.sourceforge.net*
26. Tran D, Sharma D (2005) Markov models for written language identification. In: Proceedings of the 12th international conference on neural information processing, pp 67–70

Chapter 17
Web OpinionPoll: Extensive Collection and Impression-based Visualization of People's Opinions

Tadahiko Kumamoto and Katsumi Tanaka

17.1 Introduction

The World Wide Web is remarkably developing as a simple and inexpensive means for sending information, and various people have been expressing their opinions from different standpoints on the Web. Consequently, the Web is rapidly becoming a treasure house of public opinion about any and all topics. In recent years, studies of extracting other people's opinions from the Web have been started [3–5]. They have proposed a system that collects other people's opinions about a user-specified event or thing from the Web, evaluates the opinions on several evaluation axes, and shows the results using a radar chart. However, their system cannot show a general view of the overall opinion because the system averages the opinions and forms a conclusion. We, therefore, propose a novel Web-based question answering system named Web OpinionPoll that collects other people's opinions about a user-given concern from the Web, and presents a general view of the opinions in a visual way. This system helps users intuitively understand answers to questions that request others' opinions.

Our proposed system extracts a character string[1] following the word *about* from the user question "What do people think about $\cdots$?" form, and then retrieves information from the Web regarding the character string as a query. For example, the

Tadahiko Kumamoto
Faculty of Information and Computer Science, Chiba Institute of Technology, 2-17-1 Tsudanuma, Narashino, 275-0016 Japan
e-mail: kumamoto@net.it-chiba.ac.jp

Katsumi Tanaka
Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501 Japan
e-mail: ktanaka@i.kyoto-u.ac.jp

[1] Because our target language of Japanese is an agglutinating language, a sentence is a character string.

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

character string “mothers always getting angry” is extracted from the user question “What do people think about mothers always getting angry?” and is entered as a query on the Web. When the number of hits is less than a user-specified number, nouns, adjectives, and verbs are extracted as target words out of the query, and candidate words for paraphrasing the target words are obtained based on information retrieval from the Web. Validity in paraphrasing these candidate words is tested using the co-occurrence and impression dictionaries that are automatically constructed by analyzing a huge volume of newspaper databases. Then, the system expands the initial query by paraphrasing one or more of the target words into the candidate words that were determined to be valid, and retrieves from the Web again. If the number of hits is still less than the user-specified number, connectivity of the expanded queries is gradually loosened. Finally, the system determines impressions of the Web pages consequently obtained, and plots the impressions on two impression planes. One is a plane spanned by two impression scales *Anticipation – Surprise* and *Acceptance – Disgust*, and the other is a plane spanned by another two impression scales *Joy – Sadness* and *Fear – Anger*. Users can form a general view of other people’s opinions about their concerns through these two impression planes. Note that, in the system, the impressions of a Web page are represented as scale values in the four impression scales mentioned above. Each scale value is a real number between 0 and 1 and is calculated from the title and snippet (three-line summary) obtained as a result of the Web retrieval.

The remainder of this chapter is organized as follows. Section 17.2 describes related work, and Sect. 17.3 describes a method of determining impressions of target text. Section 17.4 presents a system that extensively collects people’s opinions concerning a user-given concern from the Web and presents a general view of the opinions in a visual way, and Sect. 17.5 verifies the effectiveness of the system based on performance evaluation. Finally, Sect. 17.6 concludes this chapter and presents some problems to solve.

17.2 Related Work

A lot of question-answering systems have already been proposed as a means of searching out one or several answers to users’ natural language questions from a very large amount of documents, and presenting the answers in natural language sentences [1, 2]. A question-answering system is a kind of dialogue system. It has two characteristics that do not require users to enter retrieval keywords because retrieval keywords are automatically generated from users’ questions, and it does not force users to look for a correct answer from too many Web pages that are presented as a retrieval result. Current question-answering systems, however, cannot deal with questions requesting others’ opinions because correct answers cannot be identified for such questions. That is, their conversation style requires the existence of correct answers. In questions requesting others’ opinions, it is important to present what opinions people have concerning the questions by showing the variety and bias of

opinions and describing a local aspect and short summary of the overall opinion. Such a new conversation style should be introduced into the new type of question-answering systems we are aiming at.

There are studies on opinion mining, aiming to obtain people's opinions about user-specified products and services on the Web. For example, Morinaga's system collects the sentences in which opinions about a user-specified product or service were expressed from the Web, keeps score of each sentence, takes an average on each of several evaluation axes,[2] and plots the average values on a radar chart [3–5]. These systems can show how much people's opinions about products and services are positive or negative on the evaluation axes, but the knowledge sources or dictionaries required are handmade and must be prepared in each target task domain. This induces a situation where the systems do not work when users can ask questions without considering the range of a target task domain. In addition, it is not suitable for showing a general view of the overall opinion because a radar chart just shows an average view of the overall opinion. There are also studies on reputation analysis [6–8]. This is a kind of opinion mining, but their aim is to develop systems suitable for more general use. These kinds of systems, however, cannot show a general view of the overall opinion because they are focusing on the classification of documents such as reviews of books and movies into *popular* and *unpopular* classes.

17.3 Determining Impression of Text

In this chapter, the impressions of text are represented as scale values in four impression scales: *Anticipation – Surprise*, *Acceptance – Disgust*, *Joy – Sadness*, and *Fear – Anger*.[3] Each scale value is a real number between 0 and 1, and is calculated by consulting our impression dictionary and obtaining the scale values and weights of the nouns, verbs, and adjectives extracted from the target text. The impressions of target text are represented by an impression vector containing the four scale values calculated from the text.

17.3.1 Creating Impression Dictionaries

An impression dictionary was created by analyzing the Nikkei Newspaper Full Text Database [10] from the 1990 to 2001 editions as mentioned below, where each edition contains about 170,000 articles (200 MB), and the database contains two million or more articles in total.

---

[2] For example, the safety, driving performance, and design of cars might be considered as evaluation axes when we purchase a car.

[3] These scales were designed based on the eight basic elements of human feelings that Prof. R. Plutchik proposed [9].

Each entry in an impression dictionary corresponds to a word,[4] and has a scale value and a weight in each of target impression scales. The scale value in an impression scale of a word indicates which of two sets of impression words composing the impression scale the word co-occurs more often with, and is calculated using the following procedure.

First, in the Y edition, let the number of articles including any word in the set e of impression words be $df(Y,e)$, and let the number of articles including any word in e and target word w be $df(Y,e\&w)$. Joint probability $P(Y,e\&w)$ of e and w is then calculated by dividing $df(Y,e\&w)$ by $df(Y,e)$ as follows.

$$P(Y,e\&w) = \frac{df(Y,e\&w)}{df(Y,e)} \tag{17.1}$$

Next, interior division ratio $R_{e_1,e_2}(Y,w)$ of $P(Y,e_1\&w)$ and $P(Y,e_2\&w)$ is calculated by dividing $P(Y,e_1\&w)$ by the sum of the two probabilities, where e_1 and e_2 are contrast sets of impression words composing impression scale $e_1 - e_2$.

$$R_{e_1,e_2}(Y,w) = \frac{P(Y,e_1\&w)}{P(Y,e_1\&w) + P(Y,e_2\&w)} \tag{17.2}$$

where $R_{e_1,e_2}(Y,w) = 0$ when the denominator is 0.

Finally, the values of $R_{e_1,e_2}(Y,w)$ in all the editions are calculated, and the scale value $S_{e_1,e_2}(w)$ in $e_1 - e_2$ of w is calculated as their mean value,

$$S_{e_1,e_2}(w) = \sum_{Y=1990}^{2001} R_{e_1,e_2}(Y,w) \Bigg/ \sum_{Y=1990}^{2001} T_{e_1,e_2}(Y,w) \tag{17.3}$$

where $R_{e_1,e_2}(Y,w)$ of the editions in which the sum of $df(Y,e_1\&w)$ and $df(Y,e_2\&w)$ is equal to 0 are all excluded from the calculation. Because we know the words (for example, the words related to the Olympics) that do not appear every year but are strongly related to specific impression words, the $T_{e_1,e_2}(Y,w)$ term was introduced with such words. In the meantime, the values of $\sum_{Y=1990}^{2001} T_{e_1,e_2}(Y,w)$ and $\sum_{Y=1990}^{2001}(df(Y,e_1\&w)+df(Y,e_2\&w))$ may be large or small. Therefore, we also define the weight $M_{e_1,e_2}(w)$ in $e_1 - e_2$ of w in proportion to the number of the editions in which the sum of $df(Y,e_1\&w)$ and $df(Y,e_2\&w)$ is larger than 0 and the sum itself.

$$\begin{aligned} M = {} & \log_{12} \sum_{Y=1990}^{2001} T_{e_1,e_2}(Y,w) \\ & \times \log_{144} \sum_{Y=1990}^{2001} (df(Y,e_1\&w) + df(Y,e_2\&w)) \end{aligned} \tag{17.4}$$

[4] When a sentence is decomposed into words, postprocessing is done for the words. For instance, the verb *sakujo-suru (delete)* is decomposed into the action noun *sakujo (deletion)* and the verb *suru (do)*, but these two words are concatenated and form *sakujo-suru (delete)* again. Similarly, negative forms of nouns, verbs, and adjectives such as *mu-jiko (no accident)*, *kesa-nai (do not erase)* and *tanoshiku-nai (be not happy)* are handled as one word. This postprocessing is also performed in computing impression values of text and in expanding queries.

Table 17.1 Examples of entries in our impression dictionary

Entry Word	Anticipation – Surprise		Acceptance – Disgust		Joy – Sadness		Fear – Anger	
	S	*M*	*S*	*M*	*S*	*M*	*S*	*M*
get angry	0.107	1.304	0.170	1.179	0.274	1.300	0.021	1.622
childcare	0.285	1.346	0.336	1.199	0.604	1.273	0.404	1.105
dispatch	0.493	1.653	0.775	1.625	0.531	1.312	0.549	1.386
travel	0.309	1.737	0.442	1.499	0.659	1.675	0.425	1.405
smell	0.133	1.304	0.098	1.205	0.485	1.309	0.469	1.113
strong	0.397	1.489	0.190	1.221	0.575	1.270	0.422	1.159
new year's present	0.393	0.877	0.516	0.456	0.897	0.877	0.564	0.348
ghost	0.338	0.849	0.416	0.617	0.395	0.869	0.793	0.803

Table 17.2 List of impression words composing each impression scale

Scale	Impression words
Anticipation – Surprise	expectation, expect, anticipation, anticipate, forecast – surprise, be surprised, astonishment, astonish, admiration, admire
Acceptance – Disgust	agreement, agree, consent, acknowledgment, acknowledge, acceptance, accept – disgust, dislike, hate, be unpleasant, antipathy, have an antipathy, evasion, evade
Joy – Sadness	pleasure, be pleased, glad, happy, enjoy, blessing, bless – sad, feel sorry, sadness, sorrow
Fear – Anger	fear, be scary, misgivings, have misgivings, be frightened – anger, get angry, resentment, resent, rage, enrage

A portion of the impression dictionary constructed in the above way is shown in Table 17.1, and the impression words used in constructing the impression dictionary are listed in Table 17.2.

17.3.2 Computing Impression Values of Text

First, input text *TEXT* is segmented into words by using the Japanese morphological analysis system called *Juman* [12], and the words whose parts of speech are nouns, verbs, and adjectives are extracted from the input text.

Next, the scale value $S_{e_1,e_2}(w)$ and weight $M_{e_1,e_2}(w)$ of each word w are obtained by consulting the impression dictionary created in Sect. 17.3.1, and the scale value $O_{e_1,e_2}(TEXT)$ of the input text is calculated using the following equation.

$$O = \sum^{TEXT} S \times |2S-1| \times M \Big/ \sum^{TEXT} |2S-1| \times M \tag{17.5}$$

This calculation is done for every impression scale, and consequently an impression vector "($O_{Anticipation,Surprise}$, $O_{Acceptance,Disgust}$, $O_{Joy,Sadness}$, $O_{Fear,Anger}$)" is generated, where the $|2S-1|$ term denotes an inclined distribution depending on the

scale value *S*. Many of the words that appear in text seem to be independent of the impression of the text. This inclined distribution described here has been introduced to remove the adverse effect that such general words cause in calculation.

17.4 Answering Question Requesting People's Opinions

Figure 17.1 illustrates how our proposed system works for a user question such as "What do people think about ···?" form. Using ten example sentences listed in Table 17.3, we show the processing stages and characteristics of the system below.

In the Web retrieval stage, our system extracts a character string following the word *about* from user questions, and retrieves using Google [13] on the Web entering the character string as a query. For instance, the character string "mothers always getting angry" is extracted from example sentence No. 1 "What do people think about mothers always getting angry?", which is enclosed with a double quotation, and is entered as a query into the search engine. If the number of hits for this query is less than *N*, proceed to the query expansion stage. Otherwise, proceed to the final stage. Note that *N* denotes the lowest number of inquired pages, and is specified by users.

The query expansion stage consists of four substages. First, the query or character string is decomposed into words by using Juman. Next, nouns, verbs, and adjectives are extracted as target words for paraphrasing from the query, and queries for obtain-

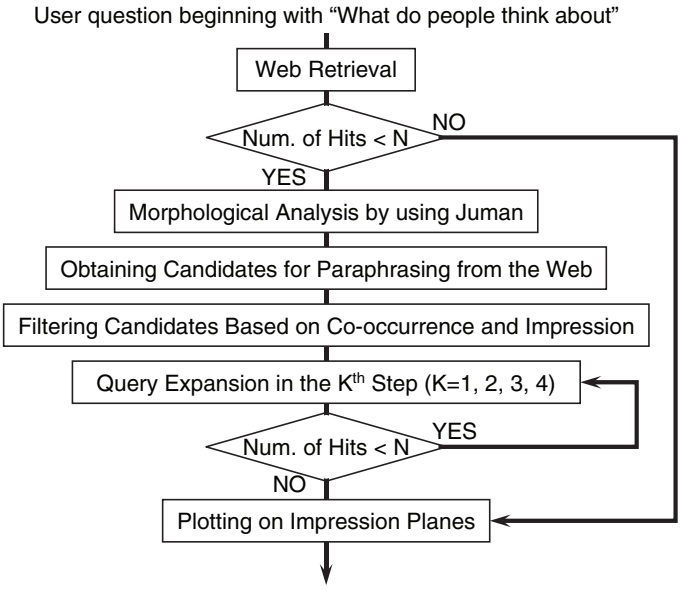


Fig. 17.1 Procedures to answer user questions requesting others' opinions

Table 17.3 Ten sentences used as user questions for test

No.	Character strings extracted from input sentences
1	mothers always getting angry
2	fathers who do not participate in childcare
3	the dispatch of the Self-Defense Forces to Iraq
4	traveling abroad during the Golden Week holiday
5	the hair liquid with a strong smell
6	giving a new year's present to infants
7	the ghost with cheerful character
8	making the child absent from school for traveling abroad
9	unlucky persons
10	shotgun marriage

ing candidate words for paraphrasing are generated by removing each of the target words from the initial query. The system retrieves information from the Web by entering the queries. For instance, the verb "getting angry" and the noun "mothers" are extracted as target words from the initial query "mothers always getting angry," and the queries "mothers always" and "always getting angry" are generated from the initial query and are retrieved from the Web. Of the retrieval results, the words that are observed in the same position with the target words removed are treated as candidate words for their paraphrasing.[5] We limited the number of candidate words per target word to ten or less in this chapter to save the time required for the query expansion stage.

Secondly, validity in paraphrasing the candidate words obtained is tested using the co-occurrence and impression dictionaries we constructed.

A co-occurrence dictionary was automatically constructed by analyzing the Nikkei Newspaper Full Text Database from the 1990 to 2001 editions, as well as the impression dictionary. First, we regard nouns in an article as entry words. The noun that is followed by and follows each entry word is extracted as a *followed-by* and *following* relationship, respectively. The verb or adjective that is followed by and follows each entry word is extracted as a *predicate* relationship. Next, we regard verbs and adjectives in an article as entry words. Similarly, the nouns immediately before and immediately after each entry word are extracted as the *followed-by* relationship. Each entry word has the followed-by, following, and predicate relationships, and these relationships are registered into a co-occurrence dictionary with their frequency data.

Validity in paraphrasing of the candidate words is tested using the co-occurrence dictionary constructed in the above way. First, the followed-by, following, and

[5] Parts of speech of candidate words for paraphrasing are limited by those of the target words. That is, when a target word is a common noun, the corresponding candidate word must be a sequence of nouns. When a target word is an action noun, the corresponding candidate word must be a sequence of nouns or a verb. When target words are a verb and an adjective, the corresponding candidate words are a verb/action noun and an adjective, respectively. Note that one or more particles (conjunctive particle "no (of)" and case particles) can be concatenated immediately before or immediately after a sequence of nouns, a verb, and an adjective.

predicate relationships are represented in a vector form.[6] Cosine similarity between a target word and each of the corresponding candidate words is computed using their vector forms by dividing the inner product of vectors generated from target and candidate words by the product of the size of the target word and the size of the candidate word. If it is less than 0.13, the candidate word is discarded. Here, the results of filtering the ten sentences listed in Table 17.3 are shown in Table 17.4.

Table 17.4 Filtering of candidate words using co-occurrence dictionary

(a) Valid paraphrasing

Target Word	Candidate word (Cosine similarity) ···
infant	child (0.50), child (0.37)
mother	mommy (0.59), man (0.42), myself (0.38), me (0.33), boss (0.24), mother (0.24), mama (0.21), life (0.19)
father	father (0.48), daddy (0.39), male (0.34), husband (0.33), child (0.33), man (0.31), child-nurturing (0.24), state (0.21), papa (0.17)
get angry	scold (0.19), nod (0.18), work (0.18)
character	voice (0.35), people (0.30)
do not participate in	participate in (0.44), face (0.29), confronting (0.21), concentrate (0.17)
abroad	domestic (0.45)
childcare	child-nurturing (0.16)
smell	smell (0.82), flavor (0.68), perfume (0.51), smell (0.21), spice (0.15)
do	consider (0.56), schedule (0.22),, plan (0.22), plan (0.19)
Golden Week	Chinese New Year (0.21)
ghost	people (0.14)

(b) Discarded candidate words

Target Word	Candidate word (Cosine similarity) ···
ryou (*)	amount (0.02)
infant	general (0.07), taka (*) (0.07), all members (0.06), whole (0.03), niece (0.02), parents and brothers (0.02), boss (0.01), chin (*) (0.00)
ghost	women (0.08), type (0.08), child (0.07), dog (0.07), student (0.06), activity (0.04), tea (0.04), needs (0.02), bar (0.01), tiger (0.00)
mother	full marks (0.12), stress (0.09)
father	concreteness (0.06), hometown (0.06)
get angry	quarrel (0.08), be greedy (0.05), birth (0.03), fight (0.03), tear (0.02), child-nurturing (0.02), growth (0.01), comparison (0.00)
character	the other side (0.07), high school (0.02), stand (0.01), green leaves (0.00)
do not participate in	involve (0.12), adapt (0.03)
abroad	family (0.09), drive (0.05), fu (*) (0.00)
childcare	home (0.06), conversation (0.05), activity (0.05), positive (0.04), region (0.04), event (0.03), education (0.03), school (0.02)
smell	oil (0.03)
Golden Week	student (0.01), positive (0.01), senior (0.01), objective (0.00), moon (0.00), fashion (0.00)

(The * mark denotes that it is a fragment word.)

[6] Each component of a vector corresponds to an element of the relationships, and its value denotes the number of cases.

Table 17.5 Filtering of candidate words using impression dictionary

(a) Valid paraphrasing

Target Word	Candidate word (Euclidean distance) ···
infant	child (0.08), child (0.12)
mother	man (0.01), life (0.02), mother (0.08), mommy (0.08), me (0.12), boss (0.16), myself (0.16), mama (0.18)
father	father (0.04), man (0.06), daddy (0.11), husband (0.15), male (0.19), child (0.21), papa (0.25)
get angry	scold (0.33)
character	people (0.10), voice (0.18)
do not participate in	participate in (0.21)
abroad	domestic (0.14)
childcare	child-nurturing (0.08)
smell	smell (0.19), flavor (0.20), smell (0.30)
do	consider (0.17), plan (0.31), schedule (0.32)

(b) Discarded candidate words

Target Word	Candidate word (Euclidean distance)···
ghost	people (0.47)
father	child-nurturing (0.36), state (0.56)
get angry	work (0.51), nod (0.72)
do not participate in	concentrate (0.37), confronting (0.39), face (0.43)
smell	perfume (0.68), spice (0.77)
do	plan (0.41)
Golden Week	Chinese New Year (0.81)

Validity of the candidate words passed is further tested using the impression dictionary. First, the Euclidean distance between impression vectors generated from target and candidate words is computed. If the distance is larger than 0.36, the candidate word is discarded. Otherwise, it is accepted. Results of the filtering are shown in Table 17.5. When a candidate word is a sequence of nouns, the similarity and distance of its sequence are defined as the maximum and minimum of the values calculated for components of its sequence, respectively. The query expansion substage proceeds from the first step to the fourth step, but it is terminated and goes to the final stage as soon as the number of hits is equal to or larger than N.

In the first step, zero or more target words in the initial query are paraphrased into the candidate words that were valid for the target words, so new queries are generated. All the queries that were generated from the example sentence No. 1 (“mothers always getting angry”) are shown in Table 17.6. We can see from this table that the term “getting angry” can be replaced with “scolding,” and “mothers” can be replaced with “me,” “mama,” or “mammy.”

In the second step, adnominal modifiers existing in each query are separated from the other part, and all the parts form a new query. For example, the query *Ikuji-ni-sanka-shi-nai-chichioya (fathers who do not participate in childcare)* is divided into two keywords *chichioya (fathers)* and *Ikuji-ni-sanka-shi-nai (do not participate in childcare)*, and these two keywords form a new query. In the third step, the character strings in each query are decomposed into phrases (bunsetsu), and all the phrases

Table 17.6 Output in the first step (paraphrasing) of query expansion substage

Queries	Number of hits
"Okotte-bakari-no-watashi (me always getting angry)"	1020
"Okotte-bakari-no-mama (mama always getting angry)"	450
"Okotte-bakari-no-okaasan (mommy always getting angry)"	430
"Okotte-bakari-no-jibun-ni (myself always getting angry)"	408
"Shikatte-bakari-no-watashi (me always scolding)"	398
"Shikatte-bakari-no-okaasan (mommy always scolding)"	223
"Okotte-bakari-no-hahaoya (mothers always getting angry)"	59
"Shikatte-bakari-no-mama (mama always scolding)"	38
"Shikatte-bakari-no-hahaoya (mothers always scolding)"	23
"Okotte-bakari-no-jinsei (my life in which I am always getting angry)"	20
"Shikatte-bakari-no-jibun-ni (myself always scolding)"	18
"Okotte-bakari-no-haha-ga (mothers always getting angry)"	9
"Okotte-bakari-no-joushi-ga (boss always getting angry)"	6
"Okotte-bakari-no-otoko-ippiki (a man who is always getting angry)"	0
"Shikatte-bakari-no-haha-ga (mothers always scolding)"	0
"Shikatte-bakari-no-joushi-ga (boss always scolding)"	0
"Shikatte-bakari-no-jinsei (my life in which I am always scolding)"	0
"Shikatte-bakari-no-otoko-ippiki (a man who is always scolding)"	0

Table 17.7 Change in number of hits by query expansion

No.	Before QE	First	Second	Third	Fourth
1 (18)	59	3,102			
2 (32)	289	739	11,292		
4 (16)	5	92		3,367	
5 (4)	0	7	47	8,503	
6 (3)	0	90		5,178	
7 (3)	1	5	250	55,710	
8 (2)	0	0		176	1,585

(The numbers in parentheses denote the number of queries generated in the first step.)

form a new query. For example, the query consisting of *chichioya (fathers)* and *Ikuji-ni-sanka-shi-nai (do not participate in childcare)* are modified, and a new query consisting of *chichioya (fathers)*, *Sanka-shi-nai (do not participate)*, and *ikuji-ni (in childcare)* is generated. In the fourth step, conjunctive particles are removed from each query. For instance, the query *Kaigai-ryokou-no-tame-ni-gakkou-wo-yasumaseru (making the child absent from school for traveling abroad)* is modified into *Kaigai-ryokou-no (traveling abroad)*, *tame-ni (for)*, *gakkou-wo (from school)*, and *yasumaseru (making the child absent)* by the second and third steps, and further into *Kaigai-ryokou (traveling abroad)*, *tame-ni (for)*, *gakkou-wo (from school)*, and *yasumaseru (making the child absent)* by this step.

Change in the number of hits by these steps of query expansion is shown in Table 17.7. Because we set the lowest number N of inquired pages to 1,000, the query expansion processing was terminated when the total number of hits exceeded 1,000.

In the final stage, impression vectors are generated from the top L Web pages finally obtained and are plotted on impression planes, where L denotes the maximum number of pages to be plotted, and is specified by users. The scale values of an impression vector are calculated from the title and snippet obtained as retrieval results.

Plotting on the impression planes for sentence No. 1 is shown in Figs. 17.2 and 17.3, where L was set to 1,000. Coordinate values and title of each plot are displayed on the right hand of the screen. Clicking a plot or a title on the screen causes the corresponding Web page to open.

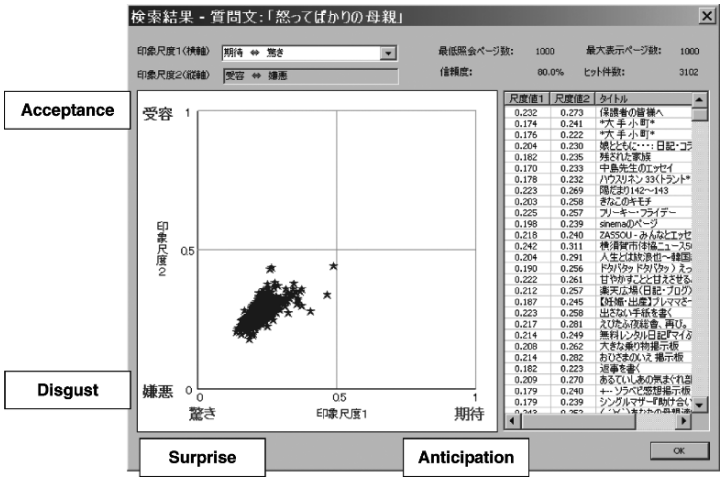


Fig. 17.2 Impression plane spanned by the two impression scales: Anticipation – Surprise and Acceptance – Disgust

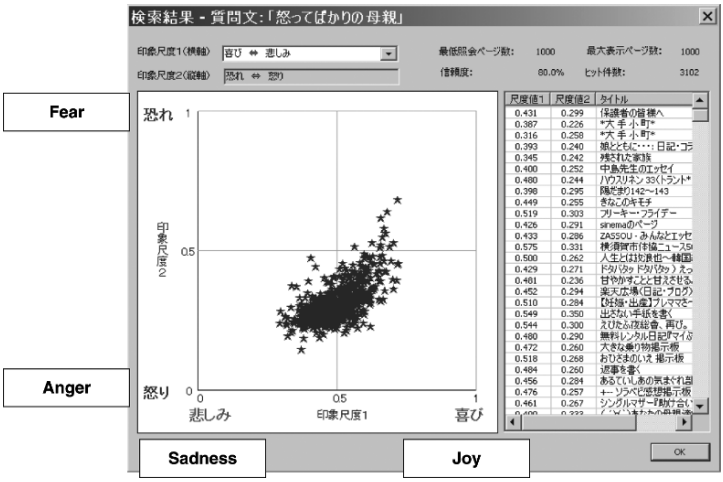


Fig. 17.3 Impression plane spanned by the two impression scales: Joy – Sadness and Fear – Anger

17.5 The Effectiveness of Web OpinionPoll

We evaluate the performance of the main parts of our proposed system, and verify their effectiveness.

Query expansion was applied to seven of the ten example sentences (see Table 17.7) under the condition that N is equal to 1,000. Consequently, 96 candidate words were obtained, as shown in Table 17.4. For each of the 96 candidate words, we estimated the validity of their paraphrasing on the four-step scale; "Always valid," "Valid in most cases," "Valid in its context," and "Not valid." The results are show in Table 17.8. Although 55 of the candidate words were rejected by the validity test using the co-occurrence dictionary, none of the 55 candidate words were estimated with *Always valid* and *Valid in most cases*, and only three were estimated with *Valid in its context*; *quarrel* for *get angry*, *education* for *child-nurturing* and *fight* for *get angry*. Although 15 of the 41 candidate words that were accepted by the validity test were estimated with *Not valid*, nine of the 15 candidate words were removed by the validity test using the impression dictionary. This demonstrated that the two-step filtering was effective in obtaining candidate words for paraphrasing. On the other hand, three of the twelve candidate words that were rejected by the impression-based filtering were *perfume* estimated with *Always valid* and *plan* and *spice* estimated with *Valid in its context*.

Next, we investigated change of accuracy and opinion rate[7] from the first step to the third step in the query expansion using sentence No. 5 *the hair liquid with a strong smell*.[8] The results are shown in Table 17.9.

Table 17.8 Filtering of candidate words by co-occurrence relationship and impression

(a) Co-occurrence relationship-based filtering

	Always valid	Valid in most cases	Valid in context	Not valid	Total
Passed	11	4	11	15	41
Rejected	0	0	3	52	55

(b) Impression-based filtering

	Always valid	Valid in most cases	Valid in context	Not valid	Total
Accepted	10	4	9	6	29
Rejected	1	0	2	9	12

[7] An opinion rate is calculated by dividing the number of Web pages including some opinions about a user-specified topic by the number of Web pages obtained as the retrieval results.

[8] The parameters were set as $N = 50$ and $L = 50$. The Web pages in which something about smell of a hair liquid were written were treated as *Correct*, and other web pages were treated as *Error*, and the web pages in which some opinions about smell of a hair liquid were explicitly written were treated as "including opinions," and other pages were as *not including opinions*. The numbers in parentheses of item *Number of pages* denote the number of Web pages that we could not access because the web pages had been deleted and rewritten. The number of pages in the third step was limited by L, but the number of Web pages hit was actually 8,503.

Table 17.9 Change of retrieval accuracy and opinion rate by query expansion

	Num of	Accuracy (Precision)		Opinion
	pages	Correct	Error	Rate
1st step	7 (2)	100% (5)	0% (0)	60.0% (3)
2nd step	47 (12)	77.1% (27)	22.9% (8)	92.6% (25)
3rd step	50 (11)	53.8% (21)	46.2% (18)	95.2% (20)

Because query expansion causes the relaxation of retrieval conditions, retrieval accuracy should decrease according to the progress of the query expansion. However, a high accuracy of 75 percent or more was obtained even when the query expansion proceeded to the second step. Here, we investigated how errors occurred. The main reason why they occurred was because *with a strong smell* and *the hair liquid* were used in different contexts in the same Web page. The reason why the third step of query expansion caused the accuracy to decrease to 50 percent was the same. We therefore consider that the system should retrieve on the Web, maintaining the proximity between character strings or that the system should filter out improper retrieval results based on a proximity criterion when a character string is divided into shorter character strings. On the other hand, a high grade was comparatively obtained for the opinion rates. We consider that the question setting of $\cdots$ *tte-dounano? (What do people think about* $\cdots$*?)* was effective in collecting Web pages, including opinions. The topic of *the hair liquid with a strong smell* may be familiar with the act of writing opinions. We should also investigate a variety of questions requesting others' opinions.

We think that plausible plotting on the impression planes was obtained for "mothers always getting angry?" as shown in Figs. 17.2 and 17.3. However, to extract impressions of the opinion that was written in a Web page, the present system does not identify the opinion part of the Web page. Instead, it uses the title and snippet (three-line summary) obtained as a retrieval result. Some of the Web pages collected by the system were blog and bulletin board pages. That is, the title of a page was irrelevant with respect to the content of the page. There were a lot of sentences or topics in a Web page, and the snippet generated from a Web page often had short sentence fragments according to the number of character strings in a query. This is one of the most important problems to solve.

Next, the time required to process the ten example sentences is shown in Fig. 17.4. If the number of candidate words increases, the number of queries sent to the Web increases. Consequently, too much time is required in the first Web retrieval, the Web retrieval for obtaining candidate words for paraphrasing, and the Web retrieval in every step of query expansion. However, we can set the parameters of the lowest number N of inquired pages, the maximum number of candidate words per target word, and the maximum number L of plots. According to the usage of the system, we are able to set the parameters to save the time required or to collect many more opinions.

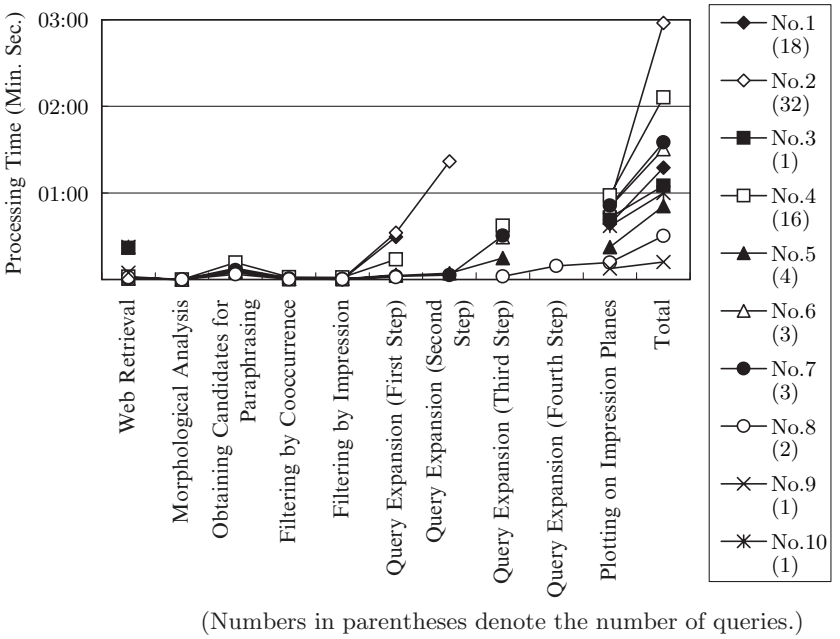


Fig. 17.4 Time required in each processing stage

17.6 Conclusion

We presented a system, named Web OpinionPoll, that collects the Web pages that are related to users' questions through Web retrieval when the users asked their concern in the form of "What do people think about ⋯?" and plots impressions of the Web pages consequently collected on two predefined impression planes. This system generates a query from a user's question, and expands the query using a paraphrasing technique, if necessary, to extensively collect the documents that are related to the question. The system uses co-occurrence and impression dictionaries to filter out improper candidate words for paraphrasing. The effectiveness of this system was verified by evaluating the validity of their paraphrasing, the accuracy of Web retrieval entering a character string as a query, and the time required to process the ten example sentences. The impressions of a web page are extracted not from the opinion part of the Web page, but from the title and three-line summary obtained for the Web page, due to the requirement to save time. Some of the Web pages collected by the system, however, were blog and bulletin pages, each involving multiple contexts, and opinions did not appear in the title very often.

We have many problems to solve and would like to focus on the following: 1) methods of identifying the opinion part in a Web page; 2) viewers for summarizing and clustering the overall opinion, extracting impressions from a cluster;

3) dialogue capabilities to answer various types of questions such as who, what, where, when, why, and how; 4) improvement of impression mining method; and 5) adaptation to individuality and personality.

References

1. Radev D, Fan W, Qi H, Wu H, Grewal A (2002) Probabilistic question answering on the web. In: Proceedings of WWW'02, Honolulu, USA
2. Tellex S, Katz B, Lin J, Fernandes A, Marton G (2003) Quantitative evaluation of passage retrieval algorithms for question answering. In: Proceedings of ACM SIGIR'03, Toronto, 41–47
3. Tateishi K, Ishiguro Y, Fukushima T (2002) Opinion information retrieval from the Internet. IPSJ SIG Technical Report, 2001-NL-144:75–82
4. Morinaga S, Yamanishi K, Tateishi K, Fukushima T (2002) Mining product reputations on the web. In: Proceedings of ACM SIGKDD'02, Edmonton, 341–349
5. Tateishi K, Fukushima T, Kobayashi N, Takahashi T, Fujita A, Inui K, Matsumoto Y (2004) Web opinion extraction and summarization based on viewpoints of products. IPSJ SIG Technical Report, 2004-NL-163:1–8
6. Turney P D (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of ACL'02, Philadelphia, 417–424
7. Dave K, Lawrence S, Pennock D M (2003) Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: Proceedings of WWW'03, Budapest, 519–528
8. Yi J, Nasukawa T, Bunescu R, Niblack W (2003) Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In: Proceedings of IEEE Int Conf Data Mining, Florida, 427–434
9. Plutchik R (1962) The emotions: facts, theories, and a New Model. Random House, New York
10. Nihon Keizai Shimbun Inc. Nikkei newspaper full text database DVD-ROM
11. Kumamoto T, Tanaka K (2005) Proposal of impression mining from news articles. Lecture Notes in Artificial Intelligence, LNAI3681:901–910, Springer, New York
12. Kurohashi S, Kawahara D (2003) Manual for Japanese morphological analysis system juman version 4.0
13. Google, http://www.google.co.jp/

Chapter 18
The Categorical Distinction Annexations to Maximal Associations Discovered from Web Database by Rough Set Theory to Increase the Quality

Erkan Ülker, Eyüp Sıramkaya, and Ahmet Arslan

18.1 Introduction

Most of the sectors transferred their information to the internet environment once the technology used became widespread and cheap. Print media organizations, which have a vital role in informing public opinion, and the data that are the services of these organizations are also shared over internet media. The fact that the continuously increasing amount of data includes the rich data it causes interesting and important data to be overlooked. Having large numbers of responses returned from queries about a specific event, person or place brings query owners face to face with unwanted or unrelated query results. Therefore, it is necessary to access textual data in press-publication sources that are reachable over internet in a fast and effective way and to introduce studies about producing meaningful and important information from these resources.

Association Rule Mining is the field of data mining, which defines the potential relations among data items in large databases. The process has two separate phases: The first one is to find the frequent item sets, and the second one is to determine the rules from these item sets. The oldest and most widely accepted algorithm used in the exploration (discovery) of association rules is the Apriori algorithm, the basic algorithm to determine the frequency of item sets, proposed by R. Agrawal et al. in 1993 [1]. Apriori is an influential algorithm for mining frequent item sets for Boolean association rules, and its different derivatives based on performance can be found in [1, 2, 16, 17], and [22]. Most of the other developed algorithms are improvements of the Apriori algorithm.

A regular association is an association that two key sets can be found in a document collection. It has several applications in different fields such as medicine [8], law [25], marketing [30], and manufacturing [4]. Regular association rules are based on the notion of frequent sets of attributes, whereas maximal association rules [28]

Erkan Ülker, Eyüp Sıramkaya, and Ahmet Arslan
Selcuk University, Department of Computer Engineering, 42075, Konya, Turkey

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

are based on frequent maximal sets [20, 23] and [29] of attributes [11]. The concept of maximal association rules is proposed by Feldman et al. [9]. One of the important advantages of maximal associations compared to regular associations is the ability to decrease the number of associations. Another advantage of maximal associations is that they do not eliminate the associations by assuming them to be weak associations, as regular associations do.

In applications in which the association rules are investigated, such as basket analysis, generally the category difference is not considered. The examples of these applications can be "73% of all customers who buy fish also buy dishwashing detergent" or "43% of all customers who buy the DVD movie also buy beer." The manufacturers of items that are from different categories but have strong relations can cooperate with each other according to the results of the obtained association. This may be useful for them regarding their profits. Furthermore, in consuetudinary methods, handling the items like one category allows us to perceive the deficiency of the algorithms, which aids in finding categorical associations.

In this study, we aim to mine significant and meaningful information from string data found on internet an newspaper. By handling binary situations such as person–person, person–place, event–place and event–date as categorical, these relationships can be learned as association rules by using data mining techniques.

In newspaper data, some events occurring in time can be focused or aimed at the interests of persons in a particular residential area. For such a query, the association analysis of Apriori and its derivatives can return disinterested information other than person and place information. Another method that can overcome this deficiency and is as efficient as Apriori is a new suggested algorithm that uses Rough Set Theory (RST).

RST is a useful tool for data mining. By using RST, rules similar to maximal associations are discovered. RST is simpler to model and to implement than the maximal relation method for knowledge discovered [11]. Bi et al. [3] try to find associations without considering the discrimination between categories. In addition to this study, we developed an algorithm that accounts for this discrimination. In this chapter, it has been shown that associations can be founded as categorical by using rough set theory. The experiments show that the proposed algorithm is very efficient and useful, and obtains the association rules faster than earlier methods.

The rest of the chapter is organized as follows: Section 2 gives a brief background on regular and maximal associations. This section also compares these two association rules via an example. Section 3 presents the proposed method for discovering categorical maximal association rules. Section 4 discusses the determination of categorical association rules using RST. Section 5 evaluates the experimental results. Finally, Sect. 6 concludes this chapter.

18.2 Backgrounds

18.2.1 Regular Association Rules Mining

If we consider the universe as the set of items at the market then each item has a Boolean variable representing the presence or the absence of that item [1, 13]. Thus, we can represent each basket by a Boolean vector. The Boolean vectors can be analyzed for buying patterns that reflect items frequently associated or purchased together. These patterns are represented in the form of association rules.

The support and confidence values are two measures of relatedness. They reflect the usefulness and certainty of discovered rules. If we consider the association rule $A \rightarrow B$ then a support of 3 percent means that 3 percent of all the transactions under analysis shows that A and B are purchased together. A confidence of 35 percent means that 35 percent of the customers who purchased A also purchased B. In general, association rules are taken into consideration if they satisfy both a minimum support threshold and a minimum confidence threshold. Such thresholds can be set by domain experts. Rules that satisfy minimum support threshold and a minimum confidence threshold are called strong. For more detailed information about regular association rules mining, the reader is referred to [1] and [13].

18.2.2 Maximal Association Rules Mining

The regular association rules [4, 7, 8, 14, 18, 30, 35, 36] are based on the notion of frequent sets of attributes that appear in many documents, whereas maximal association rules are based on the frequent maximal sets of attributes that appear maximally in many documents. [3, 9–11, 13] state that "for attribute sets X and Y, while a regular association rule X $\rightarrow$ Y means that if X then Y (with some confidence), a maximal association rule X $\rightarrow$ Y says that if X maximally then Y maximally." We also use this rule in this study. For detailed explanations about maximal association rule mining, the reader is referred to [9, 10], and [3, 11]. In the following paragraphs, we illustrate the principles of the maximally association mine with an example using the formalism presented in (Feldman et al. 1997, 1998; Guan et al. 2003; and Bi et al. 2003).

Let us consider a web-news database. We assumed that the news identification numbers P_i, the people in news, the location names, and details of news (or the content of news) are stored in a database. The appearance of databases is depicted in Table 18.1.

Table 18.1 The news database

News ID	David	Michael	George	…	Istanbul	Athens	…
P_1	Yes	Yes	No	…	Yes	No	…
…	…	…	…	…	…	…	…
P_{10}	Yes	Yes	No	…	Yes	No	…
P_{11}	Yes	Yes	Yes	…	No	Yes	…
…	…	…	…	…	…	…	…
P_{30}	Yes	Yes	Yes	…	No	Yes	…
…	…	…	…	…	…	…	…

Table 18.2 The categorical representation of Table 1 (a) T_1 = the people and (b) T_2 = the places

T_1 The People				T_2 The Places		
News ID	David	Michael	George	News ID	Istanbul	Athens
P_1	Yes	Yes	No	P_1	Yes	No
…	…	…	…	…	…	…
P_{10}	Yes	Yes	No	P_{10}	Yes	No
P_{11}	Yes	Yes	Yes	P_{11}	No	Yes
…	…	…	…	…	…	…
P_{30}	Yes	Yes	Yes	P_{30}	No	Yes
…	…	…	…	…	…	…

The regular associations and the confidence values can be found using Table 18.1 as follows:

{David, Michael} → {Athens} with confidence

$$\frac{|\sup(David, Michael, Athens)|}{\max\left(\sup\left(David, Michael\right), \sup\left(Athens\right)\right)} = \frac{20}{\max\left(30, 20\right)} = 66\%,$$

{David, Michael, George} → {Athens} with confidence $\frac{20}{20} = 100\%$, and
{David, Michael} → {Istanbul} with confidence $\frac{10}{30} = 33\%$

The vocabulary of keywords of these terms can be partitioned in terms of their categories. In the given example, there are two categories: the *People* names and the names of *places*, and they are shown in Table 18.2. A = {David, Michael, George,…, Istanbul, Athens,…} is the set of terms. This set includes two categories: T_1 = the People names and T_2 = the name of places. The classification is determined from the form {the People names, the name of places} or in short {*names, places*}.

Then, the sets having maximal support are found. The form {*the subsets of categories*: *categories*} in an "*information table*" is called "*a taxonomy pair*" and represented by $\{X_j : T_{Xj}\}$. In an information table, columns are attributes, rows are the objects of interest, and entries of the table are attribute values. For any subsets (territory) in a category, the notions of support of the subset is the number of rows supporting the subset, which is denoted by $Stg(X_j)$. The supported sets are obtained from Table 18.2 as follows:

{{David, Michael, George}: People} Stg=|{$P_{11} - P_{30}$}|=20
{{David, Michael}: People} Stg=|{$P_1 - P_{10}$, $P_{11} - P_{30}$}|=30
{{David, George}: People} Stg=|{$P_{11} - P_{30}$}|=20
{{Michael, George}: People} Stg=|{$P_{11} - P_{30}$}|=20
{{Michael}: People} Stg=|{$P_1 - P_{10}$, $P_{11} - P_{30}$}|=30
{{David}: People} Stg=|{$P_1 - P_{10}$, $P_{11} - P_{30}$}|=30
{{George}: People} Stg=|{$P_{11} - P_{30}$}|=20
{{Istanbul}: Places} Stg=|{$P_1 - P_{10}$}|=10
{{Athens}: Places} Stg=|{$P_{11} - P_{30}$}|=20

For each member of the supported set, the remaining attribute-value of associated categories, except the member of the set's value, is checked. If this value is not "0" (i.e., No) then the maximally supported sets are found by deducting the corresponding row from the number of members. For any subset in a category, the notions of maximally supports of the subset is the number of rows maximally supporting the subset, which is denoted by $\mathrm{Msg}(X_j)$. Next, the maximally supported sets from supported sets are obtained as follows;

{{David, Michael, George}: People} Msg=|{$P_{11} - P_{30}$}|=20
{{David, Michael}: People} Msg=|{$P_1 - P_{10}$}|=10
{{Istanbul}: Places} Msg=|{$P_1 - P_{10}$}|=10
{{Athens}: Places} Msg=|{$P_{11} - P_{30}$}|=20

In Table 18.1, thirty rows (30 news) are checked as 1 or "Yes" under the columns named as "David" and "Michael" in the people category. For the same thirty rows in Table 18.1, twenty rows are checked as 1 or "Yes" under the columns named "George". To determine the maximal support for "David" and "Michael," we eliminate the same rows. As a result, ten rows remain, which are {{David, Michael}: People} Msg=|{$P_1 - P_{10}$}|=10.

In the next phase, the sets containing maximal support are split based on their categories. This process is shown in Fig. 18.1. In this figure, all taxonomy pairs and their maximal supports are placed in the box at the top. Then, their element in this box is distributed into two sub boxes containing people and place categories. All members of the distributions are crossed with each other. Then, all created pairs are checked if they provide maximality by comparing with Table 18.1. Any pair that does not provide maximality is eliminated. For example, when we look at the first row in Fig. 18.1, in Table 18.1, twenty rows with attribute values are checked as "1" ("Yes") under columns "David," "Michael," "George" and "Athens" where remaining attribute values are checked as "0" ("No"). Similarly, for the second row in Fig. 18.1, in Table 18.1, ten rows with attribute values are checked as "1" under the columns "David," "Michael," and "Istanbul" where remaining attribute values is checked as "0." However, for the third row, there is no row with attribute values checked as "1" under columns "David," "Michael," "George," and "Istanbul."

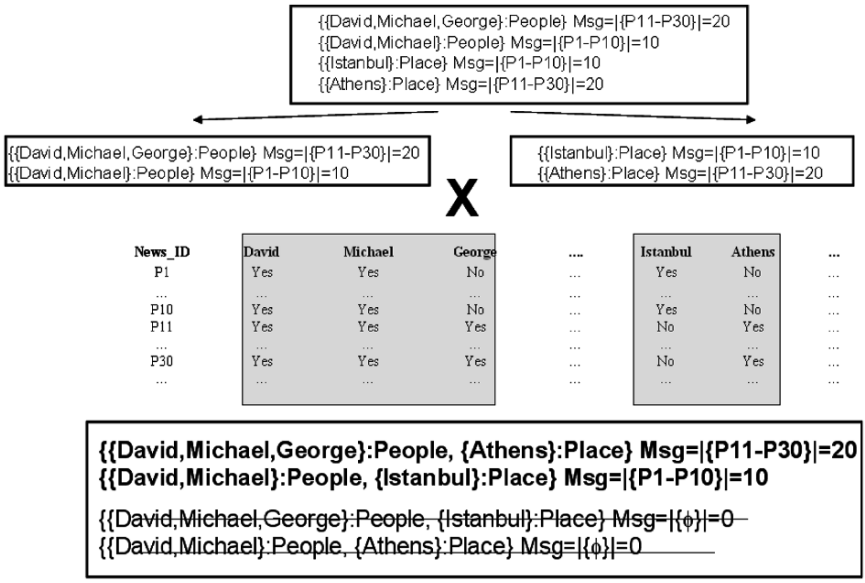


Fig. 18.1 Categorizing the maximal supported sets in Table 18.2

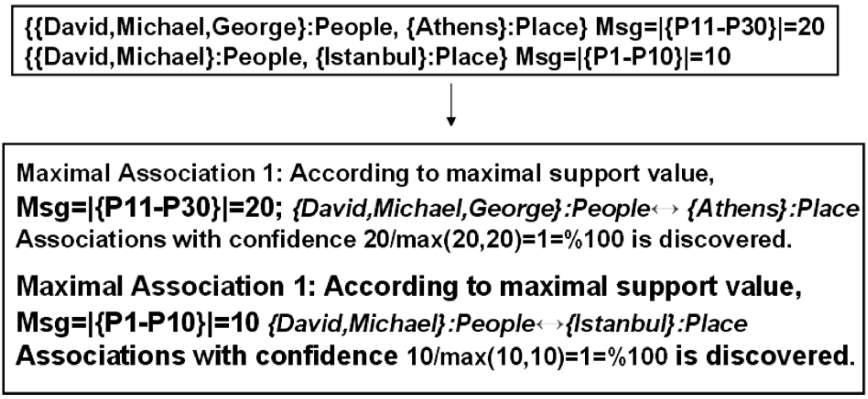


Fig. 18.2 The discovered maximal associations from Table 18.1

In the next step, the maximal association X ↔ Y between X and Y is achieved from the maximal supported sets. The maximal confidence of association (*Maxconf*) is calculated by using maximal supports as follows:

$$Maxconf(X \leftrightarrow Y) = \frac{Msg(X \cup Y)}{Max(Msg(X), Msg(Y))} \tag{18.1}$$

The obtained maximal associations and the confidence values are depicted in Fig. 18.2.

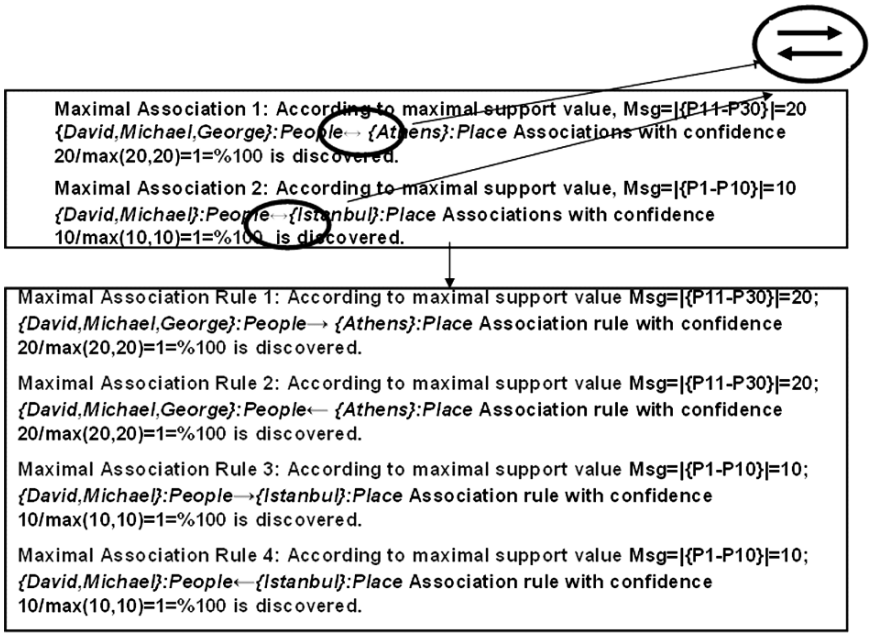


Fig. 18.3 The discovered maximal association rules from Table 18.1

Next, the maximal association rules X → Y or X ← Y between X and Y are achieved from the maximally supported sets. The maximal confidence of association rules are calculated by using maximal supports as follows:

$$Maxconf(X \rightarrow Y) = \frac{Msg(X \cup Y)}{Msg(X)}, Maxconf(X \leftarrow Y) = \frac{Msg(X \cup Y)}{Msg(Y)} \quad (18.2)$$

The maximal association rules and the maximal confidence values obtained from Fig. 18.2 are depicted in Fig. 18.3.

As can be seen in Fig. 18.3, the confidence value of association rule {David, Michael} → {Istanbul} with the maximal association rule mining is $10/10 = 100$ percent while the confidence value of association rule {David, Michael} → {Istanbul} with the regular association rule mining is $10/30 = 33$ percent. This will strengthen the judgment of "One of the important advantages of maximal associations compared to regular associations is the ability to decrease the number of associations. Another advantage is that maximal associations do not eliminate the associations as regular associations do since regular associations techniques always assume these as weak associations" as we stated in Sect. 1. The steps we discussed in this section can be done more easily using RST. In the next section, the basic concepts of RST will be given.

As has been realized, the method we discussed so far is very complex and may not find all the association rules. In the following sections, we will present a new

technique that is easy to implement, and it finds more association rules according to maximal association mining. In the technique, we use RST, which we present next.

18.3 Proposed Categorizing Associations Rule Mining Method

In this section, we first explain the fundamentals of RST. We then discuss the determination of categorical association rules using RST. Furthermore, we illustrate the effectiveness of using RST by comparing the proposed method with traditional maximal association mining.

18.3.1 The Basics of Rough Set Theory (RST)

The rough set philosophy was founded on the assumption that with every object of the universe of discourse we associate some information (e.g., data, knowledge). Objects that are characterized by the same information are indiscernible (similar) in the view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of RST [26]. The RST approach, which was first introduced by Pawlak in 1982 [25], is a formal construction to transform data into knowledge. The appearance of knowledge can be presented as classification or decision rules [12]. Data is often presented as a table—columns and rows of which are labeled by attributes and by objects of interest and entries, respectively. Such tables are known as information systems, attribute-value tables, or information tables [26]. An information system is composed of a 4-tuples: $S =< U, Q, V, f >$, where U is the closed universe (i.e., a finite set of N objects $\{x_1, x_2, \ldots x_n\}$), Q is finite attributes $\{q_1, q_2, \ldots.. q_n\}$, $V = \cup_{q \in Q} V_q$ where V_q is a value of the attribute of q, $f : UxQ \rightarrow V$ is the total decision function called the information function such that $f(x, q) \in V_q$ for every $q \in Q$, $x \in U$. The set of attributes, $Q = \{C \cup D\}$, consists of a few condition attributes, C, and one or more sets of decision attributes, D. Any set of all indiscernible objects is called an elementary set, and it forms a basic granule (atom) of knowledge about the universe. Any union of some elementary sets is referred to as a crisp (precise) set—otherwise the set is rough (imprecise, vague). Approximations are two basic operations in RST. The lower approximation consists of all objects that surely belong to the concept, whereas the upper approximation contains all objects that possibly belong to the concept. Obviously, the difference between the upper and the lower approximation constitutes the boundary region of the vague concept.

The use of artificial intelligence techniques for discovering the association rules is increasing quickly. Especially, genetic algorithms [21,27], and [35], artificial immune systems [7], and fuzzy logic [14,36] have been used recently. Some techniques such as self organizing map (SOM) [15,19], and fuzzy clustering [32] have been implemented as the preprocessing techniques. The more interesting implementations,

which were implemented by Coenen et al. [6], are T-Trees and P-Trees. RST is especially used for classification in artificial intelligence. Wang et al. [33] have used rough set method for data discrimination and finding the associations. Clare et al. [5] benefited from rough sets in multiple association rules. Ma et al. [24] obtained associations with single and multiple dimensions via rough sets.

When discovering the maximal association rules using RST, the database at hand can be considered as an information system. The basic information is found by using the indiscernibility relation and then associations are discovered with lower approximation by using the concept of approximation. For more detailed information about the rough set method, the reader is directed to [26]. In the next section, an algorithm to discover the categorical maximal associations will be presented.

18.3.2 The Discovery of Categorical Maximal Associations by Rough Set Theory

Rough set theory is a useful tool for data mining. The rules that are similar to the categorical associations can be found by using rough set theory and this process is simpler than regular association method for info survey [11]. The proposed algorithm, which is composed of six steps, is summarized as follows:

Step 1: The Information System Table is created;
Step 2: The elementary sets belonging to the Information System are found;
Step 3: The Information Table is made meaningful for associations by using the elementary sets;
Step 4: The information table is divided into separate attributes;
Step 5: The lower approximations are calculated and the supported sets of attributes are found until all combinations of each attribute pair are discussed and finished;
Step 6: Each attribute pair are turned into a rule according to a confidence value.

For handling the problem by Rough Set Theory, firstly current item sets are reformed as information system tables. Here, concerned objects are the links of the news. Attributes: name of the persons (Bush, Erdogan etc.), name of the places (Istanbul, Konya etc.), intervals of the date (April 2005, July 2005, etc.) and events (negotiation, terror, etc.). Entries of the table are setting by giving the binary values 1 and 0 to persons, places and events that take place in that news for each object. An example information system is given in Table 18.1. In Table 18.1, the universe of the information system, U, is constructed by a finite number of objects, P_i and $U = \{P_1, P_2 \ldots P_{30} \ldots P_n\}$. The attributes of information table are people and places categories (e.g. $Q = \{\text{David, Michael, George} \ldots \text{Istanbul, Athens} \ldots\}$). If one or more attributes occur in any news, the Boolean value of "1" or "0" is overwritten onto the value of "Yes" or "No" in the corresponding table entry (e.g. $V_1 = \{1,0\}$, $V_2 = \{1,0\}$, $V_3 = \{1,0\}$). According to the first forty-five

records, the obtained information system table derived from Table 18.1 is shown in Table 18.3.

In the second step, the elementary sets are found by using the indiscernibility relation of RST. The total supports of the elementary sets are added to the information system table as a column. The elementary sets achieved from the information system table in Table 18.3 are given in Table 18.4.

In the third step, the valid attributes at the elementary sets are grouped in terms of their categories. The process of categorization is partially performed in this step. In this step, no process is applied to the support column (the last column in Table 18.3). The meaningful information table for associations extracted from the Table 18.4 is presented in Table 18.5. Here the term T_k represents the kth category, and T_1 and T_2 represent people and places categories, respectively.

In fourth step, a categorical table is derived from the information table. The number of rows of this table equals the number of categories and the number of columns

Table 18.3 The information system table derived from Table 1

U	David	Michael	George	….	Istanbul	Athens	…
P_1	1	1	0	…	1	0	…
…	…	…	…	…	…	…	…
P_{10}	1	1	0	…	1	0	…
P_{11}	1	1	1	…	0	1	…
…	…	…	…	…	…	…	…
P_{30}	1	1	1	…	0	1	…
P_{31}	0	1	1	…	1	0	…
…	…	…	…	…	…	…	…
P_{45}	0	1	1	…	1	0	…
…	…	…	…	…	…	…	…

Table 18.4 The elementary set belonging to the information table (After grouping of indiscernible objects)

U/A	David	Michael	George	….	Istanbul	Athens	Support MSG(XUY)
$\{P_1,\ldots,P_{10}\}$	1	1	0	…	1	0	10
$\{P_{11},\ldots,P_{30}\}$	1	1	1	…	0	1	20
$\{P_{31},\ldots,P_{45}\}$	0	1	1	…	1	0	15
…	…	…	…	…	…	…	…

Table 18.5 The categorical information system the meaningful for associations

U/A	T_1	T_2	….	T_K	Support MSG(XUY)
$Set_1\{P_1 : P_{10}\}$	{David, Michael}	{Istanbul}	…	…	10
$Set_2\{P_{11} : P_{30}\}$	{David, Michael, George}	{Athens}	…	…	20
$Set_3\{P_{31} : P_{45}\}$	{Michael, George}	{Istanbul}	…	…	15
…	…	…	…	…	…
Set_N	…	…	…	…	…

Table 18.6 The categorical table after calculating the total maximal supports

Category	Items($Set_i : T_i$)	Items($S : T_i$)	Items($S : T_i$)	Items($S : T_i$)
T_1	{David, Michael} MSG() = 10	{David, Michael, George} MSG() = 20	{Michael, George} MSG() = 15	...
T_2	{Istanbul} MSG() = 10 + 15	{Athens} MSG() = 20	{Istanbul, Athens} MSG() = ...	...
...	...	...	...	...
T_K	...	...	...	...

equals the number of maximal elementary sets plus one. For each number of a category, its maximal supports (MSG) are assigned. The categorical table obtained from Table 18.5 is presented in Table 18.6.

In Table 18.6, for the attributes $T_1(=\text{People}),\ T_2(=\text{Places}),\ldots,T_K$, the fragment of $U/T_1 = U/\text{People}$ consists of the elementary sets of $W_1 = \{P_1,..,P_{10}\}$, $W_2 = \{P_{11},..,P_{30}\}$, and $W_3 = \{P_{31},..,P_{45}\}$. In the same way, the fragment of $U/T_2 = U/\text{Places}$ contains the elementary sets of $Z_1 = \{P_1 \ldots\ P_{10},\ P_{31} \ldots\ P_{45}\}$ and $Z_2 = \{P_{11} \ldots\ P_{30}\}$.

In the fifth step, the categories in which the associations are searched are decided. The combinations of T_1XT_2, T_1XT_3, T_2XT_3, $T_1T_2XT_3$, $T_1T_3XT_2$ or $T_1XT_2T_3$ are considered for the three-category-information system, whereas the categories T_1 and T_2 are considered for the two-category-information system. Note that the combination of T_1XT_2 is not a one-to-one mapping from T_1 to T_2, rather all the member of T_1 mapped to only one member of T_2. For example, the combination of T_1XT_2 is {all members}:T_1 X {Istanbul}:T_2. The maximal support of this combination is $10+15=25$, that is, the total of MSG () for the rows consisting of "Istanbul" at the column of T_2 in Table 18.5.

As defined above, the supported sets from the attribute People to the attribute values of Places can be listed as follows:

The maximally supported set from the attribute People to the attribute value $Z_1 = \{P_1,\ldots,P_{10},\ P_{31},\ldots,P_{45}\} = \{\text{Istanbul}\}$ is $Support_{people}(Z_1) = \cup_{i=1,2;W_i\subseteq Z_1} W_i = W_1 + W_3 = \{P_1,\ \ldots,\ P_{10},\ P_{31},\ \ldots,\ P_{45}\}$, and the maximally support is Msg(Istanbul) $= 10+15=25$.

The maximally supported set from the attribute People to the attribute value $Z_2 = \{P_{11} - P_{30}\} = \{\text{Athens}\}$ is $Support_{people}(Z_2) = \cup_{i=1,2;W_i\subseteq Z_2} W_i = W_2 = \{P_{11},\ \ldots,\ P_{30}\}$, and the maximally support is Msg(Athens) = 20.

In a similar manner, the supported sets from the attribute Places to the attribute values of People are as follows:

$Support_{places}(W_1) = \cup_{i=1,2,3;z_i\subseteq w_1} Z_i = \{P_1 - P_{10}\}$, and the maximally support is Msg (David, Michael) = 10.
$Support_{places}(W_2) = \cup_{i=1,2,3;z_i\subseteq w_2} Z_i = Z_2 = \{P_{11} - P_{30}\}$, and the maximally support is Msg(David, Michael, George) = 20.

$Support_{places}(W_3) = \cup_{i=1,2,3;z_i \subseteq w_3} Z_i = \{P_{31} - P_{45}\}$, and the maximally support is Msg(Michael, George) = 15.

In the sixth step, every attribute pair is transformed into a rule according to a confidence value. The X ↔ Y association confidence calculated by the formula is

$$X \leftrightarrow Y_{Conf} = \frac{MSG(X \cup Y)}{MAX\,(MSG(X), MSG(Y))} \tag{18.3}$$

Whereas the X → Y association rules confidence is calculated by the formula

$$X \rightarrow Y_{Conf} = \frac{MSG(X \cup Y)}{MSG(X)} \tag{18.4}$$

For the example above, the obtained association rules after processing the sixth step of RST and the confidences calculated according to the Apriori algorithm are presented below for comparison:

{David, Michael}: People → {Istanbul}: Place with confidence $\frac{|Support_{people}(Z_1)|}{|W_1|} = \frac{|\{P_1{:}P_{10}\}|}{\max(|\{P_1{:}P_{10}\},\ \{P_1{:}P_{10},\ P_{31}{:}P_{45}\}|)} = \frac{10}{25} = 40\%$ according to People (W_1) = {David, Michael}. (The Confidence for Apriori Algorithm = %33).
{David, Michael, George}: People → {Athens}: Place with confidence $\frac{|Support_{people}(Z_2)|}{|W_2|} = \frac{|\{P_{11}{:}P_{30}\}|}{\max(|\{P_{11}{:}P_{30}\}|)} = \frac{20}{20} = 100\%$ according to People (W_2) = {David,Michael, George}. (The Confidence for Apriori Algorithm = %100).
{Michael, George}: People→{Istanbul}: Place with confidence $\frac{|Support_{people}(Z_1)|}{|W_3|} = \frac{|\{P_{31}{:}P_{45}\}|}{\max(|\{P_1{:}P_{10}\},\{P_1{:}P_{10},P_{31}{:}P_{45}\}|)} = \frac{15}{25} = 60\%$ according to People (W_3) = {Michael, George}. (The Confidence for Apriori Algorithm = %33).

In same way, the association rules discovered from the attribute Places to the attribute People are illustrated below;

{Istanbul}: Place → {David, Michael}: People with confidence $\frac{10}{25} = 40\%$. (The Confidence for Apriori Algorithm = %40).
{Athens}: Place → {David, Michael, George}: People with confidence 100%. (The Confidence for Apriori Algorithm = %100).

These rules are the same as those maximal association rules discovered from Table 18.1 as shown in Fig. 18.3. However this example demonstrates that the categorical rough set approach to discovering categorical knowledge is much simpler and faster than the maximal association method.

18.4 Categorical Association Rules Mining System in Press-Publication Field

The present study aims to access visual and text data taking place in press-publication sources that are reachable over the Internet in a fast and effective way and to obtain important and meaningful information from these sources. The studies are focused on the news about persons and organisms that are important in terms of secret information or intelligence. The servers have databases, which consist of processed text documents that are collected from news sources existing on the Internet, and application software that process this information. Initially, information in pages related to terror groups, prepared by Işık University [34], were examined and important names recorded. Later on, the names mentioned in these resources are searched using Google, and results consisting of text and pictures were found and saved to a database for processing. Saved documents also consisted of unnecessary information. Data was processed partly by text analysis methods, and only the required information was saved to the database to be used later.

The relationships in collected data such as person, place, event and date were learned by Apriori algorithm as can be seen in the study of the author of this chapter [31]. But it is not the issue to have category separation in the present study. Databases can be richer by providing research on the new names obtained after the results of learned relationships and repeating these calculations mentioned above.

A document that includes all information about searched people is presented to the user. In this information, there are people's names, events that take place in news, places and dates of events, and the accompanying people's names if mentioned. The user can do new searches guided by the new information since the relationships are presented to the user. The difference between the present and previous study of the author of this chapter appears in the method (Rough Set Theory) used and the founded relationships (categorical).

An outline of the system is showed in Fig. 18.4.

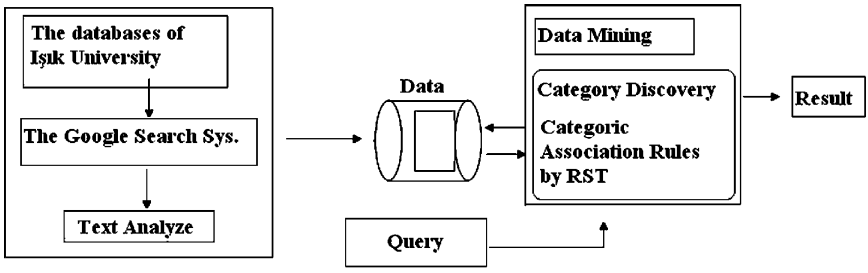


Fig. 18.4 The categorical association rule mining system

18.5 Experimental Results

A storage mechanism is needed that provides for saving, processing and querying the obtained data. Thus, a database should be designed that is safe and efficient but also lets the required inquiries be done as quickly as possible. For this purpose a textual database was formed by using the MySQL database manager system. For collecting the information about people who are important from the standpoint of intelligence, the database ([34]) about Terrorist Organizations and relations of these was chosen as the prior source. The database was prepared by Sonmez Koksal, an instructor at Işık University and former undersecretary of a national intelligence agency. This source was used for getting primary information about founders and important members of the organizations.

Textual information is a very important part of the news. For this reason it is very important to make a detailed investigation of the text, to extract items from the text such as time, place and people's names, and to parse them from other words. This information has been parsed partially and manually until now [31].

Then, the inputs of the rough set information system table that was formed by accepting these parsed keywords as items are filled. If a concerned attribute (person, place etc.) exists in that news, then the suitable column and row is valued as 1; if the attribute does not exist, it is categorized as 0. For decomposition of these attributes into the categories, these four different tables of categories are used. In Table 18.7, a part of the person-place-event-date query is seen, including confidence and support values.

Table 18.7 The partition from queries of person-place, person-person and person-event with confidence and support values

A	B	Support $(A \Rightarrow B)$	Confidence $(A \Rightarrow B)$	The Founded Items
Kemal pir, Ocalan	Turkey	0.31	0.573	Terror, Organization, Martyr, 2001, 1999, 1985, 1995, Şemdin Sakık, Talabani,...
The Problem of Kurd.	Barzani	0.672	0.634	Saddam Hüseyin, Usame Bin Laden, Talabani, Al Qaeda, Iraq, Turkey, Tayyip Erdoğan,..
PKK, Ocalan	USA, Greece	0.48	0.681	Al Qaeda, Roj TV, Belgium, Denmark, Greek Cyprus, Terrorist Organizing List,...
Laden	Bush	0. 198	0. 1212	USA, Bombing, Iraq, Attack, Searching, Ring Leader, Twin Tower, Accusation, Terror,...
Abdullah	Iraq	0.8671	0.003	Bülent Ecevit, Turkey, 2002, Interpretation, Terror, Tayyip Erdoğan,...

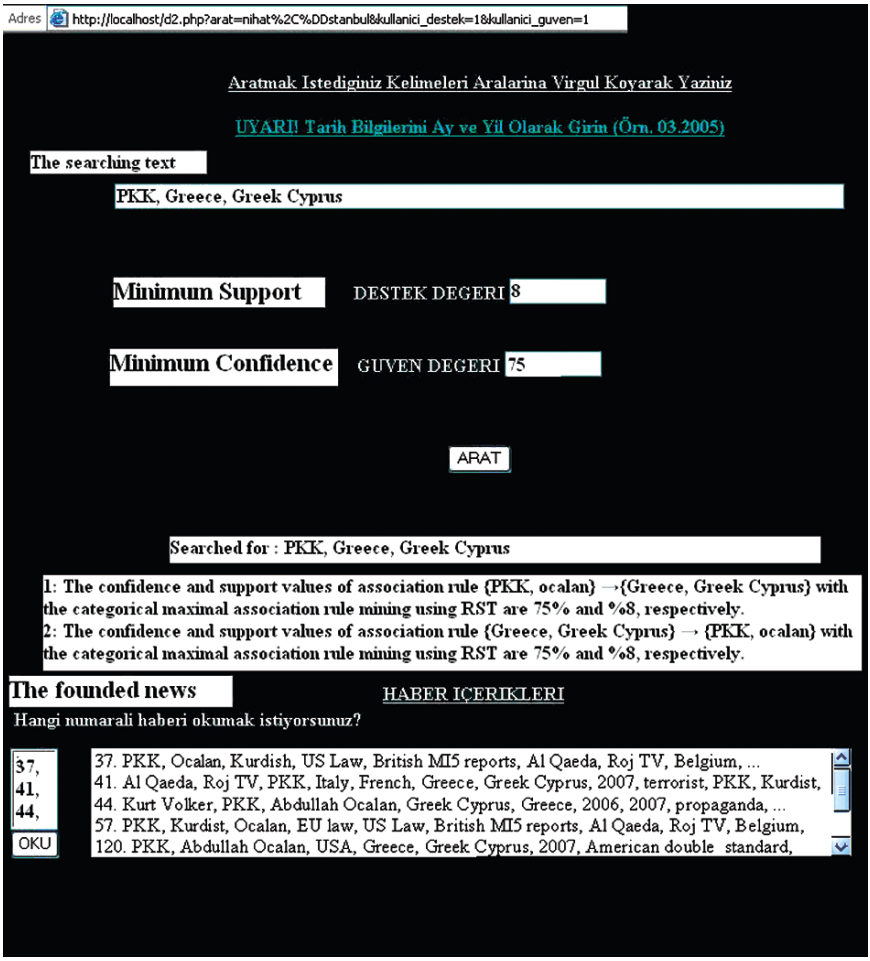


Fig. 18.5 The window of query input

A view of the PHP interface that enables the query is presented in Fig. 18.5. The display obtained when the concerned news is obtained as a result of the query is given in Fig. 18.6.

Fig. 18.6 The content of the news

18.6 Conclusions

In recent years, there have been several studies about association in large databases. It has been observed that the proposed methods in these studies cannot find the associations among the items having strong relations as they focus on the regular associations, or else these associations are eliminated as a result of user specified support and confidence values. Motivated by this observation, some researchers have introduced maximal associations to improve the quality of the discovered association rules. In a transaction database, regular association rules are based on the notion of frequent sets of attributes that occur in many rows of a database whereas maximal association rules are based on frequent maximal sets of attributes that appear maximally in many rows of a database.

Association algorithms that are generally accepted, such as Apriori, are mining by accepting all items belonging to one category. However the category discrimination is done in the most of the area of the day life—for example, people, animals, plants, etc. It may be presumed that this category discrimination also exists in the current working database. For example, in a basket analysis the items can be grouped under categories such as clean-up products, home devices and food products. Generating results according to categories can lead to new ideas for the algorithms that find the associations, and maybe this will increase the quality of

the discovered rules. Implementing discrimination of categories with rough set methodology is easier than implementing it with Apriori methodology.

In this chapter, how the categorical association rules will be discovered from databases using Rough Set Theory has been investigated, and a related application has been represented. In applications, it has been seen that reduction of calculation times and category discrimination can be done by Rough Set. Also any final opinion should be reached carefully since most of the factors carry a great amount of influence.

A document is presented to user that includes all information about the people who are sought. This information contains the name of the person, influential events, place, dates and the names of the accompanying people if mentioned in the news. Since the relations are represented to the user, the user can make new queries in the guidance of new information. This system, which is constituted by using a database approach and the algorithm that finds the categorical associations with a rough set, is an application prototype that shows how the association rules mining can works in the domain of intelligence works. These kinds of applications will also be very useful for discovering the secret relations between the members of the terrorist organizations.

References

1. Agrawal R, Imielinski T, Swami A (1993) Mining Association Rules between Sets of Items in Large Databases. In: Proc. of ACM SIGMOD Conf. on Management of Data. Washington, D.C., United States, pp. 207–216
2. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proc. 20th Very Large Databases Conference in Santiego Chile, pp. 487–499
3. Bi Y, Anderson T, McClean S (2003) A rough set model with ontologies for discovering maximal association rules in document collections. J Knowledge-Based Systems 16:243–251
4. Chen WC, Tseng SS, Wang CY (2005) A novel manufacturing defect detection method using association rule mining techniques. J Expert Systems with Applications 29:807–815
5. Clare A, Williams HE, Lester N (2004) Scalable multi-relational association mining. In: Proc. of the Fourth IEEE Int. Conf. on Data Mining (ICDM'04). Brighton, UK, pp. 355–358
6. Coenen F, Leng P, Ahmed S (2004) Data structure for association rule mining: t-trees and p-trees. IEEE Transactions on Knowledge and Data Engineering 16(6):774–778
7. Do TD, Hui SC, Fong AC (2005) Mining class association rules with artificial immune system. Lecture Notes in Computer Science 3684:94–100
8. Doddi S, Marathe A, Ravi SS, Torney DC (2001) Discovery of association rules in medical data. Medical Informatics and the Internet in Medicine 26:25–33
9. Feldman R, Aumann Y, Amir A, Zilberstein A, Kloesgen W (1997) Maximal association rules: a new tool for mining for keywords cooccurrences in document collections. In: Proc. of the Third Int. Conf. on Knowledge Discovery (KDD'1997). Newport Beach, CA, pp. 167–170
10. Feldman R, Fresko M, Kinar Y, Lindell Y, Liphostat O, Rajman M, Schler Y, Zamir O (1998) Text mining at the term level. In: Proc. of the 2nd European Symposium on Knowledge Discovery in Databases. Lecture Notes in Computer Science 1510:65–73
11. Guan JW, Bell DA, Liu DY (2003) The rough set approach to Association Rule Mining. In: Proc. of the Third IEEE International Conference on Data Mining (ICDM'03). Melbourne, Florida, USA, pp. 529–532

12. Guo JY (2003) Rough set-based approach to data mining. Ph.D. thesis, Case Western Reserve University
13. Han J, Kamber M (2001) Data Mining Concepts and Techniques. Academic Press, San Diego, USA
14. Hong TP, Lin KY, Wang SL (2003) Fuzzy data mining for interesting generalized association rules. J Fuzzy Sets and Systems 138:255–269
15. Huang Z, Hu YQ (2003) Applying AI technology and Rough Set Theory to mine association rules for supporting knowledge management. In: Proc. Of Second Int. Conf. on Machine Learning and Cybernetics. 3:1820–1825
16. Hu XG, Wang DX, Liu XP, Guo J, Wang H (2004) The analysis on model of association rules mining based on concept lattice and apriori algorithm. In: Proc. of the Third International Conference on Machine Learning and Cybernetics in Shangai, pp. 1620–1624
17. Iváncsy R, Juhász S, Kovács F (2004) Performance prediction for association rule mining algorithms. In: Proc. of the IEEE International Conference on Cybernetics, Vienna, Austria, pp. 265–271
18. Ivkovic S, Yearwood J, Stranieri A (2002) Discovering interesting association rules from legal databases. Information and Communications Technology Law 11:35–47
19. Le SC, Huang MJ (2002) Applying AI technology and rough set theory for mining association rules to support crime management and fire-fighting resources allocation. J Information, Technology and Society. 2002(2):65–78
20. Lee HS (2005) Incremental association mining based on maximal itemsets. Lecture Notes in Computer Science 3681:365–371
21. Li C, Yang M (2004) Association rules data mining in manufacturing information System based on Genetic Algorithms. In: Third Int. Conf. On Computational Electromagnetics and Its Applications Proceedings. Beijing, China, pp. 153–156
22. Liu XW, He PL (2004) The research of improved association rules mining apriori algorithm. In: Proc. of the Third International Conference on Machine Learning and Cybernetics in Shangai, pp. 1577–1579
23. Loo KK, Lap YC, Kao B, Cheung D (2000) Exploiting the duality of maximal frequent itemsets and minimal infrequent itemsets for i/o efficient association rule mining. Lecture Notes in Computer Science 1873:710–719
24. Ma X, Ma J (2004) Rough set model for discovering single-dimensional and multidimensional association rules. IEEE International conference on Systems, Man and Cybernetics 4:3531–3536
25. Pawlak Z (1982) Rough sets. International J Cooperative Information Systems 1:335–376
26. Polkowski L, Skowron A (1998) Rough Sets in Knowledge Discovery Part I. Physica-Verlag, New York, USA
27. Saggar M, Agrawal AK, Lad A (2004) Optimization of association rule mining using improved genetic algorithms. IEEE Int. Conf. On Systems, Man and Cybernetics 4:3725–3729
28. Sartipi K, Kontogiannis K (2001) Component clustering based on maximal association. In: Proc. Working Conf. on Reverse Engineering (WCRE'01). Suttgart, Germany, pp. 103–114
29. Satoh K, Uno T (2003) Enumerating maximal frequent sets using irredundant dualization. Lecture Notes in Computer Science 2843:256–268
30. Techapichetvanich K, Datta A (2004) Visual mining of market basket association rules. Lecture Notes In Computer Science 3046:479–488
31. Ülker E, Sıramkaya E, Çomak E, Uğuz H, Arslan A (2005) Finding Regular Association Rules by applying the apriori algorithm. In: Proc. Of 4th International Advanced Technologies Symposium (IATS). Konya, Turkey, pp. 59–62
32. Vincenti G, Hammell RJ, Trajkovski G (2005) Data mining for imprecise temporal associations. In: Proc. Of sixth Int. Conf. On Software Engineering, AI, Networking and Parallel/Distributed Computing (SNPD/SAWN'05). Maryland, USA, pp. 76–81
33. Wang QD, Wang XJ, Wang XP (2002) Variable precision rough set model based dataset partition and association rule mining. In: Proc. Of First Int. Conf. On Machine learning and Cybernetics. Beijing, China, pp. 2175–2179

34. http://www.isikun.edu.tr/isam/teror/Graph_Teror.htm
35. Yan X, Zhang C, Zhang S (2005) ARMGA: Identifying interesting association rules with genetic algorithms. J Applied Artificial Intelligence 19:677–689
36. Zhang S, Lu J, Zhang C (2004) A fuzzy logic based method to acquire user threshold of minimum-support for mining association rules. J Information Sciences 164:1–16

Chapter 19
Business Logic Description by End-user for Web Service Integration Based on Abstract Forms

Takeshi Chusho, Ryousuke Yuasa, Shinpei Nishida, and Katsuya Fujiwara

19.1 Introduction

The number of end-users using the Internet inside and outside of the office has been increasing. As a result, the number of Web applications that end-users use has also been increasing. Most of these applications are developed by IT professionals. Thus, the work to be automated is limited to particular tasks such as electronic commerce relating to B-to-B and B-to-C, for example, which calculates profit over the cost of development. Furthermore, it is difficult to develop and maintain applications quickly. Primarily, Web applications should be supported by domain experts themselves because Web applications must be modified frequently based on the domain experts' ideas.

Therefore, end-user initiative development of applications has become important for the automation of end-users' own tasks. In the near future, the information society will require such new technologies, empowering domain experts and enabling them to automate their own work independently without extra training or the help of others.

Recently in the business world, the external specifications of application software have been considered as services. The keyword of software as a service (SaaS) becomes familiar. Some infrastructure technologies such as SOAP [15], UDDI [12] and WSDL [16] are used for rapid Web service provision [9] [11]. For example, Amazon [1] and Yahoo [18] provide Web service interfaces.

As for UDDI, problems concerning security, trust, and quality etc. must be resolved for practical use in e-marketplaces. Therefore, such infrastructures may pervade from local or private areas such as private UDDI to global areas such as e-marketplaces.

Takeshi Chusho, Ryousuke Yuasa, Shinpei Nishida
Meiji University, Kawasaki, Japan
e-mail: chusho@cs.meiji.ac.jp

Katsuya Fujiwara
Akita University, Akita, Japan
e-mail: fujiwara@ie.akita-u.ac.jp

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

As a solution based on component-based software engineering (CBSE) [2] and service-oriented architecture (SOA), this paper describes a form-based approach to Web service integration [14] for end-user computing, because end-users consider their applications as a level of service, not as a level of software. That is, the service counter is considered as a metaphor to describe the interface between service providers and their clients for Web services, and is designed based on the simple concept that *one service = one form*. This concept provides form-based interfaces enabling Web service integration to be defined as the form transformation from input forms into output forms. In addition, our approach will be applied—primarily in local areas such as within a group of trusted organizations—at an experimental stage of the end-user initiative development.

There are other works related to the end-user initiative development. In the database field, the example-based database query languages [13] such as query-by-example (QBE) were studied. User-friendly inquiry languages were proposed in comparison with SQL. In the programming field, the technologies for programming by example [10] were studied. The nonprogramming styles for various users, including children, and for various domains, including games, were proposed. Our research target is business experts and business domains.

This chapter presents the end-user initiative approach in Sect. 36.2, technical issues for automatic integration of these services in Sect. 36.3 and our solutions in Sect. 36.4, Sect. 19.5 and Section 19.6.

19.2 The End-user Initiative Approach

19.2.1 Basic Concepts for Web Services

For Web applications supporting Web services, the following two features are considered to be essential:

1) Rapid development and continuous maintenance.
2) End-user initiative.

Business based on Internet technology is rapidly changing. For this reason, the application development period from defining a new business model through releasing new services must be kept short. If it is not, the business chance will be lost. Furthermore, after the release, the application should be maintained continuously because the business world changes. Conventional ways for application development and maintenance by system engineers are not suitable because of the lack of timeliness.

Our approach to how to make Web applications supporting Web services is shown in Fig. 19.1 [5]. The three layers of the left zone and the central zone imply the abstract level and the concrete level, respectively. The right zone implies technologies. The business model at the business level is proposed by end-users of domain experts. Then, at the service level, the domain model is constructed and

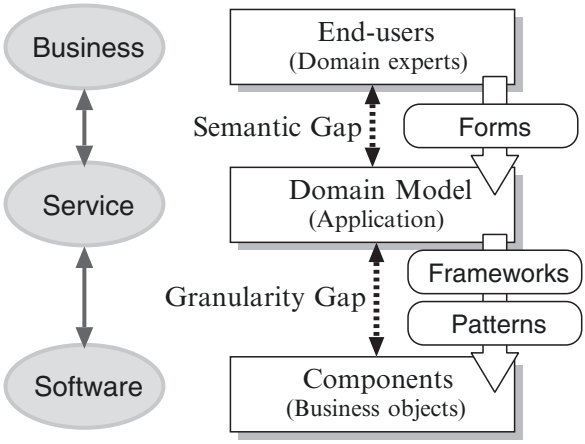


Fig. 19.1 Technologies for bridging gaps between end-users and components

the required services are specified. That is, the requirement specifications of the application for the business model are defined. At the software level, the domain model is implemented by using components to be combined.

In this approach, there are two technological gaps: the granularity gap between components and the domain model, and the semantic gap between the domain model and end-users. The granularity gap is bridged by business objects, patterns [8] and application frameworks [7] based on CBSE. Conversely, the semantic gap is bridged by form-based technologies.

As for the granularity gap, our previous studies verified the effectiveness of application framework technologies by development of service counter frameworks.

Concerning the semantic gap, this gap was bridged by agent technologies detailed in our previous studies [4]. That is, an intelligent form with an agent that learns business logic, was introduced. However, in our recent studies on end-user initiative development, it is considered that a simple form is better for end-users.

19.2.2 Metaphors for Web Services

We direct our attention to an application system for a service counter. Such a service counter is not limited to the actual service counter in the real world. For example, in a supply chain management system (SCM), exchanges of data among related applications can be considered data exchanges at the virtual service counter.

Of course, many kinds of service counters have already been put to practical use on the Internet and in intranets. However, these systems must have been developed by IT professionals, not by end-users, and are expensive. Furthermore, although the domain experts require frequent modification of specifications in these systems for service upgrades, it is difficult to modify software quickly because the

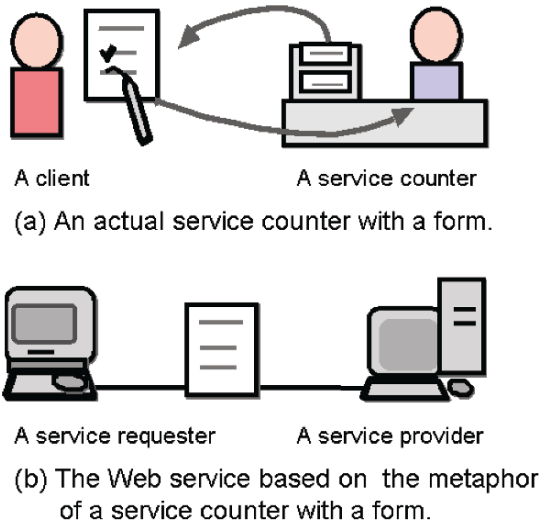


Fig. 19.2 A service counter as a metaphor for Web service

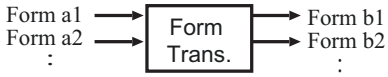


Fig. 19.3 The form transformation for Web service integration

domain experts do not maintain the systems themselves and need to rely on the IT professionals instead. Therefore, our goal is the development and maintenance of form-based applications by end-users.

Generally, the service counter receives service requests from clients to service providers as shown in Fig. 19.2. Forms to be filled in are considered as the interface between them. That is, the following concept is essential for our approach:

One service = One form.

The integration of some individual Web services is considered as transformation from some input forms into some output forms as shown in Fig. 19.3 with the goal being that domain experts make an application by form transformation.

In this case, most of these forms are not visual forms, but abstract forms in XML. Because end-users consider such Web service integration as work flow with visual forms that they are familiar with, IT skills are not required of end-users.

For example, in making reservations for travel and accommodations, when one service provider receives the input form for reservations for both a hotel room and a flight, the provider transforms the input into the first output form for reservation of the hotel room and the second output form for reservation of the flight. Then, the provider sends the first output form to the service provider providing hotel room reservations and the second output form to the service provider providing flight reservations, respectively. When the provider receives both replies to these output forms, that is, the first input form on available hotel rooms and the second input form on available seats on a flight, the provider transforms these two input forms into the

output form which is a combination of available hotel rooms and available seats on a flight. Finally, this output form is returned as the reply to the initial request. In Web service integration, forms that are used as the interface between service providers and those who request service, are abstract forms in XML.

19.2.3 A Shift from Message Flow to Form Flow

Our previous model-driven solution for end-user initiative application development was based on an object-oriented model in which one task in a domain model of cooperative work corresponds to one object in a computation model. This solution requires, out of necessity, fixed architecture and ready-made components such as business objects for component-based development. The application development environment, M-base, was developed for supporting this solution [3]. At the first stage in the process, the system behavior is expressed as a message-driven model by using a visual modeling tool while focusing on message flow and components.

However, because end-users are not familiar with object-oriented technologies, practical development processes must be provided based on metaphors of an office. That is, cooperative work at an office is expressed by using a form-flow model. Thus, our approach to end-user initiative development has shifted from a message-flow base to a form-flow base.

19.3 Technical Issues

Let us suppose that we are planning an itinerary by using reservation services for both hotel accommodations and flight booking via the Internet, as shown in Fig. 19.4. We usually visit the website of a hotel and the website of an airline

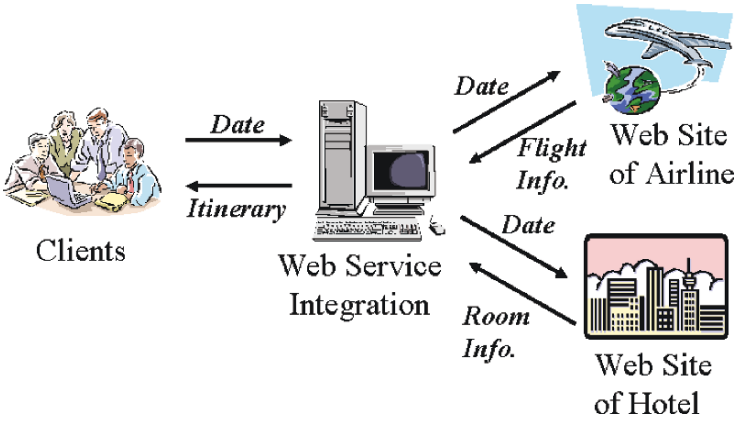


Fig. 19.4 An example of Web service integration

company separately, and then book a hotel room and a flight seat independently of each other.

There are two technical issues for integration of these services. One is how to communicate with conventional Web applications that support HTML documents as user interfaces. Two solutions are considered for this problem. One is that a front-end part of the Web application could be newly developed in accordance with the XML-based Web service specifications. This solution is not realistic, however because many companies will not do so until the XML-based Web service pervades in the Internet world. The other, and more practical, solution is Web-page wrapping in which the HTML documents are transformed automatically into XML documents.

The other issue is how to merge two outputs in XML. The reasonable combination of information pertaining to an available hotel room and an available seat on a flight should be derived from both the hotel room information in XML and the flight information in XML.

19.4 The Basic XML Merger

19.4.1 Target Application

For applying our solutions to practical Web service integration, we select an application that generates individual examination schedules for each student [6]. In our university, actually, the examination schedule is posted on bulletin boards. Conversely, the university supports individual portal sites for each student. The student visits the portal site that displays the individual timetable for classes.

Therefore, we foresee that the XML-based Web service for the examination schedule will be realized in the near future. In our experiment, an actual examination schedule is transformed into an XML document manually. As for the individual timetable for classes, an actual HTML document is extracted from the individual portal site for each student.

The target application generates an individual examination schedule for each student from the individual timetable for classes and the examination schedule, as shown in Fig. 19.5.

19.4.2 System Architecture

The system architecture is shown in Fig. 19.6. The input of the subsystem, WS-A, is the individual timetable for classes. This is an actual HTML document that is extracted from the individual portal site for each student. This document includes information about subjects for each student—that is, subject names, instructor names,

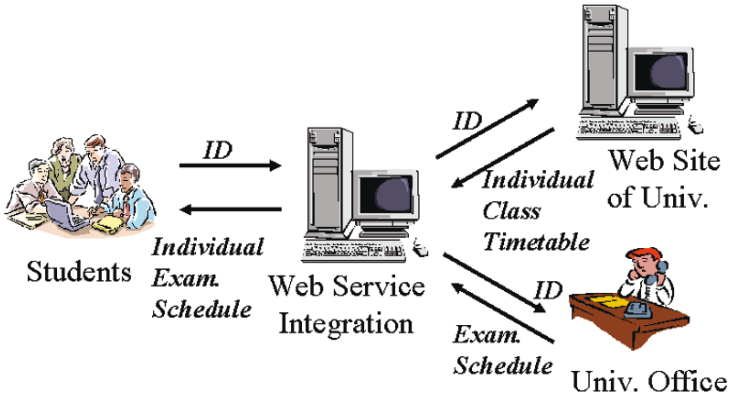


Fig. 19.5 Web service integration for an individual exam schedule

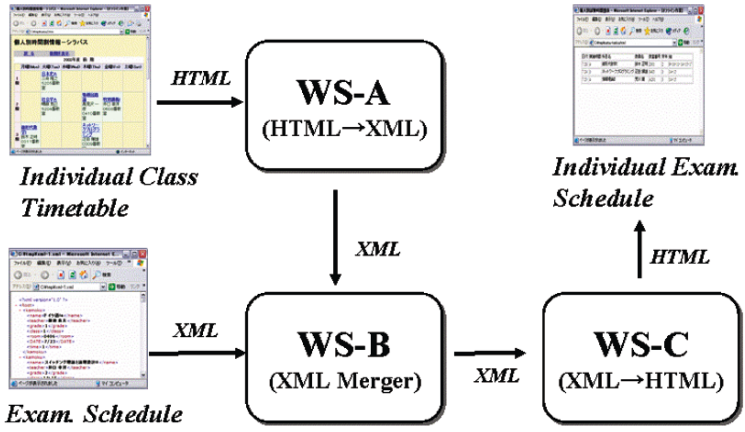


Fig. 19.6 System configuration for Web service integration

and class numbers. The WS-A transforms this HTML document into an XML document by using the wrapping technology mentioned previously.

The inputs of the subsystem WS-B are two XML documents. One is the individual timetable for classes for each student in XML. The other is the examination schedule in XML that includes information about subject names, dates and periods, and room numbers. The WS-B merges these two XML documents into the individual examination schedule in XML format for each student.

The input of the subsystem WS-C is the individual examination schedule in XML. The WS-C transforms this XML document into an HTML document that can be displayed on the Web browser for each student.

19.5 Scripting Business Logic

19.5.1 End-User Support

The system administrator of this application is not an IT professional, but a clerk in the university office. Such an end-user does not have the ability to perform programming, but needs only to modify the system when the inputs change. For the solution to this problem, basically, the procedure of this application is described in a script language. Furthermore, a visual tool supports the end-user.

19.5.2 HTML-to-XML Transformation

The WS-A subsystem transforms the HTML document into an XML document. The input is the individual timetable for classes in HTML. The following XML document is generated from the information about subject names and instructor names that are extracted from this HTML document.

```
<?xml version="1.0" encoding="UTF-8"?>
<personaltimetable>
 <subject>
  <name>Software Engineering</name>
  <instructor>Txx Cxx</instructor>
 </subject>
</personaltimetable>
```

This procedure is composed of two steps for use of XSL Transformations (XSLT) [17]. In the first step, the HTML document is transformed into the XHTML document because XSLT cannot accept HTML documents, but can accept XHTML documents. In the next step, XSLT is used to transform this XHTML document into the XML document by using the following XSLT stylesheet:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl=
"http://www.w3.org/1999/XSL/Transform">
 <xsl:output encoding="UTF-8" method="xml"/>
 <xsl:template match="/">
  <personaltimetable>
   <xsl:apply-templates
    select="/html[1]/body[1]/table[3]/tbody[1]/tr/td/a[1]/b[1]"/>
  </personaltimetable>
 </xsl:template>
 <xsl:template match="/html[1]/body[1]/
table[3]/tbody[1]/tr/td/a[1]/b[1]">
  <subject>
   <name>
    <xsl:value-of select="/text()"/>
   </name>
   <instructor>
```

Fig. 19.7 A part of the timetable embedded with checkboxes (in Japanese)

```
    <xsl:value-of select="../../font[1]/text()"/>
   </instructor>
  </subject>
 </xsl:template>
</xsl:stylesheet>
```

In this procedure, only the XSLT stylesheet is dependent on the individual application, and must be described by the end-users. The other parts are automated. It may be difficult, however, for the end-users to describe the XSLT stylesheet yet, although this scripting is comparatively easer than programming. This is because the end-user must understand the concepts of XPath and the template.

We developed a visual tool that generates the XSLT stylesheet. This tool is used in four steps. In the first step, the file name of the HTML document is input into the text input area that is displayed by the visual tool. In the second step, the structure of the XML document to be output is defined in the text input area under guidance with examples that are displayed by the visual tool. Alternatively, the user can include the predefined file. In the third step, a class timetable is displayed. A checkbox is located in front of items such as subject name, instructor name, or room number in this timetable (as shown in Fig. 19.7) although the original timetable does not have such checkboxes. The user selects data to be extracted from the HTML document by checking the checkboxes. At the last step, XPath is identified for selecting elements to be processed, and the XSLT stylesheet is generated by entering the output file name.

19.5.3 XML Document Merging

The WS-B subsystem merges two XML documents—the individual timetable for classes for each student and the examination schedule—into the individual examination schedule in XML for each student. The following XML document is a part of the examination schedule:

```
<?xml version="1.0">
<examine>
 <subject>
  <name>Software Engineering</name>
  <instructor>Txx Cxx</instructor>
  <grade>3</grade>
  <class>14 ~15</class>
  <room>0405</room>
  <date>7/24</date>
  <time>3</time>
 </subject>
 <subject>
   ....
   ....
 </subject>
</examine>
```

This subsystem generates the XML document as the output by extracting classes that are included in both input files. This procedure is composed of the following four steps:

1. Assign the nodes of two input XML files into variables.
2. Extract the element to be compared from each input file by using the variables for counting.
3. Compare two elements.
4. Output the element in the specified format if these two elements are the same.

For these processes, the XSLT stylesheet is used. XSLT is a language for transforming XML documents into other XML documents. Furthermore, XSLT makes use of the expression language defined by XPath for selecting elements for processing for conditional processing, and for generating text. In the XSLT stylesheet, a document function and a position function are used. The document function allows access to XML documents other than the main source document. The input files, xml-1.xml and xml-2.xml, are assigned to the variables *first* and *second*, respectively, as follows:

```
<xsl:variable name="first" select="document('xml-1.xml')" />
<xsl:variable name="second" select="document('xml-2.xml')" />
```

The position function returns the position of one of the child elements that have the same parent node. The current position of one of the subject elements that are included in the xml-1.xml file is assigned to the variable *firstposition*. The one included in the xml-2.xml file, is assigned to the variable *secondposition*. Then, the class names and the instructor names of two input files are compared as follows:

```
<xsl:for-each select="$first/examine[1]/subject">
 <xsl:variable name="firstposition" select="position()" />
 <xsl:for-each select="$second/personaltimetable[1]/subject">
  <xsl:variable name="secondposition" select="position()" />
   <xsl:if test="$first/examine[1]/subject[$firstposition]/name[1]
= $second/personaltimetable[1]/subject[$secondposition]/name[1]">
   <xsl:if test="$first/examine[1]/subject[$firstposition]/
```

```
instructor[1]= $second/personaltimetable[1]/
subject[$secondposition]/instructor[1]">
   ....
```

In this procedure, the XSLT stylesheet is dependent upon the individual application, and must be described by the end-users. The other parts are automated. It may be difficult, however, for the end-users to describe the XSLT stylesheet, although this scripting is rather easy compared to programming. This is because the end-user must understand the concepts of iterative processes with variables.

We developed a visual tool that generates the XSLT stylesheet as shown in Fig. 19.8: This tool is used in six steps. In the first step, file names are input into the text input areas. In the second step, two input XML documents, are transformed into HTML documents, in which the checkboxes are located at the front of each node and are displayed. The user selects the parent node of the element to be compared. Next, the user selects the elements to be compared. In this application, the subject is selected first, and then the subject name and the instructor name are selected. In the third step, the selected elements are displayed as follows:

```
F0  /examine[1]/subject[n]/name[1]
F1  /examine[1]/subject[n]/instructor[1]
S0  /personaltimetable[1]/subject[n]/name[1]
S1  /personaltimetable[1]/subject[n]/instructor[1]
```

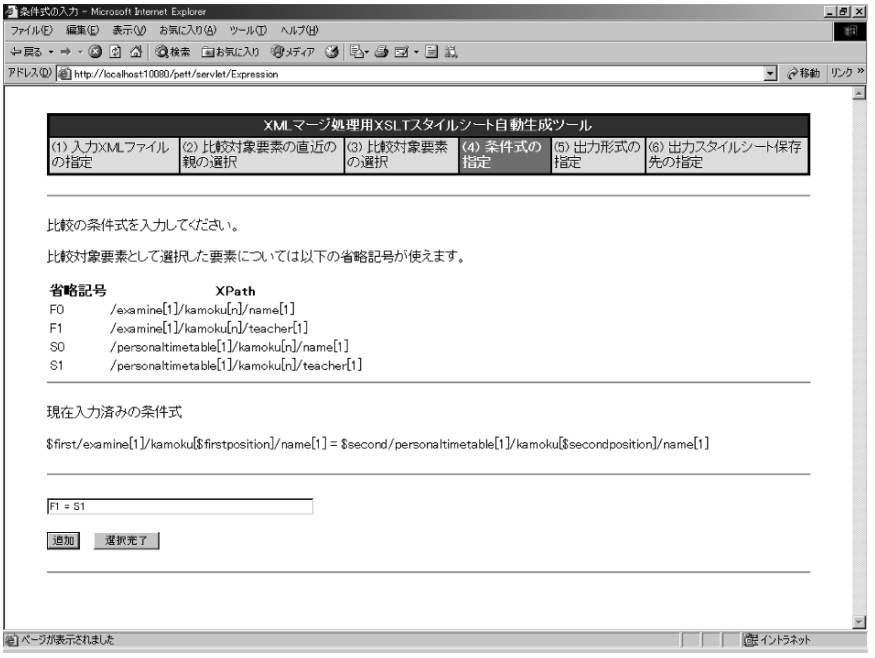


Fig. 19.8 A visual tool for generating the XSLT style sheet. (in Japanese)

F0, F1, S0, and S1 are symbols for the corresponding XPaths. The user can define conditions for the comparison by using these symbols as follows:

```
F0 = S0
F1 = S1
```

The remaining steps are similar to the second, third and fourth steps in the WS-A subsystem.

19.5.4 Generality of the Merging Method

This process can be applied to other general domains. For example, let us consider that a client wants to acquire a concert ticket from company C, a flight ticket from airline company A and to reserve a room at hotel H during a specified period, by using the system with Web service integration of these three companies.

In this case, there are the following four input abstract forms (AFs): AFin1 with the specified period, AFin2 with information about the concert ticket reservation from company C, AFin3 with information about the flight reservation from airline company A, and AFin4 with information about the room reservation from hotel H.

The basic merger is executed repeatedly for multiple inputs. That is, the merging processes shown in Fig. 19.9, are required for the client request. In the first step, a list of dates with the available concert ticket is extracted by merging the dates of AFin1 and the dates with available tickets of AFin2, that are then is output as AF-1. In the second step, a list of dates with both the available concert ticket and the available flight ticket, is extracted by merging the dates of the AF-1 and the dates with available flight tickets of AFin3, and then output as AF-2. In the last step, a list of dates in conjunction with the client's request is extracted by merging the dates of AF-2 and the dates with available rooms of AFin4, and then outputs as the AFout.

Because these merging processes are the same as the process for generating an individual examination schedule, the XSLT stylesheet can be generated by using our visual tool.

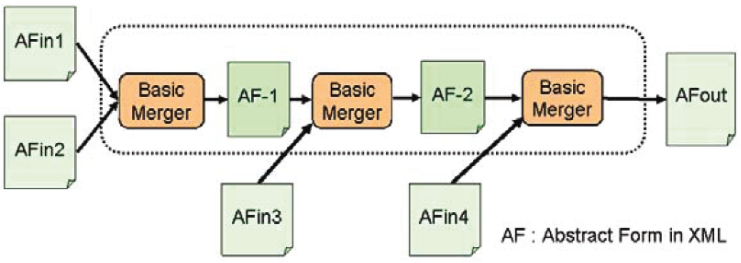


Fig. 19.9 Repetition of the basic merger for multiple inputs

19.5.5 XML-to-HTML Transformation

The WS-C subsystem transforms the XML document into an HTML document. There are some conventional tools used for this transformation. The XSLT stylesheet for this application is generated by using one of conventional tools.

19.6 Multistage XML Merger

The basic merger transforms input abstract forms in XML into output abstract forms in XML with simple business logic. Next, this merger is extended for dealing with complicated business logic. As a sample application, the advisory system for graduation is chosen. This system advises a student to take specific subjects necessary for graduation from inputs of both his/her school report and a manual on graduation requirements. The school report shows a list of subjects and the units that the student has already completed. The manual shows conditions for graduation, are considered complicated business rules. That is, subjects are categorized into several groups that compose a hierarchy, and each category has individual conditions.

For dealing with such complicated business logic, the multistage merger is introduced as shown in Fig. 19.10. The previous basic merger is considered a special case in this multistage merger that the output of the first stage, AF-1, is the final output, AFout. In the multistage merger, generally, the intermediate output AF-k is transformed into the next intermediate output AF-(k+1).

In the multistage merger, four kinds of templates for the root element, the subject category, compulsory subjects, and semi-compulsory subjects are introduced. For example, the template for compulsory subjects is used at the second stage. Compulsory subjects have four kinds of attributes. The attribute of *requiredunit* is the required number of units. The attribute of *totalunits* is the number of units which the student has already completed. The attribute of *comparison* is the condition of comparison of the number of completed units with the required number of units. The attribute of *full* is the state of satisfaction of conditions.

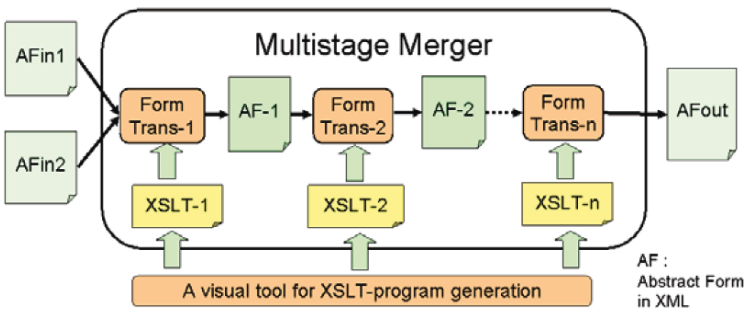


Fig. 19.10 A configuration of the multistage merger for complicated business logic

In the template for compulsory subjects, the *requiredunits* and *comparison* in AF-1 are output into AF-2 without modification. The *totalunits* is calculated by using Xpath with the sum function and the current function as follows:

```
   <xsl:attribute name="totalunits">
    <xsl:value-of select="sum(current()/
subject[@done = &quot;true&quot;]/@units)"/>
   </xsl:attribute>
```

The *full* attribute is calculated by using Xpath with the *not* function as follows:

```
   <xsl:attribute name="full">
    <xsl:value-of select="not(current()/
subject[@done = &quot;false&quot;])"/>
   </xsl:attribute>
```

Finally, all data on elements concerning compulsory subjects in AF-1 are output into AF-2.

In this application, the seven stages were necessary for the confirmation of conditions required for graduation.

19.7 Conclusions

The form-based approach for Web services integration by end-user initiative application development was proposed. Our experiments with prototyping were herein described. For communication with conventional Web applications, the Web page wrapping tool for HTML-to-XML transformation was developed. For merging XML-based Web services, the basic merger for simple business logic and the multistage merger for complicated logic were developed. In these methods, application-specific processes are described in XSLT stylesheets with visual tools, and application-independent processes are generated automatically.

References

1. Amazon (2006) Web Services, *http://www.amazon.co.jp/*
2. Brown AW (Ed) (1996) Component-based software engineering, IEEE CS Press
3. Chusho T., Ishigure H., Konda N. and Iwata T. (2000) Component-Based Application Development on Architecture of a Model, UI and Components, *APSEC2000*, IEEE Computer Society, pp. 349–353
4. Chusho T and Fujiwara K, (2000) FACL : A Form-based Agent Communication Language for Enduser-Initiative Agent-Based Application Development, *COMPSAC2000*, IEEE Computer Society, pp. 139–148
5. Chusho T, Fujiwara K, Ishigure H, and Shimsda K, (2002) A Form-based Approach for Web Services by Enduser-Initiative Application Development, *SAINT2002 Workshop (Web Service Engineering)*, IEEE Computer Society, pp. 196–203

6. Chusho T, Yuasa R, Nishida S and Fujiwara (2006) A Form-based Approach for Application Development By Web Service Integration, *Applied Computing 2006*, IADIS, pp. 600–605
7. Fayad M and Schmidt D C (Ed) (1997) Object-Oriented Application Frameworks, *Commun. ACM*, V39, N10, pp. 32–87
8. Gamma E, Helm R, Johnson R and Vlissides J (1995) *Design Patterns*, Addison-Wesley
9. Gold N, Mohan A, Knight C, and Munro M (2004) Understanding Service-Oriented Software, *IEEE Software*, V21, N2, pp. 71–77
10. Lieberman H. (Ed) (2000) Special issue on Programming by example, *Comm. ACM*, V43, N3, pp. 72–114
11. Malloy B A, Kraft N A, Hallstrom J O and Voas J M (2006) Improving the Predictable Assembly of Service-Oriented Architectures, *IEEE Software*, V23, N2, pp. 12–15
12. OASIS (2004) Advancing Web Services Discovery Standard, *http://www.uddi.org/*
13. Ozsoyoglu G and Wang H (1993) Example-Based Graphical Database Query Languages, *IEEE Computer*, V26, N5, pp. 25–38
14. Peltz C (2003) Web Services Orchestration and Choreography, *IEEE Computer*, V36, N10, pp. 46–52
15. W3C (2003) Latest SOAP versions, *http://www.w3.org/TR/soap/*
16. W3C (2006) Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, *http://www.w3.org/TR/2006/CR-wsdl20-primer-20060327/*
17. W3C (2006) The Extensible Stylesheet Language Family (XSL), *http://www.w3.org/Style/XSL/*
18. Yahoo-Japan, (2006) Yahoo!Developer Network, *http://developer.yahoo.co.jp/*

Chapter 20
OPTIMISE: An Internet-Based Platform for Metamodel-Assisted Simulation Optimization

Amos Ng, Henrik Grimm, Thomas Lezama, Anna Persson, Marcus Andersson, and Mats Jägstam

20.1 Introduction

Computer simulation has been described as the most effective tool for de-signing and analyzing systems in general and discrete-event systems (e.g., production or logistic systems) in particular (De Vin et al. 2004). Historically, the main disadvantage of simulation is that it was not a real optimization tool. Recently, research efforts have been focused on integrating metaheuristic algorithms, such as genetic algorithms (GA) with simulation software so that "optimal" or close to optimal solutions can be found automatically. An optimal solution here means the setting of a set of controllable design variables (also known as decision variables) that can minimize or maximize an objective function. This approach is called simulation optimization or simulation-based optimization (SBO), which is perhaps the most important new simulation technology in the last few years (Law and McComas 2002). In contrast to other optimization problems, it is assumed that the objective function in an SBO problem cannot be evaluated analytically but have to be estimated through deterministic/stochastic simulation.

Despite reports that SBO has been successfully applied to address a wide range of industrial problems (April et al. 2004), the full potential benefits of SBO and related tools are not yet explored or understood on the part of prospective users (Andradóttir et al. 2000). It has been observed that such a lack of understanding may be stemmed from two major barriers:

1. So far, despite the popularity of powerful simulation modeling software and the advent of robust commercial off-the-shelf optimization packages, the knowledge and expertise to integrate them together to run optimization are still not trivial. Managers who take decision-making responsibility seldom possess the time/skills to run experiments and optimization on a simulation model directly in order to draw significant conclusions for making important decisions. It is a

Amos Ng, Henrik Grimm, Thomas Lezama, Anna Persson, Marcus Andersson, and Mats Jägstam
Centre for Intelligent Automation, University of Skövde, Sweden

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

common practice to hire other simulation specialists in the company or external consultant firms to carry out the experiments and formally report the results. From our experience, this would seldom complete in a single round. In many cases, other than costing much higher overhead, the long lead times involved may prohibit the interest in using simulation in general or SBO in particular for decision-making support.
2. Running SBO with a simulation model of significant size may require unacceptably long computing time, especially if the result is used for making some "real-time" decisions, e.g., daily operations planning. A simulation model built for an ordinary automotive machining/assembly factory with significantly detailed level may take minutes or even hours for one simulation replication. This is not justified if the optimal result is used for adjusting operations planning or scheduling on a weekly basis, let alone daily.

To address the above-said barriers, a highly flexible Web-based parallel SBO platform called OPTIMization using Intelligent Simulation and Experimentation (OPTIMISE) has been developed. The main characteristic of this platform is that all complex components, including various metaheuristic search algorithms, neural-network-based metamodels, deterministic/stochastic simulation systems and the corresponding database management system are integrated in a parallel and distributed platform and made available for multiple users to access with ease through eXtended Markup Languages (XML) and Web Services technology. There are two long-term goals of this research: (1) to promote and broaden the use of simulation in general and SBO in particular in industry; (2) to facilitate further research on metamodel-assisted SBO using hybrid search methods for real-time decision making and/or weekly/daily operations planning.

The rest of this chapter is organized as follows: Section 2 covers a literature review on related work, mainly in the area of Web-based parallel SBO that are highly relevant to this research effort. Section 3 details the multi-tier client/server architecture of OPTIMISE and its current implementation. Section 4 discusses the use of artificial neural network (ANN) as the metamodeling method. Sections 5 and 6 illustrate how OPTIMISE has been used to address a multi-objective and a single-objective optimization problem found in manufacturing respectively. Conclusions and future work are given in Sect. 7.

20.2 Literature Review

Parallel and distributed simulation (PADS) represents the computing technology that enables a simulation program to execute on a computing platform containing multiple processors, interconnected by a communication network. It can be used to reduce execution time and/or addressing problems like geographical distribution (e.g., multiple participants), heterogeneous simulators from different manufacturers and fault tolerance (Fujimoto 2000). In recent years, the ability to connect multiple distributed simulation models/sub-models into a larger, complex simulation

has gained more attention from domains like military, telecommunication and education. Nevertheless, in many simulation applications, the primary benefit offered by PADS is the execution speedup of running many replications on parallel processors. Early work can be found in Biles et al. (1985) in which different computer architectures for carrying out a large number of simulation replications in a parallel computing environment were examined. Subsequent work was done by Heidelberger (1988), who proposed a parallel replications environment equipped with more advanced statistical methods for supporting replication analysis. In this approach, several replications of a sequential simulation are run to completion independently on different processors. In the jargon of parallel computing, these kinds of applications belong to the so-called *embarrassingly parallel problem* because no particular effort is needed to segment the problem into a number of parallel tasks, and no essential communication between those parallel tasks is required[1]. Embarrassingly parallel problems are ideally suited to large-scale problems over the Internet. Well-known public applications include climatepredication.net (Stainforth et al. 2002) and SETI[2].

With the advent of Internet technologies, many efforts in PADS have been made for developing simulation languages and building model libraries that can be assembled and executed over the Internet (Yoo et al. 2006). In that regard, Web-based parallel and distributed simulation (WPADS) is understood as the research area where PADS methodologies and Web-based technologies are conjoined. As noticed by zu Eissen and Stein (2006), the term Web-based simulation is used collectively for describing various applications and may have very different meanings. However, in general, it refers to the use of Web-based technologies that enable users to remotely access and execute simulation languages and simulation programs over the Web.

Many similar efforts have been made in Web-based simulation. A Java-based Simulation Manager (*SimManager*) is described in (Marr et al. 2000; Biles and Kleijnen 2005) that is essentially a parallel-replication approach in which the *SimManager* identifies and controls a number of slave processors (simulation engines) that run the actual simulation trials. Through the concept of Alliance, computer owners can make their processors "available" to the *SimManager* by entering into an agreement to participate in a simulation consortium dynamically. A simulation engine is run as a low-priority "background" task if the slave processor is used as for some other application in the front end. This bears similarity with other public WPADS system such as climateprediction.net. Later, their work has extended to integrate commercial simulation packages such as Arena, applied to compare alternative material handling configurations for automated manufacturing (Biles and Casebier 2004).

Kumara et al. (2002) depicted a Web-based three-tier client/server framework for allowing multiple users to access a simulation program for evaluating and predicting order-to-delivery systems. The simulation program, developed in GM Enterprise

[1] See Wikipedia [online] http://en.wikipedia.org/wiki/Embarrassingly_parallel.

[2] SETI@home, Search for Extraterrestrial Intelligence [online]. Available: http://setiathome.ssl.berkeley.edu/.

Systems Laboratory (GMESL), was originally designed as a standalone program accessed in a single-user mode. The framework separates the functions of presentation, data management and analysis into three tiers: 1) Web client; 2) relational database server and 3) multi-agent based virtual executor server. Focus of their work was on the scalability and user responsiveness of the system enabled by the information model in the database server and the multi-agent execution model.

With Web services as the new enabling Internet technology, WPADS is now seen as a more viable simulation option than ever before and many researchers are aware of the benefits it can offer. There are many answers to the question "What is a Web service?". Within the context of this paper, we define a Web service as a remotely accessible application component that listens to and reacts for certain requests made over HTTP. When compared to all other standard object architecture for distributed applications (i.e. DCOM, Java and CORBA), Web services technology is the only one that truly enables heterogeneous platform interoperability. Different realization alternatives for Web-based simulation services are explained and discussed with respect to their advantages and disadvantages in (zu Eissen and Stein 2006). The authors also implemented a prototype Web service that allows the analysis and execution of technical models for DES, continuous time or hybrid simulation described in the modeling language *Modelica*. Their focus is on fast model building and quick experimentation using *Modelica* model libraries. Experience using Web-service based SBO through the distribution of simulation replications across different servers is presented by Yoo et al. (2006). Their focus is on using an *Optimal Computing Budget Allocation* (OCBA) to allocate different number of simulation replications to different servers to improve the overall execution efficiency.

20.3 OPTIMISE: An Internet-based SBO Environment

As shown in Fig. 20.1, at the heart of the OPTIMISE architecture is a number of optimization engines, surrounded by a set of OPTIMISE Server Components that spread across three tiers: (1) Web Server; (2) Optimization and (3) Simulation subsystem. In an SBO application supported by the OPTIMISE framework, the optimization engine (*OptEngine*) in the optimization tier is the most important component because it provides the core functionality for major algorithmic processing and acts as the hub for coordinating other functions, including data logging and metamodeling. Server components can be accessed by client applications through consuming the OPTIMISE Web services, hosted by the Web server as shown in Fig. 20.1. The Web server listens to the XML requests and acts accordingly. Most often the functionalities that client applications request are launching/controlling an SBO process (through *OptManager*) and retrieving data from *OptDB* (Optimization Database). We shall look into these in detail.

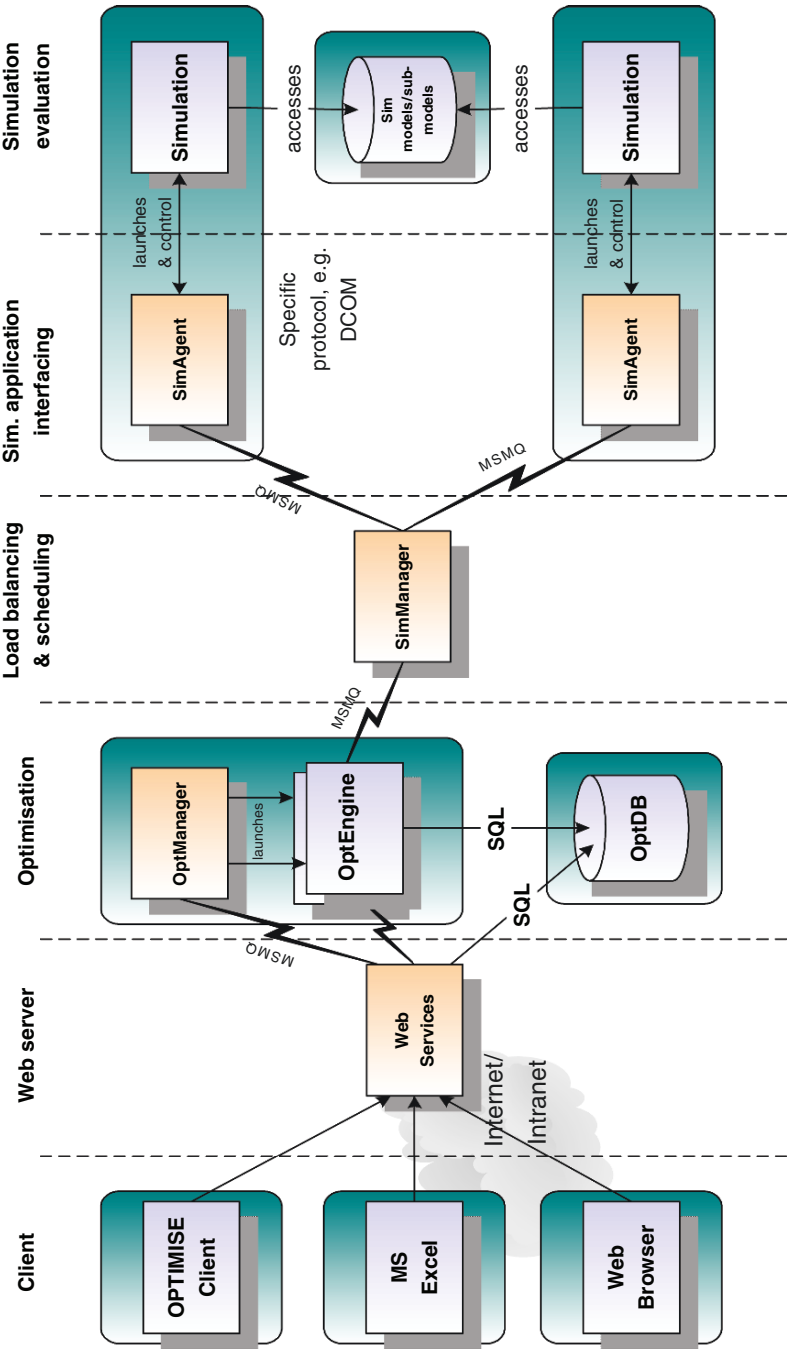


Fig. 20.1 At the heart of the OPTIMISE architecture is a number of optimization engines, surrounded by a set of OPTIMISE Server Components that spread across three tiers: (1) Web Server; (2) Optimization and (3) Simulation subsystem

20.3.1 OPTIMISE Server Components

The Optimization Manager (*OptManager*) is a Windows service that listens to the request from the Web Server to launch different *OptEngines* according to the preferences and parameters specified by the users through client applications. Data that is required to start an SBO procedure may include (1) simulation settings such as warm-up time, simulation time and number of replications, (2) objective function, (3) list of input variables, (4) list of output variables, (5) constraints to input variables, (6) which optimization algorithm, and (7) optimization parameters (e.g., population size, crossover rate). The last two options are very important for ordinary users as well as advanced researchers to select and compare the performance of different search methods with different parameters. Currently, OPTIMISE supports several versions of metamodel-assisted hill climbing and evolutionary algorithms but new algorithms can be added easily by compiling the modified algorithm core with the object-oriented libraries supplied by OPTIMISE. Generic algorithm software or templates deserve research on their own right (Voß and Woodruff 2000). In our case, we have developed a pragmatic OO class library that allows new algorithms to inherit or override class methods for selection, crossover and mutation operations, which are commonly used in any evolutionary algorithms. Auxiliary functions for training metamodels, data normalizations, and communication with other components are all available in the common libraries. These enable new algorithms to be rapidly developed or customized and fit into the OPTIMISE framework by reuse.

By allowing all active *OptEngines* to save their optimization trajectories and other experiment results into a central database, i.e., *OptDB*, OPTIMISE supports the following advanced features:

- The quality and diversity of the initial solutions play a crucial role in the performance of an optimization process, especially when a population-based search method is used. By saving all experimental results into *OptDB*, OPTIMISE enables a user to choose the set of initial solutions from previous experimental records when starting a new optimization process. This is usually used in combination with the Design of Experiments (DoE) functions provided in the OPTIMISE client applications.
- Dynamically changing the metaheuristic algorithms in an optimization process. This is especially useful when a global search method like GA is used for exploration first and then followed by a local search method for exploitation to further improve the optimization result. An industrial-based example is presented in Sect. 6.
- Fault tolerance—while faults in a simulation can be easily detected and recovered by restarting the run by a *SimAgent*, software faults occurring in an *OptEngine* may cause a single point of failure and waste the time spent for all previous simulation runs. OPTIMISE indirectly facilitates error recovery by allowing a

user to start an *OptEngine* and reload the previous simulation records saved in *OptDB* as the initial solutions and training data set for the metamodel when the search process is re-started.

20.3.2 Simulation Components

Simulation components are decoupled from the core server components because they can be highly distributed (across the Internet) to support parallel simulation runs to speed up the computationally expensive optimization and experimentation processes. The simulation tier may contain various types of deterministic or stochastic simulation systems that actually carry out the simulation runs. Heterogeneous simulation systems are connected to *SimManager* homogenously via *SimAgents* in the SimAgent tier. Depending on the application interfacing methods supported by the target simulation system, *SimAgent* can start the corresponding *BackEnd* object to launch, interact and control the simulation software. Currently there are a handful of *BackEnd* objects that support distributed component object model (DCOM) and socket communications for connecting to different simulation systems. Using the specific protocol, the *BackEnd* object can control the simulation software to load the right model, put input data and collect output data according to the specific format supported by the simulation software/model. The output data is then "assembled" and sent back in a standard format, in form of XML via *SimManager*, and return to the corresponding *OptEngine* for evaluation and data logging. Unlike the *SimManager* described in (Biles and Kleijnen 2005), which "must possess software for all of the statistical methodology and optimization techniques necessary to analyze the simulation results sent back to it by the slave processors," the *SimManager* developed for OPTIMISE is a generic and lightweight job dispatcher.

Depending on the number of processors and computing resources of a computer reserved for running simulation, more than one *SimAgent* can be started on a single node. When a *SimAgent* is launched, it will register to the *SimManager* to announce its existence. By knowing the number of available *SimAgents*, the *SimManager* can dispatch multiple jobs received from *OptEngines* to multiple actual simulation systems running in parallel. Currently, a simple free-drop dispatching strategy is being used—an idle *SimAgent* will pick up a job that is pending in the message queue of the *SimManager* and launch the targeted simulation system. Instead of using some advanced algorithm like OBCA (Yoo et al. 2006) to calculate the number of replications required in each simulation within the OPTIMISE framework, replication analysis and advanced output data analysis is respectively left to the statistical analysis capability of the simulation systems and client applications to facilitate the genericity of the server components and *SimAgents*.

20.3.3 OPTIMISE Client Applications

In principle, any applications that consume the Web services provided by OPTIMISE can be called an OPTIMISE client application. To address the needs in different real-world industrial case studies, a number of applications have been developed. The windows with blue title bar in Fig. 20.2 are the graphical user interface (GUI) developed for allowing users to modify the optimization and simulation settings and viewing optimization results (an operations schedule, in this case study) for a postal distribution terminal in Sweden. For this particular case, the GUI allows the user to adjust the weights in a preference-based multi-objective function, in order to test if improved results can be achieved with different tradeoffs in the weighting. More details of this OPTIMISE client application and the case study can be found in (Persson et al. 2006). An OPTIMISE client application can be developed by extending an existing spreadsheet application, if an existing input data interface in e.g. Microsoft Excel is already available for simulation data input. An example application can be found in (Andersson et al. 2007). Other than these specific OPTIMISE client applications, there are some other generic applications developed for the general monitoring/control of the OPTIMISE Server Components and manage data for all optimization projects. One such application that can be very useful for a wide range of SBO applications is the OPTIMISE Browser, with which it is possible to browse new and historical optimization data from *OptDB*. The windows with red title bars in Fig. 20.2 are the screen shots showing how the OPTIMISE Browser can be used to view the results of a multi-objective optimisation problem (MOOP) by plotting the *Pareto front*, as well as monitoring the accuracy of the underlying metamodel

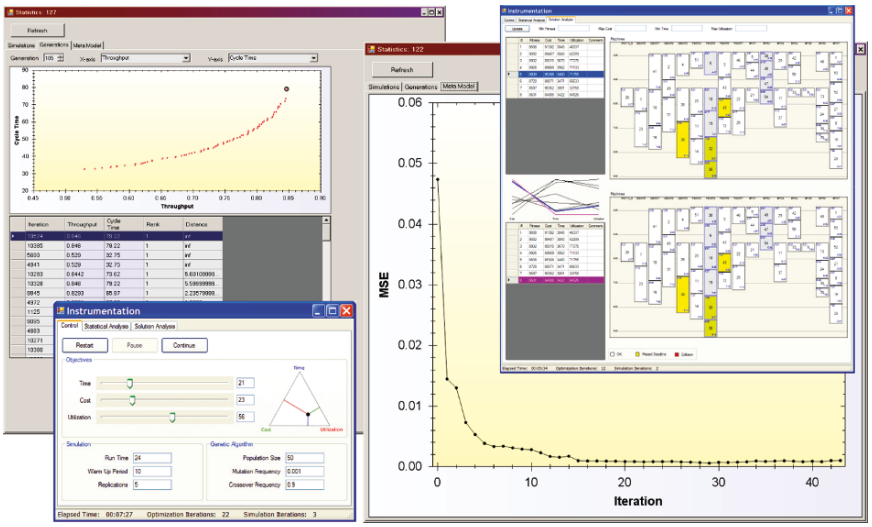


Fig. 20.2 Screen shots of some OPTIMISE client applications

trained in parallel with the optimisation process. Metamodeling, MOOP and Pareto front, etc., shall be the main topics discussed in the coming sections.

20.4 Artificial Neural Network Based Metamodeling

As mentioned, one of the biggest barriers in promoting the use of SBO for "real-time" decision making or daily operations scheduling is the unacceptably long computing time to run a significantly complex or detailed simulation model. While this problem can be alleviated by using parallel and distributed simulation as supported inherently by OPTIMISE and other Web-based simulation platforms, there are some other techniques that can further enhance the efficiency of the simulation runs by replacing the computationally expensive models with a surrogate model. It is intuitive to see the advantages of replacing a complex detailed simulation model with a "simplified" model to enhance an SBO process. This is usually referred to as models with lower fidelity or higher abstraction in the literature (Madan et al. 2005). Within the scope of this paper, we concentrate on a more versatile and generic metamodeling method, namely the use of artificial neural network (ANN) based metamodels, inherently supported by the OPTIMISE framework to enhance the performance of the SBO processes, for both single-objective and multi-objective problems.

ANNs are mathematical models that attempt to imitate the behavior of biological brains. They have universal function approximation characteristics and also the ability to adapt to changes through training. Instead of using a pre-selected functional form, ANNs are parametric models that are able to learn underlying relationships between inputs and outputs from a collection of training examples. ANNs have very good generalization capability when processing unseen data and are robust to noise and missing data. Hurrion and Birgil (1999) summarized several other advantages of using ANN for metamodeling when compared to classical regression-based techniques. All these advantages make ANNs very suitable to be used as the surrogates for computationally expensive simulation models. From a system perspective, ANNs provide versatile metamodeling methods for OPTIMISE because they can be developed, trained and evaluated off-line by using the data saved in *OptDB* or on-line by using the new input-output data pairs generated and returned from the simulation runs. Hence, in every SBO process that uses ANN metamodeling, there is in parallel an ANN training problem, handled simultaneously by the core algorithm in OptEngines. The ANN training process is in principle an optimization problem by itself because the goal is to find the optimal topology and parameters (e.g., weights and bias) to minimize the *mean squared error* (MSE), which is common to many ANN training algorithms.

There exist commercial optimization packages that use ANN to filter out possibly inferior solution candidates. Nevertheless, most of these packages treat the optimization engine as a black box, so there is no way to monitor or control the metamodeling process. One exception is the *OptQuest Callable Library* (OCL) (Voß and Woodruff 2000). With OCL it is possible to develop applications that retrieve data

from the ANN in use by the scatter search and Tabu search algorithm. But until now OCL does not explicitly support MOOP that aims at finding Pareto fronts. In contrast, the proposed OPTIMISE system provides a set of features that support decision makers to collect information effectively from multi-objective optimizations, which is illustrated in the coming section.

20.5 Metamodel-Assisted Multi-Objective Optimization

20.5.1 Multi-Objective Optimization Problems (MOOP)

In a general MOOP, there exists no single best solution with respect to all objectives as improving performance on one objective would deteriorate performance of one or more other objectives (Deb et al. 2001). A simple method to handle a MOOP is to form a composite objective function as the weighted sum of the conflicting objectives. Because a weight for an objective is proportional to the preference factor assigned to that specific objective, this method is also called preference-based strategy. Apparently, preference-based MOO is simple to apply because by scalarizing an objective vector into a single composite objective function, a MOOP can be converted into a single-objective optimization problem and thus a single trade-off optimal solution can be sought effectively. However, the major drawback is that the trade-off solution obtained by using this procedure is very sensitive to the relative preference vector. Therefore, the choice of the preference weights and thus the obtained trade-off solution is highly subjective to the particular decision maker. On the other hand, without the detailed knowledge about the system under study, it is also a very difficult task to select the appropriate preference vector. Also from a mathematical point of view, this weighted function approach may not be able to find an efficient point if the Pareto front is not convex. Therefore, for a decision maker, it would be very useful if the posterior Pareto front can be generated quickly by using a multi-objective SBO algorithm so that he/she can choose a configuration that is the "best" trade off among these conflicting performance measures. We shall illustrate this by using an example of a buffer space allocation problem, commonly found in manufacturing industry.

20.5.2 Multi-Objective Buffer Space Allocation

Since the publication of Buzacott's seminal work on the impact of limited buffer spaces on the performance of automatic transfer lines, the problems of optimal buffer allocation (OBA), commonly found in manufacturing or other types of systems, are studied extensively (Buzacott and Shanthikumar 1993). Generally stated, OBA problems can be classified into either *primal* or *dual* (Gershwin 1987). In a

primal problem, the objective is to minimize the total buffer space subject to a production rate (throughput) constraint. In a dual problem, subjecting to a total buffer space constraint, maximization of the *throughput* (TH) is desired. While TH is the most common measure to determine the productivity of a line, there are two other important performance measures used in production systems analysis, namely, *cycle time* (CT), also known as sojourn time or manufacturing lead time (MLT), and *work-in-process* (WIP). A short average CT means the total time a part or a product spent in the system is short, which means customer orders can be fulfilled within a shorter time, and thus leverages the company's competitiveness. A low WIP means the cost spent on material handling and storage is lowered and thus is also highly desired. Therefore, to a production manager, an ideal configuration is the one that maximizes TH while simultaneously minimizing CT and WIP. Unfortunately, this is never an easy task because in most of the real-world complex systems, these objectives are in conflict with each other. Consider a balanced but unsynchronized and unreliable flow line with 12 workstations (WSs). To absorb the disturbances of the WSs due to break down, buffers between two sequential WSs are needed. There are hence 11 inter-station buffers. Such a system design can be found commonly in machining lines in industry making discrete products, e.g., automotive engines. By neglecting other parts of the system before and after the line, we assume the first WS runs with a pull policy and is never starved, and the rest of the line runs with push policy. This also means that the last WS will never be blocked. The processing times (balanced) at each station are assumed to be independent random variables that follow exponential distribution, with mean service rate 1 part/min. The exponentiality of the processing time is rather unrealistic in real-world production systems, but it can provide good approximation to the variability due to failures and repairs (Spinellis and Papadopoulos 2000).

A simple DES model is built for this line, and CT-TH plots with varying buffer capacity (b_i) are generated by using the data in *OptDB*. For this simple OBA problem, two MOO algorithms have been tested and compared: a fast elitist non-dominated sorting genetic algorithm (NSGA-II) (Deb et al. 2001) and its metamodel-assisted version, MA-NSGA-II. Figure 20.3 illustrates the pseudo code of the MA-NSGA-II implemented and compiled with some standard OPTIMISE libraries into an *OptEngine*. The algorithm is primarily an extension of the original NSGA-II to incorporate (1) the non-dominated sorting and simulation evaluations (i.e., *Sim_Evaluate()*) running in parallel; (2) estimation of output variables using ANN metamodel (*MM_Estimate()*) and (3) update *OptDB* when evaluations and sorting by crowding distance on the whole generation is completed.

Both NSGA-II and MA-NSGA-II are run using an enlarged sampling space $(\mu + \lambda)$ selection strategy with population sizes of parents and offspring each equal to 100 (i.e. $\mu = \lambda = 100$). In every generation, the μ parents and the λ offspring compete for the survival based on a fast non-dominating sorting (FNS) and the crowded distance tournament selection. Nevertheless, for the MA version, in every generation, 1000 candidate solutions are generated. Rather than running expensive simulation for all these candidates, evaluations are made using the ANN metamodel. An FNS is used to sort the candidates and then the best λ best candidates are selected

function MA_NSGA_II( )

returns: Solutions in the current population sorted according to ranks

$T \leftarrow$ Get_max_gen( ), $N \leftarrow$ Get_pop_size( )

$P_0 \leftarrow$ Generate_Initial_Population( )

$t \leftarrow 0$

while $\left(t \le T \ \wedge\ \textbf{not}\ \text{Stop\_optimisation}(\ )\right)$ **do**

Sim_Evaluate(P_t) || $F_t \leftarrow$ Fast_Non_Dominated_Sort(P_t)

Train_Meta_Model$\left(\bigcup_{i=0}^{t} P_i\right)$

$Q_t \leftarrow$ Generate_Offspring(P_t)

MM_Estimate(Q_t)

$F_{t+1} \leftarrow$ Fast_Non_Dominated_Sort$(P_t \cup Q_t)$

$P_{t+1} \leftarrow \{\varnothing\}$

while $\left(|P_{t+1}| < N\right)$ **do**

Assign_crowding_distance(F_k)

$P_{t+1} \leftarrow P_{t+1} \cup F_k$

endwhile

Crowding_Sort$(P_{t+1}, >_c)$

$P_{t+1} \leftarrow P_{t+1}[1:N]$

Update_OptDB(P_{t+1})

$t \leftarrow t+1$

endwhile

return (F_t)

Fig. 20.3 Pseudo code of MA-NSGA-II

to be the offspring solutions based on the estimated values in the multi-objective functions. Simulation runs are then performed on these λ candidates for accurate evaluations.

Figures 20.4 and 20.5 are the CT-TH plots, showing the solution points found at the end of the 4th generation, generated using NSGA-II and MA-NSGA-II respectively. These two figures are extracted directly from the data plots displayed in the OPTIMISE Browser. By comparing Figs. 20.4 and 20.5, it can be seen that the MA-NSGA-II outperforms NSGA-II in terms of both convergence towards the Pareto front (rank 1) and diversity (compare the low TH regions) in early generations. Since the major computational burden in almost all SBO applications is owing to the simulation runs. The overheads due to the metamodeling and additional sorting

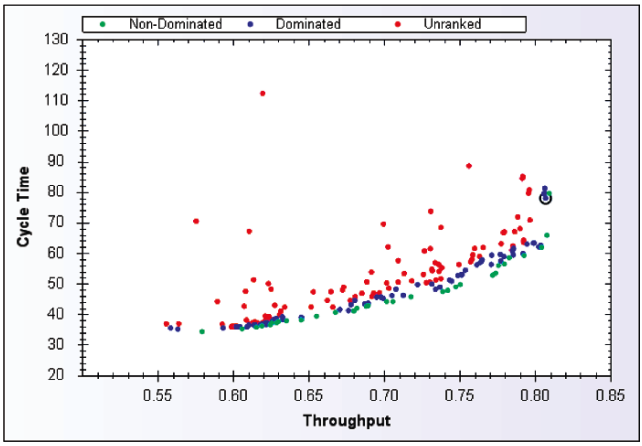


Fig. 20.4 CT-TH plot (generation 4) for the OBA problem using NSGA-II

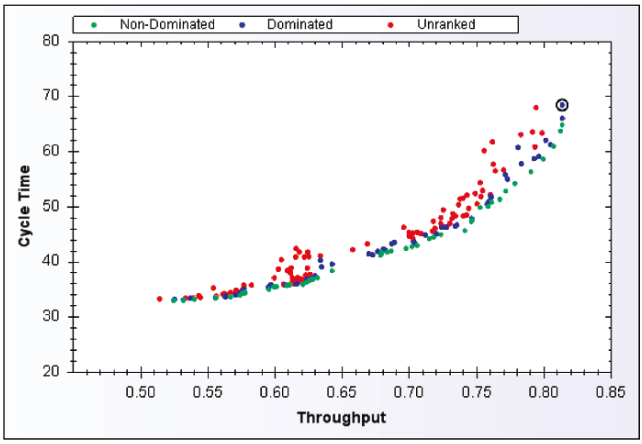


Fig. 20.5 CT-TH plot (generation 4) using MA-NSGA-II

are hence negligible. As a short summary, this simple example has demonstrated the benefit of using the metamodeling feature in OPTIMISE to quickly generate information for industrial decision-making support for problems that can be put into the context of a MOOP.

20.6 Metamodel-Assisted Single-Objective Optimization

The benefit offered by OPTIMISE in allowing users to dynamically change metaheuristic algorithms in an optimization process can be exemplified by a test case applied to a multi-task machining line in Volvo Aero, Sweden. A DES model is

developed using the SIMUL8 software package[3]. The simulation takes planned lead time of different product types as inputs and the mean tardiness of different product types as outputs. For the scenario considered, 11 product types are included, and a duration corresponding to one year of production is simulated. The aim of the optimization is to minimize the mean lead time and minimize the mean tardiness of different products in the machining line. This has been formulated into a single-objective optimization problem with the objective function described by

$$\sum_{i \in P} \left(w_s i_{shortage} + w_t i_{tardiness} \right) - w_u utilisation$$

where P is the set of all products and w is the weighted importance of an objective.

As shown in Fig. 20.6, when compared with using either a GA or a hill climbing algorithm alone, lower objective function value can be found by running a combination of GA and hill climbing sequentially (switched at iteration 200). GA helps in exploring the search space and avoiding falling into a local optimum. When the GA is almost saturated, switching to hill climbing helps in further exploitation. This example shows the feature of allowing dynamically changing search methods in an optimization process can be useful for OPTIMISE users to find better results.

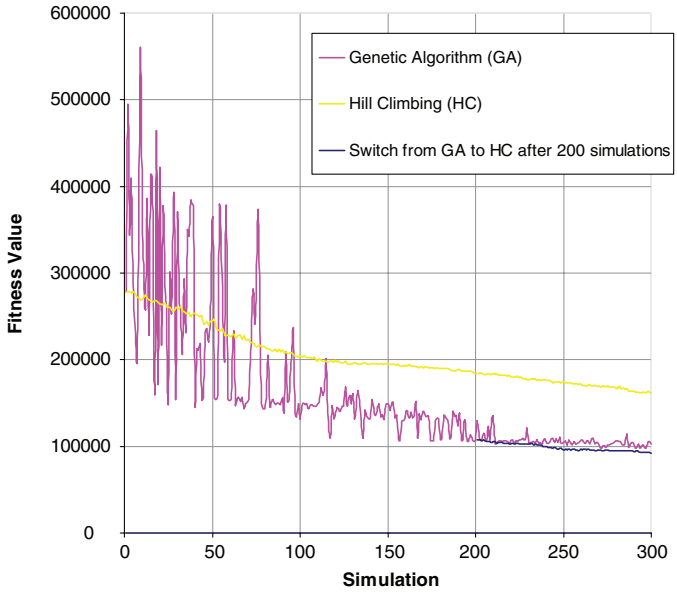


Fig. 20.6 Fitness values found by different search methods in the Multi-Task industrial case

[3] www.simul8.com.

20.7 Conclusions

We have implemented OPTIMISE as a Web-based parallel and distributed simulation platform to enable managers/engineers who are neither simulation nor optimization experts to access complex optimization and simulation components that are located and maintained remotely by other specialists or so-called *optimization service providers* (OSP). This architecture is believed to be particularly of interest to small and medium enterprises (SME) that do not have access to complex simulation systems or possess the knowledge and resource to develop, house and maintain their own SBO applications.

OPTIMISE is now functional and being used in several industrial-based test cases, together with a number of Swedish companies (see Acknowledgement). Some of the partners have started to use OPTIMISE at their offices to retrieve experimentation and optimization data from their simulation models and monitor the optimization processes. The "illusion of simplicity" provided by the Web services technology has been recognized by our partners as viable to promote and broaden the use of SBO among industry in the future. Nevertheless, new features have to be added to OPTIMISE if it has to be used to serve a large amount of "clients." Our future work is therefore to develop and test different dispatching algorithms, based on advanced concepts and techniques from research in multi-agent systems and parallel computing.

Acknowledgements The authors gratefully acknowledge the Knowledge Foundation (KK Stiftelsen), Sweden, for the provision of research funding and the industrial partner companies, including Posten AB, Volvo Aero, Volvo Cars Engine, Volvo Powertrain and Volvo Technology, for their support in this study.

References

Andersson M, Persson A, Grimm H, Ng A (2007) Simulation-based scheduling using a genetic algorithm with consideration to robustness: A real-world case study. In: The Proceedings of the 17th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM '07), 18–20, Philadelphia, USA, in press. June

April J, Better M, Glover F, Kelly J (2004) New advances for marrying simulation and optimization. In: Proc. of the 2004 Winter Simulation Conference, Washington, D.C., 5–8, pp 80–86. December

Biles WE and Casebier JB (2004) Web based evaluation of material handling alternatives for automated manufacturing: A parallel replications approach. In: Ingalls RG, Rossetti MD, Smith JS, Petters BA (eds) Proc. in 2004 Winter Simulation Conference, Orlando, Florida, USA

Biles WE, Daniels CM, O' Donnell TJ (1985) Statistical considerations in simulation on a network of microcomputers. In: Blais GC, Soloman SL, Gantz DT (eds) Proc. of the 1985 Winter Simulation Conference, San Francisco, California

Biles WE and Kleijnen JPC (2005) International collaborations in web-based simulation: a focus on experimental design and optimization. In: Kukl ME, Steiger NM, Armstrong FB Joines JA (eds) Proc. in Winter Simulation Conference, Orlando, Florida, USA

Boesel J, Bowden RO Jr, Kelly JP, Westwig F (2001) Future of simulation optimization. In: Peters A, Smith JS, Medeiros DJ, and Rohrer MW (eds) Proc. of the Winter Simulation Conference, pp 1466–1469

Buzacott JA and Shanthikumar JG (1993) Stochastic Models of Manufacturing Systems, Prentice Hall, New Jersey

De Vin LJ, Ng AHC, Oscarsson J (2004) Simulation based decision support for manufacturing system life cycle management. Journal of Advanced Manufacturing Systems 3(2):115–128

Deb K, Agrawal S, Pratap A, Meyarivan T (2001) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Parallel Problem Solving from Nature VI (PPSN-VI)

Fu MC, Andradóttir S, Carson JS, Glover F, Harell CR, Ho YC, Kelly JP Robinson SM (2000) Integrating optimization and simulation: research and practice. In: Joines JA, Barton RR, Kang K, Fishwick PA (eds) Proc. of the Winter Simulation Conference, pp 610–616

Fujimoto RM (2000) Parallel and Distributed Simulation Systems. New York, Wiley, 2000

Gershwin SB (1987) An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. Operation Research 35(2):291–305

Heidelberger P (1988) Discrete-event simulation and parallel replications: statistical properties. Scientific and Statistical Computing 9:1114–1132

Hurrion RD and Birgil S (1999) A comparison of factorial and random experimental design methods for the development of regression and neural network simulation metamodels. Journal of Operational Research Society 50:1018–1033

Kumara SRT, Lee YH, Tang K, Dodd C, Tew J, Yee ST (2002) Simulation anywhere any time: Web-based simulation implementation for evaluating order-to-delivery systems and processes. In: Yücesan E, Chen CH, Snowdon JL, Charnes JM (eds) Proc. in Winter Simulation Conference, Piscataway, New Jersey, pp 1251–1259

Law AM, McComas MG (2002) Simulation-based optimization. In: Yücesan E, Chen CH, Snowdon JL, Charnes JM (eds) Proc. of the 2002 Winter Simulation Conference, 8–11, San Diego, California, pp 41–44 Dec

Madan M, Son YJ, Cho H, Kulvatunyou B (2005) Determination of efficient simulation model fidelity for flexible manufacturing systems. International Journal of Computer Integrated Manufacturing 18(2–3):236–250

Marr C, Storey C, Biles WE, Kleijnen JPC (2000) A Java-based simulation manager for web-based simulation. In: Joines JA, Barton RR, Kang K, Fishwick PA (eds) Proc. in Winter Simulation Conference, Orlando, Florida, USA

Persson A, Grimm H, Ng A (2006) On-line instrumentation for simulation-based optimization. In: Perrone LF, Wieland FP, Liu L, Lawson BG, Nicol DM, Fujimoto RM, (eds) Proc. of the Winter Simulation Conference, Monterey, CA, USA, 3-6 December, pp 304–311

Spinellis DD and Papadopoulos CT (2000) Stochastic algorithms for buffer allocation in reliable production lines. Mathematical Problems in Engineering 5:441–458

Stainforth D, Kettleborough J, Martin A, Simpson A, Gillis R, Akkas A, Gault AR, Collins M, Gavaghan D, Allen M (2002) Climateprediction.net: Design principles for public-resource modeling research. 14th IASTED International Conference Parallel and Distributed Computing and Systems, Cambridge, USA

Voß S and Woodruff DL (2000) Optimization Software Class Libraries. Kluwer Academic Publishers, Secaucus, NJ, USA

Yoo T, Cho H, Yücesan E (2006) Web service based parallel and distributed simulation experience. WSEAS Transactions on Systems 5(5):973–980

zu Eissen SM and Stein B (2006) Realization of Web-based simulation services. Computers in Industry 57:261–271

Chapter 21
Natural Language Processing Methods for Extracting Information from Mathematical Texts

Nicole Natho, Sabina Jeschke, Olivier Pfeiffer, and Marc Wilke

21.1 Introduction

Managing knowledge takes up a significant role in modern organization and society. Every day, numerous books and publications are released to disseminate new knowledge. In particular, the World Wide Web is an example of a huge unsystematic knowledge base that provides access to a lot of information. Too much information leads to the proliferation of knowledge called "information glut," which is not easy to handle. Applications to specialize and generalize information gain in importance in the search for solutions to face the new challenges.

In this work a reasonable knowledge management system approach to access mathematical information from different sources such as textbooks and publications is presented. Information is extracted from these sources and put into a suitable format for integration into a knowledge base. Many of the current approaches of information extraction are based on statistical analysis of the correlation between terms and knowledge elements. These approaches, while very successful neglect the advantages offered by the precision of the mathematical language. This precision enhances the use of alternative natural language processing methods like "head-driven phrase structure grammar" for automated semantic annotation of mathematical texts. Thereby we obtain information within a given context (knowledge) that imparts mathematical correlation between terms and concepts. A knowledge management system based on this technology provides new opportunities and challenges: automated knowledge acquisition, sophisticated solutions for user interfaces, requirement of intelligent search and retrieval mechanisms, visualization of knowledge, an extend ontology spanning several sources for an overview about

Nicole Natho and Olivier Pfeiffer
Berlin University of Technology, MuLF—Center for Multimedia in Education and Research, 10623 Berlin, Germany

Sabina Jeschke and Marc Wilke
University of Stuttgart, RUS—Center of Information Technologies, 70550 Stuttgart, Germany

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

mathematics itself, etc. A comprehensive knowledge management system needs to address the accumulation, storage, merger, evaluation and representation of mathematical information for different applications like encyclopedias, context sensitive library search systems, intelligent book indexes and e-learning software using standardized methods.

The MARACHNA System implements the knowledge acquisition of such a knowledge management system. MARACHNA generates mathematical ontologies of extracted information represented as elements of mathematical concepts and objects as well as relations and interconnections between them. These ontologies are mapped onto the structure of a knowledge base. This is possible because mathematical texts possess a distinctive structure caused by the theoretical design of mathematics itself and author's preferences. This structure is characterized by representative text module "entities" such as definitions, theorems and proofs. Entities are commonly used to describe mathematical objects and concepts and represent the key-content of mathematical texts. Furthermore, the information contained within these entities forms a complex relationship model in the shape of a network defining an ontology of the specific mathematical text. The basic ontology that forms the skeletal mathematical structure for all ontologies follows the ideas of Hilbert [1] and Bourbaki [2]: Mathematics as a whole can be derived from a small set of axioms using propositional logic[1]. Propositional logic is reflected in the structure and diction of mathematical texts. Therefore, an ontology of mathematical content can be directly extracted from mathematical texts. As a result, MARACHNA is able to integrate mathematical entities from very different sources, such as different mathematical textbooks, articles or lectures, independent of the upper ontology (describing general concepts) preferred and used by the authors of those sources. This approach—based on natural language processing techniques—offers several advantages over the more common approach based on purely statistical methods as used in, e.g., Google, while avoiding the increased complexity of automated reasoning systems. It utilizes both the ontology inherent in mathematics itself and the ontology defined by the structure of the text chosen by the author. Thus, MARACHNA provides mechanisms for the creation of knowledge networks from mathematical texts, such as textbooks, digital lectures or articles. The extracted mathematical terms and concepts are further annotated regarding their role in the ontology of the original text, thus preserving the upper ontology used by the authors, and can map relationships in a very fine-grained way. In addition, these networks can be used to create an overview of mathematical content. Therefore, they offer different levels of information detail. To store and manage these information snippets of the texts in a knowledge base the Jena framework is used. Additionally the search engine Apache Lucene [4] is utilized for information retrieval and testing purposes.

[1] This approach is valid within the context of MARACHNA, despite Gödel's Incompleteness Theorem [3], since we map existing and proven mathematical knowledge into a knowledge base and do not want to prove new theorems or check the consistency of the theorems we store.

21.2 Linguistic Approach

The main carriers of information in mathematical texts are entities. Based on a linguistic classification scheme [5,6] they are analyzed using natural language processing methods. The levels of this scheme are depicted in Fig. 21.1:

- on the *entity level* the used entities and their arrangement within the text are described;
- on the *structure level* the internal structure of an entity is described (e.g., the proposition and assumptions of a theorem);
- on the *sentence level* characteristic sentence structures, frequently found in mathematical texts, are described;
- on the bottommost and most detailed level of this scheme, the *word and symbol level*, single symbols and words and the relations between them are schematized [7].

Using these structures and linguistic relations, mathematical information is educed from the text and integrated into a knowledge base, consisting of one or more directed graphs representing terms, concepts, and their interrelations. The knowledge base is encoded in the web ontology language OWL [8] and is based upon an ontology of the mathematical language. Linguistic analysis of entities produces triples of two nodes (representing mathematical terms and relations) and one relation (describing the type of connection between the nodes). The triples can be nested, i.e., they can be used as nodes of other triples, thus facilitating representation of complex interrelations. Here different types of relations describe different types of keywords or linguistic phrases in mathematical texts (e.g. two nodes representing

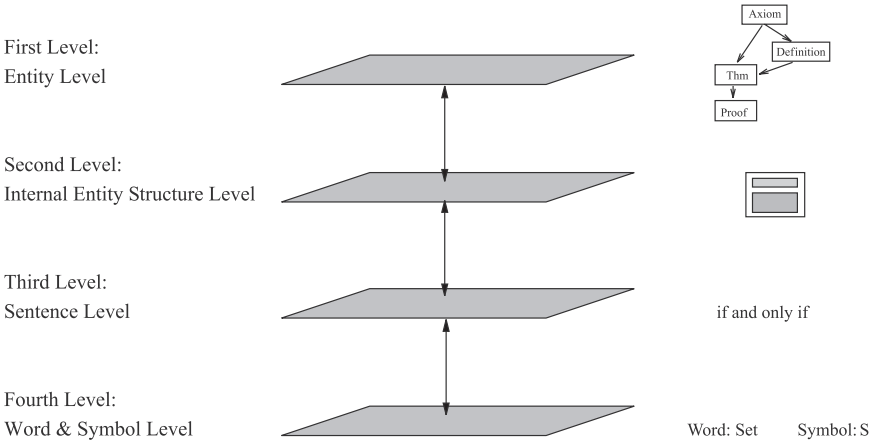


Fig. 21.1 Linguistic classification scheme

two theorems, connected with the relation “is equivalent to”). Finally the triples are integrated into the knowledge base. This transfers the knowledge from mathematical texts to the knowledge base, generating a very fine-grained structure.

21.3 Technical Aspects of MARACHNA

MARACHNA consists of four separate modules (see Fig. 21.2):

- the *preliminary analysis* of the input,
- the *syntactic analysis*,
- the *semantic analysis*, and
- the *integration of the results* into a knowledge base.

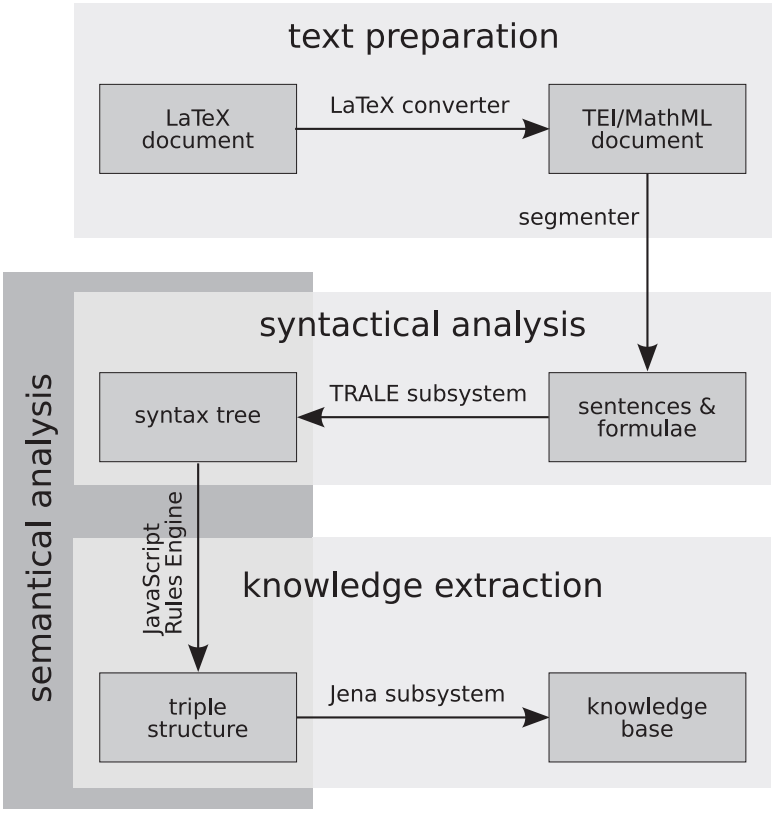


Fig. 21.2 Processing of natural language texts

21.4 Design Decisions

The basic design of MARACHNA adheres to the following general guidelines:

- **Modularization**: Components (input segmentation, syntactic analysis, semantic analysis, knowledge base, retrieval) have been designed as independent modules that can be replaced easily;
- **LaTeX as input format**: LaTeX was chosen as input format since it is the standard format for publication in mathematics and the natural sciences;
- **Standard internal data formats**: All data is represented in standard data formats (TEI, MathML, XML, RDF/OWL);
- **Control of the syntactic and semantic analysis**: All rules governing the syntactic and semantic analysis should be modifiable by non-programmers to facilitate future extensions.

21.5 Preliminary Analysis

Currently for the internal representation of the processed texts MARACHNA uses TEI (Text Encoding Initiative [9]) with all mathematical symbols and formulae being encoded in Presentation MathML [10]. For this purpose MARACHNA provides a LaTeX-to-TEI converter. The entities and their interrelations, as well as the information provided on their internal structure level are analyzed and educed from the TEI representation. In the next step the entities are segmented into single sentences and annotated with some meta-information (their parent entity and their role within the internal structure of that entity). For further processing formulae are separated from the enveloping text (replacing incomplex formulae by the associated natural language texts). The described preliminary analysis is implemented in Java using rule-based string comparison.

21.6 Syntactical Analysis

At this point natural language processing techniques are employed to analyze every extracted natural language sentence. This natural language analysis is implemented using the TRALE-System (a PROLOG adaptation [11] of ALE for German [12]) based on a head-driven phrase structure grammar. For use in MARACHNA the underlying dictionary and grammar of TRALE have been expanded to include the particularities of the mathematical language, in order to provide a comprehensive syntactic and some partial semantic information about each sentence. The output of TRALE is then transformed to an abstract syntax tree, symbolizing the structure of the analyzed sentence. At this stage the formulae, which have been processed

separately, and the natural language part of the text have to be restored to unity for further semantic analysis.

The processing of complex formulae has to be done in a separate step and is, however, not yet implemented.

21.7 Semantic Analysis

The semantic analysis is implemented as an embedded JavaScript interpreter [13]. The syntax trees are categorized according to typical structures characteristic for specific mathematical entities and semantic constructs. Each category is transformed into a specific triple structure defined by external JavaScript rules. These rules map typical mathematical language constructs onto the corresponding basic mathematical concepts (e.g., proposition, assumption, definition of a term, etc.). The resulting triples are annotated with additional information from the original text. At present, the resulting OWL documents are only satisfying the OWL full standard. In addition, for each triple element it has to be determined if it represents OWL classes or individuals of OWL classes—complicating the semantic analysis.

21.8 Knowledge Base

The Java-based framework Jena [14] is responsible for inserting created triples into knowledge bases. Jena works with OWL-documents and provides APIs for persistent storage. Additionally, it contains a SPARQL [15] engine for querying RDF documents. It is of particular importance that each analyzed text creates its own new knowledge base, by reason of each author having his own preferences and idiosyncrasies of the representation of a mathematical field. The plain knowledge base consists of fundamental mathematical knowledge including axiomatic field theory and first order logic (cf. Fig. 21.3).

While adding new knowledge to the knowledge base inconsistencies have to be avoided. Therefore a semi-automated approach for integrating new information into the knowledge base is used. Information is only integrated if it is not redundant and does not conflict or contradict older entries. Consequently, a new node has to be linked to an already existing one.

Semi-automatic means here that the user of the MARACHNA System eliminates such conflicts by adding additional information or manually deleting the conflicting entries. This user intervention is performed through a web-based interface by using the framework Jena. The basic idea of this approach is the well-known model of human knowledge processing in psychology: Integration of new knowledge by humans is only possible if it can be linked to existing knowledge conformed by the person's world view. Consequently, correct knowledge has to be newly interpreted. Fortunately textbooks follow strict rules so that automated integration should be the normal case and semi-automated integration the exception.

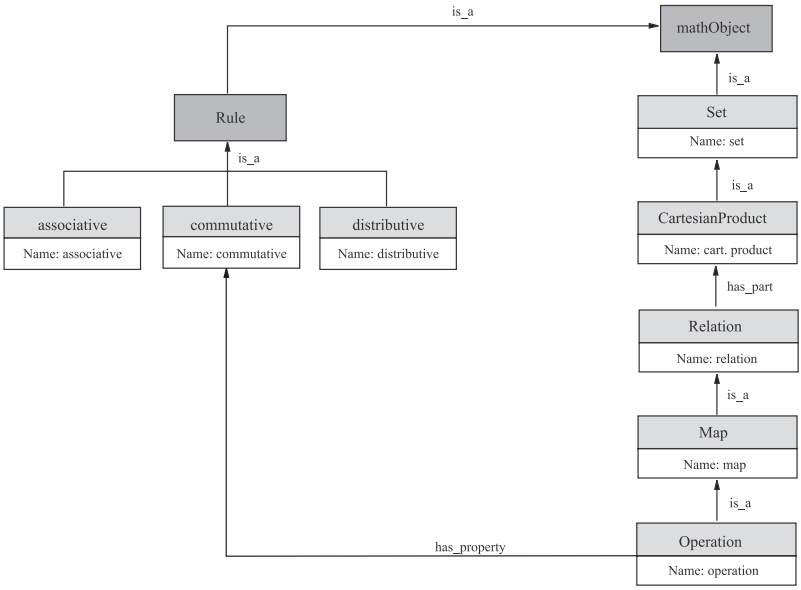


Fig. 21.3 Basic knowledge base

21.9 Linguistic Analysis: An Example

The following definition of a mathematical group [16] is illustrated (cf. Fig. 21.4) as an example.

Let G be a set, $G \neq \emptyset$,
$a, b \in G$:
$^* : G \times G \rightarrow G$,
$(a, b) \rightarrow a^* b$.

The tupel $(G, ^*)$ *is called a Group if the following conditions hold*:

1. Associativity: $\forall a, b, c \in G,\ a^*(b^*c) = (a^*b)^*c$,
2. Neutral Element: $\exists e \in G$ such that $\forall a \in G,\ a^*e = e^*a = a$,
3. Inverse Element: $\forall a \in G \exists e \in G$ such that $a^{-1}{}^*a = a^*a^{-1} = e$ where e is the neutral element.

The assumption "Let $\boldsymbol{G}$ be a set…" is recognized based on the knowledge of the use of definite keywords of mathematical texts. In the same way "*The tupel* $(\boldsymbol{G}, ^*)$ is called…conditions hold:" is identified as the proposition and the three following points (Associativity, Neutral Element, and Inverse Element) as definition or properties. Morphologic, syntactic and semantic analysis results in the representation as triples of predicate, subject, and object. The phrase "Let **G** be a set" is mapped to (is_a, set, **G**), for example. In the end the network of relations originates from the identification of identical subjects and objects of different triples.

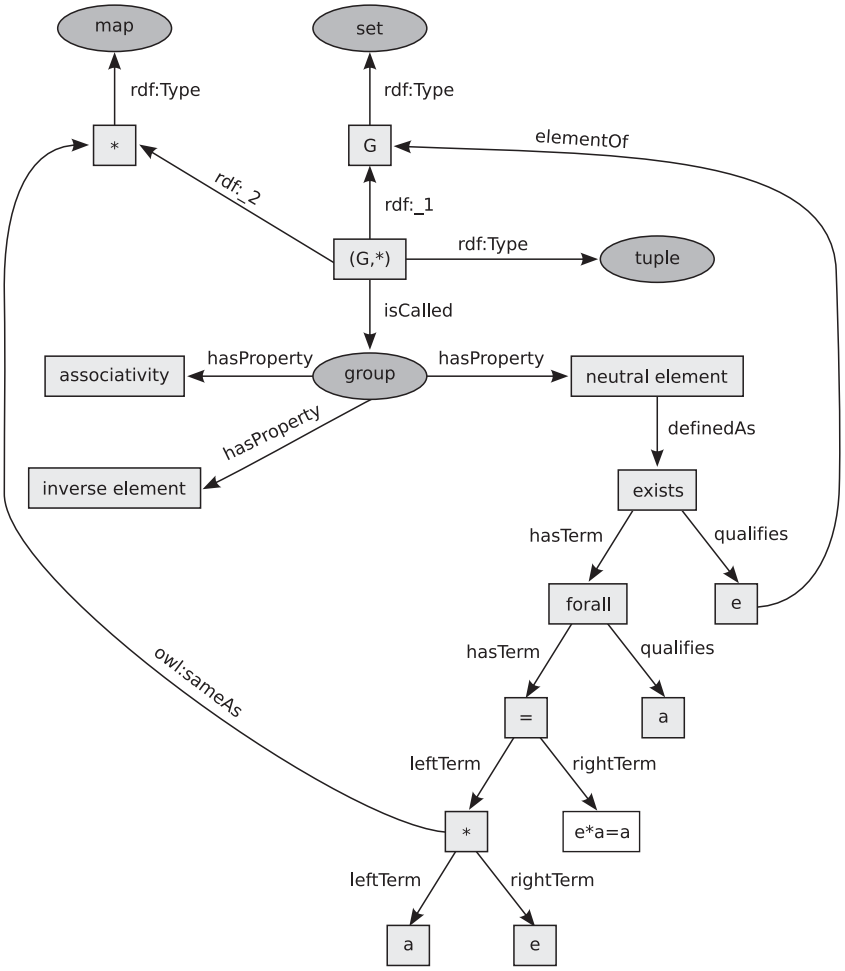


Fig. 21.4 Knowledge base: definition of a group

The generated pattern of triples describing the entities and the relations between them are integrated into the database using OWL. The OWL representation is generated by extraction of the significant information from the text and supplies an outline of the mathematical knowledge presented by the author.

21.10 Evaluation of the Basic Concept

While the prototypical implementation of MARACHNA was able to analyze German texts only, its capabilities have been currently extended to analyze English texts as well. The current implementation proves the viability of this semi-automated

method of semantic extraction for selected text elements, as illustrated above. Semantic extraction results in information snippets of the analyzed mathematical text that can be assimilated in the knowledge base (cf. Fig. 21.4). The ontology mapping approach shall be evaluated in a next step by analyzing whole textbooks on linear algebra with a subsequent merging of the resulting knowledge bases and a concluding comparison of the generated outcome with some mathematical "standard" encyclopedia—respectively the corresponding part of the used encyclopedia.

21.11 Related Works

MARACHNA consists of several separate subprojects. While we are unaware of any current research mirroring MARACHNA's global concept, there are a number of projects with similar aims (if different approaches) to the subprojects within MARACHNA. Fundamental research into ontologies for mathematics was performed by T. Gruber and G. Olsen [17]. Baur analyzed the language used in mathematical texts in English [18]. The linguistic analysis of mathematical language is based on TRALE [11]. MBASE is an example of a mathematical ontology created by humans. The structuring of mathematical language and the modeling of the resulting ontology is similar to the work performed by humans in the creation of several mathematical encyclopedias like MathWorld [19]. The ontologies represented in these projects can be used as an interesting comparison to the ontologies and knowledge bases automatically generated by MARACHNA. WordNet/GermaNet [20–22] are intelligent thesauri providing sets of suitable synonyms and definitions to a given queried expression. WordNet uses a similar approach to the automated semantic analysis of natural language compared to MARACHNA. However, WordNet does not provide detailed correlations between dissimilar or not-directly related terms. Helbig [23] has designed a concept for automated semantic analysis of German natural language texts. The Mizar [24–26] system uses a both human and machine-readable formal language for the description of mathematical content as input for automated reasoning systems. An example for a different approach towards information extraction from mathematical natural language texts, based on automated reasoning, is the work of the DIALOG [27] project. DIALOG is aimed at processing natural language user input for mathematical validation in an automated reasoning system. Both Mizar and DIALOG, unlike MARACHNA, require complex reasoning systems for their extraction of information. The HELM [28] project defines metadata for semantic annotation of mathematical texts for use in digital libraries. However, these metadata have to be provided by the original authors. MoWGLi [29] extracts the metadata provided in HELM and provides a retrieval interface based on pattern-matching within mathematical formulae and logic expressions. The PIA [30–32] project extracts information from natural language texts (not limited to mathematics) using machine learning techniques. PIA requires an initial set of human expert annotated texts as a basis for the information extraction performed by intelligent agents.

21.12 Ontology Engineering in MARACHNA—An Outlook

21.12.1 Summary

Exact transfer of knowledge is the central point of mathematical language. Hence the knowledge base contains pure mathematical knowledge with no interpretation needed. The semantic extraction (acquisition process) obeys well defined rules based on the structure of the entities. The same holds true for storage and organization of the mathematical content in the knowledge base. Still some subproblems remain unsolved regarding the merging and the extraction from knowledge bases and are subject of further developments.

21.12.2 Knowledge Management and Ontology Engineering

MARACHNA generates several levels of ontologies from mathematical texts. The low-level ontology describes the correlation between mathematical terms both within mathematical entities and between these entities. The upper ontology reproduces the structure of the processed text and thus the field ontology used by the author. Both of these ontologies facilitate data-retrieval, the upper ontology in direct response to queries, the low-level ontology by showing interconnections and relations not directly included by the author or by offering additional links between the upper ontologies of several authors.

Different authors use their own notations and phrasing, based on their personal preferences, their audience, and their didactical goals. Therefore each text is stored in its own dedicated knowledge base. On the one hand creating a sensible ontology mapping algorithm to unify knowledge from numerous small knowledge bases, each of them related to one text, in one superior knowledge base is one of developments intended for MARACHNA's near future. On the other hand, tools and strategies for unitizing the unified knowledge base into smaller knowledge bases, particularly tailored to the preferences and requirements of specialized user groups have to be devised and implemented and realized. The unitized knowledge bases could be used as the basis of an intelligent retrieval system, based on the concept of personal information agents (PIAs): Using different versions of an entity for elementary and high school might be wise, for example. Moreover unitized knowledge bases using different notations could contribute to the harmonization of the teaching material within a sequence of courses.

21.12.3 Retrieval

The design of the retrieval interface aims at supporting users in understanding and learning mathematics. The interface shall offer tools to select information based

on the user's personal preferences (e.g., an example oriented vs. an axiomatic approach). Additionally the integration of different roles (learner, instructor, administrator) into a generic interface is desirable. The retrieval interface shall be implemented as a web service.

21.12.4 Processing of Mathematical Formulae

As formulae form a major portion of mathematical texts and constitute a primary source of information in these texts, it is desirable to be able to include their content in the analysis and representation. Currently, this important feature is not implemented. However, we are investigating an approach to rectify this deficiency. Therefore the use of a syntactical analysis, similar to those used in computer algebra systems, in combination with contextual grammars (e.g., Montague grammars) to correlate the information given in a formula with information already provided in the surrounding natural language text, is proposed. Using this approach should allow for the integration of formulae and their informational content in the network created by the analysis of the natural language text. It should be pointed out that we do not aim for machine-based understanding of the formulae, as automatic reasoning systems would require. Instead, formulae are to be treated as a different representation of mathematical knowledge, to be integrated into the knowledge base in a similar manner to that used for the natural language text. However, the analysis proposed here can be used as a first step in a further process leading to viable input for such reasoning systems, providing additional assistance in building the knowledge base.

References

1. Hilbert D (1928) Die grundlagen der mathematik. Abhandlungen aus dem Mathematischen Seminar der Hamburgischen Universität 6:65–85
2. Bourbaki N (1974) Die Architektur der Mathematik. Mathematiker über die Mathematik. Springer, Berlin
3. Gödel K (1931–1932) Über formal unentscheidbare Sätze der Principia Mathematica und verwandte Systeme I. Monatsheft f. Mathematik und Physik 147f
4. The Apache Software Foundation: Apache Lucene. http://lucene.apache.org/
5. Jeschke S (2004) Mathematik in Virtuellen Wissensräumen—IuK-Strukturen und IT-Technologien in Lehre und Forschung. PhD thesis, Technische Universität Berlin
6. Natho N (2005) MARACHNA: Eine Semantische Analyse der Mathematischen Sprache für ein Computergestütztes Information Retrieval. PhD thesis, Technische Universität Berlin
7. Grottke S, Jeschke S, Natho N, Seiler R (2005) mArachna: A classification scheme for semantic retrieval in elearning environments in mathematics. In: Proc. of the 3rd International Conference on Multimedia and ICTs in Education, June 7–10, 2005, Caceres/Spain
8. W3C: Web Ontology Language. http://www.w3c.org/2004/OWL
9. The TEI Consortium: Text Encoding Initiative. http://www.tei-c.org
10. W3C: MathML. http://www.w3.org/Math
11. Müller S TRALE. http://www.cl.uni-bremen.de/Software/Trale/index.html

12. Müller S (2005) Deutsche syntax deklarativ: Head-driven phrase structure grammar für das Deutsche. In: Linguistische Arbeiten 394. Max Niemeyer Verlag, Tübingen
13. Mozilla Foundation: Rhino. http://www.mozilla.org/rhino/
14. Jena: A Semantic Web Framework for Java. http://jena.sourceforge.net
15. W3C: SPARQL. http://www.w3.org/TR/rdf-sparql-query/
16. Wüst R (2005) Mathematik für Physiker und Mathematiker, Bd.1. Wiley-VCH
17. Gruber T, Olsen G (1994) An Ontology for Engineering Mathematics. Technical Report KSL-94-18, Stanford University
18. Baur J (1999) Syntax und Semantik Mathematischer Texte. Master's thesis, Universität des Saarlandes, Fachbereich Computerlinguistik
19. Wolfram Research: MathWorld. http://mathworld.wolfram.com
20. Fellbaum C (1998) WordNet: An Electronic Lexical Database. MIT Press, Cambridge, London
21. Fellbaum C WordNet. http://wordnet.princeton.edu
22. GermaNet Team. GermaNet. http://www.sfs.uni-tuebingen.de/lsd/english.html
23. Helbig H (2006) Knowledge Representation and the Semantics of Natural Language. Springer, Berlin
24. Urban J (2006) MoMM—Fast interreduction and retrieval in large libraries of formalized mathematics. International Journal on Artificial Intelligence Tools 15(1):109–130
25. Urban J (2005) MizarMode—An integrated proof assistance tool for the mizar way of formalizing mathematics. Journal of Applied Logic doi:10.1016/j.jal.2005.10.004
26. Urban J (2002) XML-izing Mizar: Making Semantic Processing and Presentation of MML Easy, MKM2005
27. Pinkall M, Siekmann J, Benzmüller C, Kruijff-Korbayova I. DIALOG. http://www.ags.uni-sb.de/~dialog/
28. Asperti A, Padovani L, Sacerdoti Coen C and Schena I (2001) HELM and the semantic web. In: Boulton RJ, Jackson PB, eds. Theorem Proving in Higher Order Logics, 14th International Conference, TPHOLs 2001, Edinburgh, Scotland, UK, September 3–6, 2001, Proceedings. Volume 2152 of Lecture Notes in Computer Science, Springer
29. Asperti A, Zacchiroli S (2004) Searching mathematics on theWeb: State of the art and future developments. In: Joint Proceedings of the ECM4 Satellite Conference on Electronic Publishing at KTH Stockholm, AMS–SM M Special Session, Houston
30. Albayrak S, Wollny S, Varone N, Lommatzsch A, Milosevic D (2005) Agent technology for personalized information filtering: the PIA-system. ACM Symposium on Applied Computing
31. Collier N, K T. May (2002) PIA-core: Semantic annotation through example-based learning. Third International Conference on Language Resources and Evaluation, pp 1611–1614
32. The PIA Project: PIA. http://www.pia-services.de/

Chapter 22
Investigation of the Effectiveness of Tag-Based Contextual Collaborative Filtering in Website Recommendation

Reyn Nakamoto, Shinsuke Nakajima, Jun Miyazaki, Shunsuke Uemura, and Hirokazu Kato

22.1 Introduction

As the Internet continues to mature and become more accessible to the common user, the amount of available information increases exponentially. Accordingly, finding useful and relevant information is becoming progressively difficult. Moreover, a lot of the information available—blogs, various types of reviews, and so forth—is highly subjective and thus, hard to evaluate purely through machine algorithms. Being subjective in nature, one person may absolutely love something while the next may loathe the same—no single authority exists. It is in these cases where people—more so than the current ability of machine algorithms—are greatly effective in evaluating and filtering this information.

For this reason, the idea of collaborative filtering (CF) was started, extensively researched, and eventually deployed to relatively good amounts of success. Using the people and the community, recommendations of subjective information can be made through the matching of similar users. Sites such as amazon.com [1] or movielens [6], etc. utilize such recommendation methods, matching users based upon their ratings and then producing recommendations. Through this, CF provides personalized recommendations to the users, while at the same time offering the ability to deal with subjective material. However, the failing of CF is that it does not consider why a user likes something and what the user is interested in now. In other words, CF can recommend relevant sites, but does not know why or when it should be appropriate.

Similarly, online social tagging systems also employ the masses to evaluate and describe information. Instead of relying purely upon machine algorithms, people

Reyn Nakamoto, Shinsuke Nakajima, Jun Miyazaki, and Hirokazu Kato
Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara, Japan
e-mail: {reyn-n,shin,miyazaki,kato}@is.naist.jp

Shunsuke Uemura
Faculty of Informatics, Nara Sangyo University, Ikoma, Nara, Japan
e-mail: UemuraShunsuke@nara-su.ac.jp

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

themselves describe some resource—whether they be photos, videos, websites—by using tags, or natural language keywords. People are motivated by differing reasons to tag, but the end result is that these resources become easily discoverable through searching through the metadata provided by tags. These tags provide the who, what, when, where and why—they essentially describe that resource, and at the same time, the reason why it was liked and subsequently tagged. Tagging however, fails to provide what CF does—it has yet to provide a system for producing personalized recommendations.

By combining the advantages of the two systems, we have tag-based contextual collaborative filtering (TCCF), as previously described in [8]. By utilizing personalized recommendations provided by CF and the context provided by tags, TCCF aims to provide effective contextual, personalized recommendations.

In this chapter, we describe the website recommendation system we built using TCCF as a recommendation method and tested against three other recommendation/search methods—plain CF, pure tag searching, and CF with tag filtering. From our user testing, TCCF showed itself to be the most effective method of the group, leading the other recommendation methods in both recommendation precision and in users' general impression of the methods. We show these results and discuss the implications in this chapter.

22.2 Related Work

22.2.1 Collaborative Filtering Systems

Collaborative filtering (CF) is the process whereby the community of users is used to sort out relevant or important information from the nonrelevant or nonimportant information. The process is based upon the idea that if users prefer the same item or items, then their preference will be similar for other items liked by similar users. In other words, a user should like the same items that similar users like. From a wider perspective, once users have recorded their preferences within the systems, subsequent users can benefit from the knowledge within them—hence the collaborative aspect of the method.

CF has been proven to work well under certain domains—mainly entertainment domains—such as usenet recommendations [9], movie recommendations [6], product recommendations [1], and so forth.

Many CF systems rely upon a matrix of numerical ratings from users against some resource. Once enough ratings are in place, a similar rating is calculated between the user and other users. Using this, recommendation can be made by calculating the average of similarity ratings times the ratings other users recorded, and then recommending those resources that have scores above a certain threshold.

However, using only numerical values tells only if a user likes something or not—not *why* a user likes something. Thus, in cases where domains are large, this often leads to issues—for example, while two users may have a similar interests

in one topic, they may not share the same for another. In particular, domains like Internet websites fall subject to this—users usually have many topics of interest and matching all interests is very difficult using numerical values alone. In addition to this, users are not always in the mood to see all of their interests—rather, they may be only interested in one or two on that day.

22.2.2 Social Tagging Systems

Tagging has been around for some time, albeit known by other terms such as metadata, categorization, labels, and so forth. Tagging is the process of attaching natural language words as metadata to describe a resource like a movie, photo, book, etc. Vocabulary for tagging is usually uncontrolled, so the users themselves can decide what word or combination of words are appropriate.

The current main use of tagging is for the purpose of retrieval, whereby users can search for a tag and the resources tagged with that tag will be returned to the user. In the case of the user who added the tag, they can use tags for later retrieval. For other users, tags serve as a way to discover new resources by searching for whatever tag they are interested in.

In recent years, the advent of social tagging systems have brought tagging back into the limelight. Currently, there are several online social tagging systems that are popular and are the subject of continuing research. They range from website bookmarking such as del.icio.us [4], photo sharing [5], research papers [2], and even people [3]! All of these sites use tagging for many purposes, but in addition to that, they focus on the social networking aspects of tagging to enhance the experience for end users. In their present form, however, tags are generally used for tag searching—user profile matching and subsequent recommendations through this are yet to implemented.

As mentioned before, tags provide the clues as to the context in which a user liked something. These tags are used for several different purposes, including denoting the subject itself, the category, or the refining characteristics of the resource [7]—for example, a picture of a dog would most likely be tagged something like *dog*, *animal*, or maybe *cute*. Thus, tags seem to provide the missing link in CF: they provide the who, what, when, where, and why of a resource—in other words, the context in which the user liked, and subsequently tagged, a resource for later retrieval. Because of this and the similar use of social networking, social tagging systems provide a ideal choice for combination with CF systems.

22.3 TCCF Website Recommendation System

TCCF is the combination of traditional CF systems and social tagging systems that allow for personalized, contextual recommendations. The essential idea is that CF provides personalization, and tags provide the *context* of the users' preferences. We

use *context* in the following two ways—first, context as in why a user liked something or why they took the time to tag something. Secondly, we use context as in the user's current state—what interest the user wants to see now. In the first case, it is important to ascertain why the user liked something. Doing so allows for more accurate and more personalized recommendations. In the second case, users often have many interests, but they do not always wish to view all of them all the time. Is it a necessity that we consider what the users' current state is to provide better recommendations. TCCF addresses both of these issues by combining the CF and tagging.

Unlike traditional CF models that use numeric ratings, our TCCF model also uses tags as an indicator of why a user likes something. For example, say we have a website bookmarking system where users can come in and bookmark websites that they enjoy using tags. Normally, the act of bookmarking a website is a strong indicator of whether something is liked. However, the used tags provide the key distinguishing factor from traditional CF systems—the tags attached the resource can be seen as the context in which the user likes the resource. Usually, the user will use tags to describe the resource as the user sees it, and in most cases it would be the context of why they liked something. Thus, from this assumption, we build upon incorporating tags along with CF to provide effective personalized, contextual information recommendations.

We now describe our TCCF method. We explain our method as it is used in our website bookmarking system. In this system, users bookmark websites they like using tags, and subsequently, they can easily retrieve their bookmarks by just searching by the tags. A example scenario is shown in Fig. 22.1. Here, users *A*, *B*, *C*, and *D* are bookmarking the websites they like using tags.

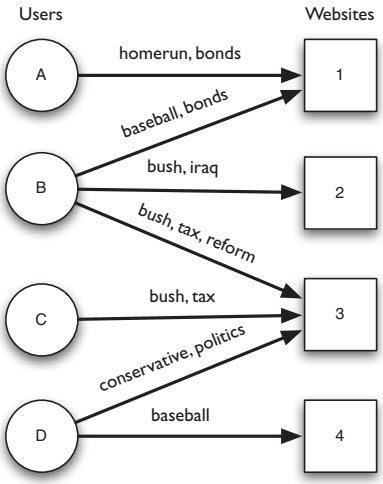


Fig. 22.1 Contextual CF model

22.3.1 TCCF User Similarity Model

Like CF, user similarity is first calculated to determine which users are similar and subsequently, those similar users' preferences are used to find recommendation candidates. Our TCCF user similarity model is based upon both commonly bookmarked websites as well as the tags that they used to bookmark. The TCCF user similarity model between two users A and B is shown in Eqn. 22.1.

$$sim_{ccf}(A,B) = \frac{1}{2n}\sum_{k=1}^{n}\{sim(T_{A\to k}, T_{B\to k}) + 1\} \tag{22.1}$$

where $sim(T_{A\to k}, T_{B\to k})$ is the cosine of the tag vectors that user A and B used to bookmark the same website. Essentially, the common bookmark's tag vectors from users A and B are compared. The same is done with all the common bookmarks A has with B. Those values are then averaged to generate the user similarity score. The addition of one in this equation is the incorporation of standard CF—value is given for having common bookmarks, regardless of whether the tag vectors match or not.

For example, in Fig. 22.1, user B, C, and D all bookmarked website 3. However, the similarity score between users B and C would be higher than C and D's because B and C used similar tags to bookmark the same website. Incidentally, C and D's are still higher than A and C due to the existence of a common bookmark.

22.3.2 TCCF Score Prediction Model

Also similar to CF, results are based upon a score prediction that the system generates. Score prediction is basically the numeric representation of how well the system thinks the user will like some resource. The TCCF score prediction model for a website x's score for a user A is as shown in Eqn. 22.2.

$$score_{pred}(A,x) = \frac{\frac{1}{2}\sum_{k=1}^{n}\{sim_{ccf}(A,S_k) * (\max(sim(T_{S_k\to 1}, T_{S_k\to x}), \ldots, sim(T_{S_k\to m}, T_{S_k\to x})) + 1)\}}{\sum_{k=1}^{n} sim_{ccf}(A,S_k)} \tag{22.2}$$

Essentially, all of similar user S_k's bookmarks are considered as recommendation candidates. Each of these candidates' sites' tag vectors are then compared with each of the tag vectors of the common bookmarks that similar user S_k has with user A. The maximum value of these comparisons is taken and then multiplied by $sim_{ccf}(A,S_k)$—the user similarity score between users S_k and A. The process is repeated for all similar users and averaged to form the score predictions. Again, the addition of one in this equation is the incorporation of standard CF to give value to the existence of a similar user's bookmark, regardless of the similarity of the tag vectors.

For example, in Fig. 22.1, user *B* and *C* are similar. Thus, because user *B* has website 1 and website 2 bookmarked, they are candidates for recommendation. However, since website 2's tag vector is similar to the commonly bookmarked website 3's tag vector, its score prediction will be higher than that for website 1. Website 1 could still be recommended, but due to its dissimilar tag vector, it would be ranked lower than website 2.[1]

22.3.3 System Design

Our system is designed around a website bookmarking system not unlike del.icio.us [4]. It has the same basic feature of bookmarking websites using tags as opposed to the traditional directory structure that most browsers use. Users can bookmark websites using whatever tags to describe the website to themselves, and similarly they can search through their bookmarks using the same tags. Additionally, users can discover other peoples bookmarks by searching through all the bookmarks within the system. By providing features as shown in del.icio.us, users were provided an easy motivation to use the system. Many of the users reported having found many interesting links through these base features themselves.

The interface to the system was done through a Firefox browser plug-in as shown in Fig. 22.2. While browsing, the user can easily access the features of the system. They can quickly search for bookmarks as well as easily add new bookmarks.

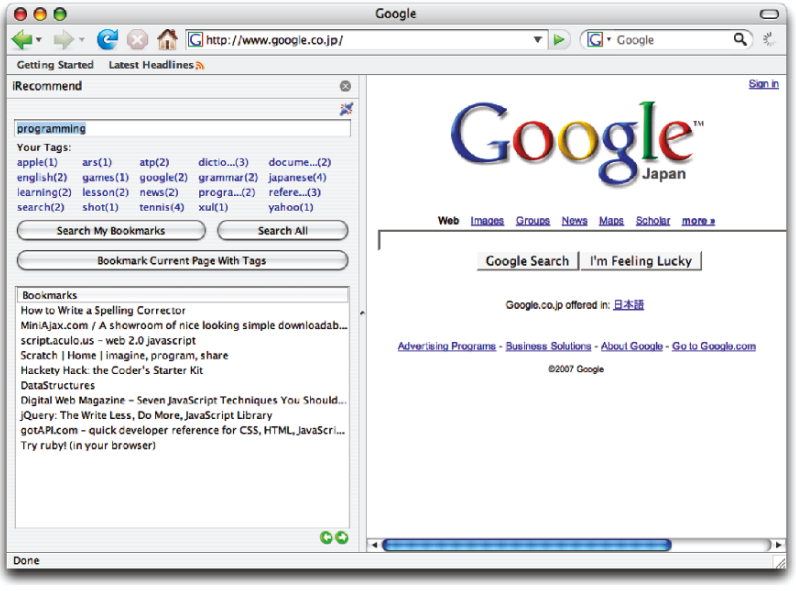


Fig. 22.2 System interface

[1] For further explanation of TCCF, see [8].

All data is stored server-side. Additionally, recommendations are also calculated server-side.

22.4 User Testing

The main goal of the testing was to determine the effectiveness of our TCCF algorithm versus already established algorithms. Furthermore, we wanted to emphasize the importance of context when recommending. Thus, recommendations were done on a *pull* basis, where recommendation results are generated at the user's request. This simulates users requesting only the topic they are interested in now. When they search through their bookmarks using tags, up to five recommendation results appear below the user's bookmarks. Depending on the algorithm, these recommendations are generated based upon the searched tags and/or the user's profile.

We tested our TCCF algorithm versus three other search and recommendation methods: plain collaborative filtering (CF), tag searching (Tag), and lastly collaborative filtering with tag filtering (TagCF).

- CF is basic collaborative filtering. First, user similarity is calculated based upon the user profile, using the number of common bookmarks that a user has with another. The higher the number, the higher the user similarity was. Following this, ranking scores for the websites were generated by averaging the ratings of similar users times their respective user similarities.
- Tag searching is akin to popularity-based searching. Basically, the system retrieves all the bookmarks that have been tagged by a certain tag or set of tags. The results are then ordered by the number of users that have that website bookmarked.
- TagCF is CF with tag filtering. CF recommendations are calculated as done with basic CF. Then, instead of displaying the results as is, the system displays only the results that have the tag that the user searched for.
- Lastly, TCCF is as described in Section 22.3. Score predictions were generated through the TCCF model and then only those results that were linked to the searched tag would be displayed.

A total of nine users were selected to participate in testing.

22.4.1 Test Procedure

Users were first asked to bookmark twenty or more webpages to build their user profile. After this, the testing of each recommendation method took place. For each test session, one of the four recommendation methods were chosen at random. They followed the following procedure:

1. Users selected 10 or more tags of their liking.
2. For each of the selected tags, the system generated up to five recommendations. For each of these recommendations, they reviewed the website.
3. If the website was useful to them *in the context of the tag's meaning*, they pushed the *yes* button that appeared in the interface after selecting a recommendation. If not, they would push the *no* button instead.
4. After finishing rating the recommendations, users completed a general survey regarding the session's recommendation method.

22.4.2 Testing Results

Does TCCF give effective recommendations?

We examine whether our TCCF method provides effective recommendations to the user. We asked the user to determine whether the recommended website satisfied the following two conditions:
1. The website is useful to them.
2. The website's content matches the meaning of the tag.

Thus, if the user was requesting *soccer*, but the system generated a useful, but unrelated tennis site, they were to vote *no*. Again, users have many different interests, and a contextual recommendation system should account for this. The precision of top five recommendation results for each method is shown in Fig. 22.3.

In this case, precision is the number of relevant results divided by the number of all results retrieved—i.e. the number of *yes* ratings divided by the total number of ratings.

As can be seen, our TCCF algorithm received the highest precision of all the four selected algorithms. It is followed by TagCF, then Tag, and finally CF.

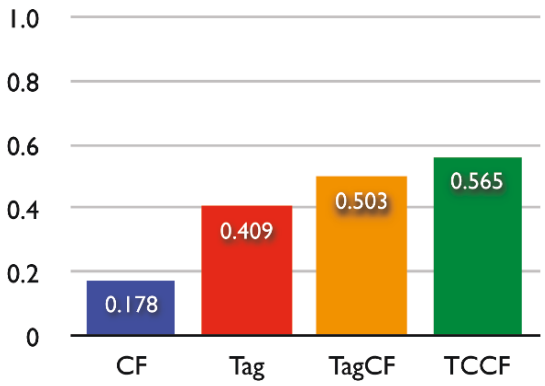


Fig. 22.3 Precision of the top five recommendations

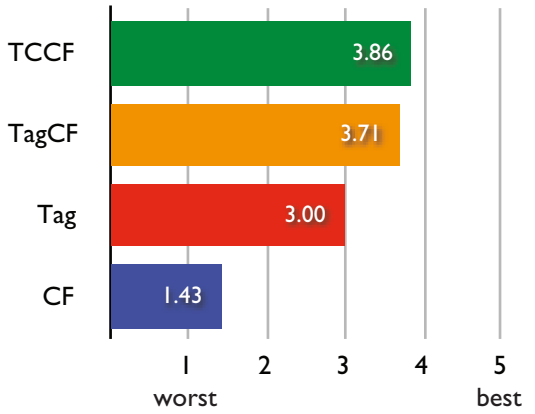


Fig. 22.4 Overall user satisfaction

Does TCCF increase user satisfaction?

From user testing, it would appear that TCCF completes its goal of recommending relevant recommendations when context is considered. Users were also asked to complete a survey after the recommendation rating step was completed. They were asked for their general impression of the algorithm on a scale from 1-5, with five being the best. The results are as shown in Fig. 22.4.

This figure also shows that TCCF comes out on top, although not as far ahead as the precision results. Overall, users were subjectively satisfied with TCCF along with TagCF. Tag searching takes the middle ground again, while CF took last place again.

22.5 Results Discussion

From these results, TCCF provides better recommendations when context is considered. On the other hand, CF does not consider context at either stage. Thus, even though it provided personalized recommendations, CF does not accommodate for changing tag requests. The Tag method considers context in that it produces results related to what the users are interested in now. However, because it does not contain personalization, nor does it consider the context in which the user liked a website, this probably affects why its score is lower. Lastly, TagCF performs reasonably well—it has personalization and considers what the user is interested in at the current moment. It does not, however, consider why the user originally liked a website and this is likely a contributing factor as to why it did not perform better.

Overall, context and personalization are important factors. TCCF accounts for both, and thus, produces more useful results and gives users higher level of satisfaction.

22.6 Conclusions and Future Work

We have described a website bookmarking system using our TCCF algorithm. Our user testing has shown that TCCF is effective in providing effective personalized recommendations when the context is considered. Additionally, user satisfaction was shown to be the highest for TCCF when compared to CF, Tag, and TagCF. From this, it would seem that personalization as well as considering the context is very important when making recommendations to users.

In the future, we plan to further refine the recommendation method. Currently, no score threshold is set for recommendations for the user. Thus, exploration into how high the threshold should be would further improve TCCF's accuracy. We also plan to experiment with altering the weights of tags and CF in the algorithm to find the optimal balance between the two. It is now assumed to be equal. Last, we also intend to experiment with natural language processing to aid in comparing tag vectors with each other.

In terms of system refinement, usability tests must be completed to fully gauge the usefulness versus other information search/recommendation methods. In our test, users were selected and required to follow a set procedure. However, further tests must be completed to find out how the users would react in open-ended tests. We must also determine whether the users would continue to like the system over other currently available search/recommendation methods.

Acknowledgements This research was partly supported by CREST of JST and MEXT (Grant-in-Aid for Scientific Research on Priority Areas #19024058).

References

1. Amazon.com, http://www.amazon.com/
2. Citeulike, http://www.citeulike.org/
3. Consumating.com, http://www.consumating.com/
4. del.icio.us, http://del.icio.us/
5. Flickr, http://www.flickr.com/
6. movielens, http://www.movielens.umn.edu/
7. S. Golder and B. Huberman. The structure of collaborative tagging systems. http://www.hpl.hp.com/research/idl/papers/tags/ tags.pdf
8. Nakamoto R, Nakajima S, Miyazaki J, Uemura S (2007) Contextual collaborative filtering based on tagging information. In *Proceedings of The International MultiConference of Engineers and Computer Scientists* pages 964–969
9. Resnick P, Iacovou N, Suchak M, Bergstorm P, Riedl J (1994) GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM Conference on Computer Supported Cooperative Work* pages 175–186, Chapel Hill, North Carolina. ACM. http://citeseer.ist.psu.edu/resnick94grouplens.html

Chapter 23
Context-Aware Ranking Method for Information Recommendation

Kenta Oku, Shinsuke Nakajima, Jun Miyazaki, Shunsuke Uemura, and Hirokazu Kato

Previously, we proposed two recommendation systems—context-aware information filtering (C-IF) and context-aware collaborative filtering (C-CF)—both of which are context-aware recommendation methods. We have also shown their effectiveness through of experiments using a restaurant recommendation system based on these methods. Furthermore, we need to rank the recommended contents to improve the performance of the C-IF and the C-CF. A ranking method ranks the recommended contents based on content parameters that the user regards as important. However, what parameter is important for the user is according to their contexts. For example, a user likes a reasonable restaurant when he is alone, but on the other hand, he likes an expensive and stylish restaurant when he is with his girlfriend. Therefore, it is important to consider his contexts when ranking recomended contents. In this study, we propose a context-aware ranking method. The system ranks recommended contents within users' contexts. We also evaluate our proposal method from experimental results.

23.1 Introduction

There have been many studies on information recommendation methods that provide users with information matching their preferences. In doing so, it is important to consider users' contexts when recommending, because their preferences change with context. However, currently there are few recommendation methods that consider user contexts.

Kenta Oku, Shinsuke Nakajima, Jun Miyazaki, and Hirokazu Kato
Nara Institute of Science and Technology
e-mail: {kenta-o,shin,miyazaki,kato}@is.naist.jp

Shunsuke Uemura
Nara Sangyo University
e-mail: uemurashunsuke@nara-su.ac.jp

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

In recent years, there have been many studies of context-aware applications with the popularization of mobile computers such as PDAs and cellular phones [1]. Shopping Assistant [2] is one that provides details of items based on shoppers' location in a store, and Cyberguide [3] and GUIDE [4] provide information such as guides, tourist resorts, and exhibits based on travelers' current location. They use only time of day and location as contexts; there are few studies that consider various contexts that influence users' decisions.

On the other hand, we think that various contexts such as relationships with companions, budget, and weather, in addition to time of day and location—are necessary for taking every context into consideration when giving recommendations. In this study, we define user contexts as situations around the user (e.g., time of day, weather, budget, companions, and so on) that influence the users' decisions.

Previously, we proposed two context-aware recommendation systems, the context-aware information filtering (C-IF) and the context-aware collaborative filtering (C-CF). We have already shown that it is effective to consider users' contexts in information recommendation through experiments using a restaurant recommendation system based on the C-IF and the C-CF in our previous work [5]. We also have discussed the following points to develop a practical context-aware recommendation system [6]:

a. To verify the appropriateness of adopting an SVM to a context-aware recommendation method by comparing it with other classifiers
b. To clarify the advantages and disadvantages between C-IF and C-CF to various user's situations
c. To consider how to optimize the parameters of feature vectors for target contents and user contexts

Both the C-IF and the C-CF can make recommendations that users may be satisfied with. To improve the performance of these systems, it is more effective to make rankings for the recommendations. A simple way to make the ranking is to calculate ranking scores based on euclidean distance from a decision plane of an SVM. However, the parameters that are important for them are based on to users' contexts. For example, a user likes a reasonable restaurant when he is alone—on the other hand, he likes an expensive and stylish restaurant when he is with his girlfriend. Therefore, it is important to consider his contexts when making rankings for recommendations.

In this study, we propose a context-aware ranking method. The method learns weight for restaurant parameters considering users' contexts. We also evaluate the method from experimental results.

First, the C-IF and the C-CF are explained in Sect. 23.2. Then, recommendation ranking is described in Sect. 23.3. We did some experiments to evaluate our systems. In Sect. 23.4, we verify the SVM's generalization performance based on comparison experiments with other classifiers' performance. In addition, three experimental results are shown in Sect. 23.5. Finally, we conclude this study.

23.2 Context-Aware Recommendation System

23.2.1 Context-Aware Information Filtering (C-IF)

A C-IF can recommend information to users adapting their current contexts.

Figure 23.1 shows a concept diagram of this C-IF. The following lists the procedure of recommending contents to users by using the C-IF.

1. The m-dimensions $(c_1, c_2, \ldots, c_m)$ that can express a target user's contexts are added to the feature space of a simple SVM (context-dependent user preference modeling; CP-Modeling) as shown in Fig. 23.1.
2. The CP-Modeling learns the user's training data (positive data ◯ and negative data △ including their contexts, and creates a model (CP-Model) as a decision plane dividing positive and negative classes.
3. If untrained data exists in the positive class of the CP-Model in the user's current context, the data item is regarded as recommendation candidate (for example, a point ☆ in Fig. 23.1).

A simple SVM can learn a user's preferences while considering contexts that they have experienced by making decision models corresponding to each context. However, it is inefficient to keep many SVM models for one user. Moreover, there are no decision planes for unlearned contexts that the user faces for the first time.

On the other hand, the CP-Model can learn a user's preferences while considering contexts that they have experienced by making only one decision model. Furthermore, the CP-Model is valid for unlearned context that the user faces for the first time.

In this chapter, *content feature parameters* denote parameters of feature vectors representing features of contents, and context feature parameters denote parameters of feature vectors representing features of contexts. Table 23.1 shows the content feature parameters in the case of restaurant recommendation. Table 23.2 shows the context feature parameters used in this study. The numbers in () in Table 23.1 and Table 23.2 denote the number of parameter dimensions.

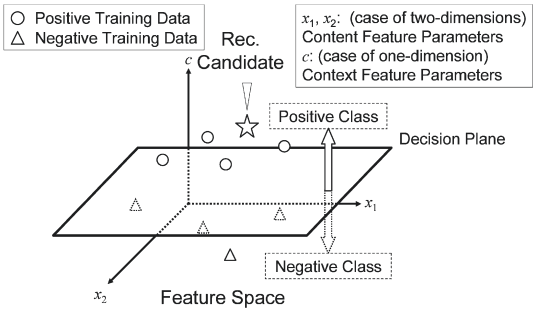


Fig. 23.1 Concept diagram of C-IF

Table 23.1 Restaurant feature parameters (28 dimensions)

Parameter Group	Parameter Value
Genre (8)	[Japanese] / [Western] / [Chinese] / [Bar] / [Asian] / [Lamian] / [Cafe] / [Others]
Budget (Day/Night) (2)	0-10,000 yen
Characteristic (11)	[Single Room] / [Car Park] / [Non-smoking Section] / [Karaoke] / [Open Late at Night] / [All You Can Eat] / [Lunch] / [Card Available] / [Live] / [Takeout] / [Reserved Seat]
Atmosphere (7)	[For Dates] / [For Business Receptions] / [For Parties] / [For Joint Parties] / [For Familiy] / [For One Person] / [For a Few People]

Table 23.2 Context feature parameters (24 dimensions)

Parameter Group		Parameter Value
Time Info.	Month (2)	Jan. - Dec.
	Day of Week (2)	Sun. - Sat.
	Time (2)	00:00 - 23:59
Weather Info.	Weather (4)	[Fine] / [Cloudy] / [Rainy] / [Snow]
	Temperature (1)	0 - 40°C
User Info.	Holiday (2)	[Holiday] / [Before Holiday]
	Budget (1)	0 - 10000 yen
	Spare Time (1)	0 - 4 hours
Companion Info.	Group (7)	[Alone] / [Family] / [Boy/Girl Friend] / [Friends] / [Boss/Subordinate] / [Business] / [Others]
	Number of Males/Females (2)	0 - 10

23.2.2 Context-Aware Collaborative Filtering (C-CF)

C-CF is collaborative filtering using the CP-Model. The similar users used for C-CF are based on the similarity between the decision planes of their CP-Models. Figure 23.2 shows how to determine similar users in C-CF and how to recommend contents to a target user by using the C-CF. The following describes the procedure for deciding similar users by applying the C-CF. Suppose that user A is a target user, and user B is a user similar to user A.

1. User A has positive (◯) and negative (△) training data, and user B has positive (●) and negative (▲) training data for some restaurant data as in Fig. 23.2(a).
2. Swap training data of user A and user B with each other as in Fig. 23.2(b).

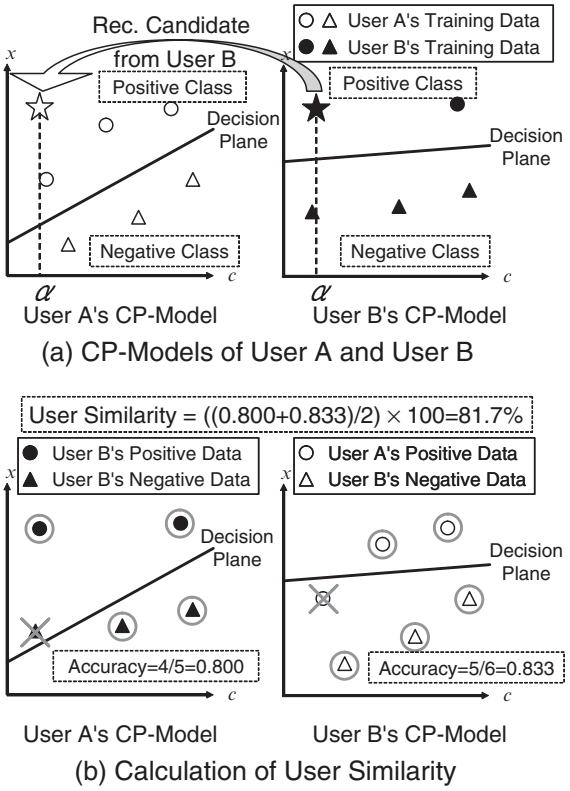


Fig. 23.2 Recommendation using C-CF

3. The accuracy of user B's (A's) data by user A's (B's) model is calculated based on whether user B's (A's) positive (negative) data are judged to be in the positive (negative) class of user A's (B's) model (in the case of Fig. 23.2(b), the accuracy of user B's data by user A's model is $4/5$, and the accuracy of user A's data by user B's model is $5/6$).
4. Calculate user similarity as the average of each accuracy, then two users whose user similarity is over a certain threshold are regarded as similar users.

Suppose that user B has already been classed as *positive* for a sample (for example, a point ★ in Fig. 23.2(a)). Then, the sample ★ also exists in the positive class of user A's CP-model in context α (a point ☆ in Fig. 23.2(a)). Therefore, when user A requests contents in context α, the sample ☆ is regarded as a recommendation candidate.

The conventional collaborative filtering requires the use of common data between users to calculate user similarity, whereas C-CF does not necessarily use common data between users.

23.3 Recommendation Ranking Based upon Adjusted Parameter Weighting

23.3.1 Basic Idea

Figure 23.3 shows examples of the amount of positive data and negative data for a parameter of training data. The figure shows three patterns for a parameter Budget [(a), (b) and (c)]. The horizontal axis denotes the parameter value. The x_{budget}—a parameter value of the Budget—is represented by a continuous value $\{x_{budget} : 0 \leq x_{budget} \leq 1\}$. The value means normalized value from 0 to 10,000 yen. Each training data has some value for the Budget parameter. In these cases, training data has parameter value as shown in the upper side in the figure. Here, ◯ and ✕ denote positive data and negative data, respectively.

In the case of (a), the positive data are distributed in the range of $\{0 \leq x_{budget} < 0.3\}$, and the negative data are distributed in the range of $\{0.3 \leq x_{budget} < 0.7\}$. In the same way, in the case of (b), the positive data are distributed in the range

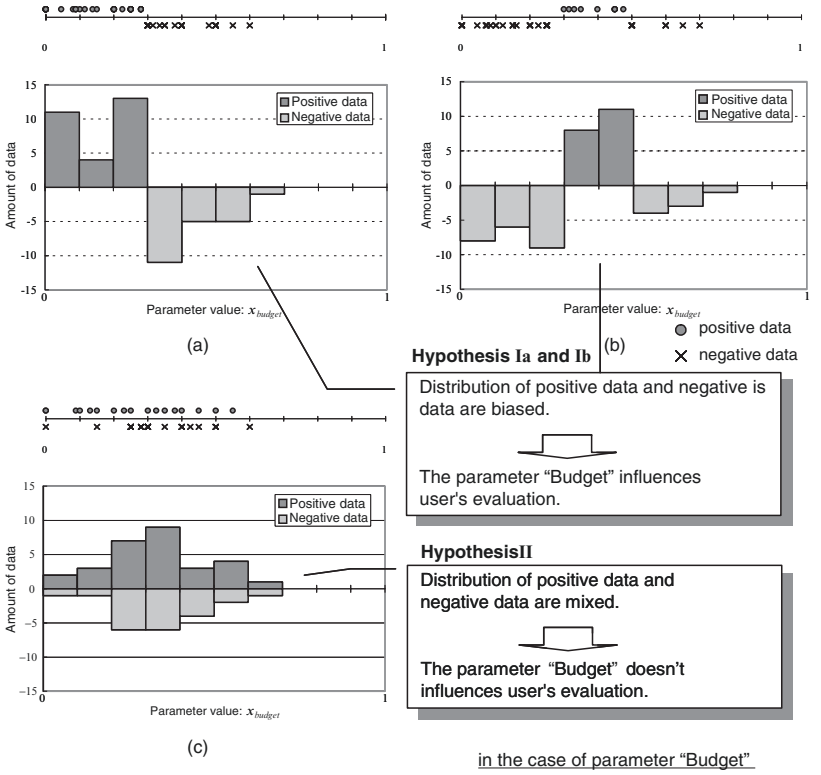


Fig. 23.3 Examples of the amount of positive data and negative data for a parameter of training data

of $\{0.3 \leq x_{budget} < 0.5\}$, and the negative data are distributed in the range of $\{0 \leq x_{budget} < 0.3, 0.5 \leq x_{budget} < 0.8\}$. In these cases, the distribution of positive and negative data are biased. From such biased data distribution, we can see that the Budget parameter influence users' evaluation. On the other hand, in the case of (c), the positive and negative data are mixed. Therefore, from such mixed data distribution, we can see that the Budget parameter does not influence users' evaluation.

Based on the above examples, we set up the following hypothesises about data distribution for each parameter:

Hypothesis Ia: Positive data distribution is biased. ⇒ The parameter influences the user's evaluation of how she is satisfied with restaurants.

Hypothesis Ib: Negative data distribution is biased. ⇒ The parameter influences the user's evaluation of how she is not satisfied with restaurants.

Hypothesis II: Distribution of positive data and negative data are mixed. ⇒ The parameter does not influence the user's evaluation.

In this way, it is possible to know how the user regards the parameter as important by investigating the bias of distribution of positive and negative data.

23.3.2 Learning of Parameter Weight

In this section, we describe a method that learns the weight for each parameter from training data. The weight corresponds to how the user regards the parameter as important. Our method learns the weight according to the following methods, which are based on the hypothesises described in Sect. 23.3.1:

Method Ia: The data distribution conforms to Hypothesis Ia. ⇒ Make the positive weight for the parameter larger ($w \rightarrow +1$).

Method Ib: The data distribution conforms to Hypothesis Ib. ⇒ Make the negative weight for the parameter larger ($w \rightarrow -1$).

Method II: The data distribution conforms to Hypothesis II. ⇒ Make both the positive weight and the negative weight 0 ($w = 0$).

Here, w corresponds to weight for a parameter. When w is close to +1, the parameter is regarded as important for the user's satisfaction. On the other hand, when w is close to -1, the parameter is regarded as important for the user's dissatisfaction.

Furthermore, the user's weight for a parameter is not always equally in the range of the parameter value $\{x : 0 \leq x \leq 1\}$. In the example of Fig. 23.3, the user is satisfied with restaurants whose budget is less than 3,000 yen, while is dissatisfied with those whose budget is 3,000 yen or more. Put another way, the user has positive weight for the parameter when $x_{budget} < 0.3$, yet negative weight when $x_{budget} \geq 0.3$. Thus, it is needed to adjust weight for each parameter according to the parameter value.

This method mainly consists of the following two steps:

1. Calculate positive weight and negative weight for parameters.
2. Adjust weight according to parameter values.

23.3.3 Calculation of Positive Weight and Negative Weight for Parameters

In this section, we explain our method using an example for a Budget parameter which is represented by a continuous value (Fig. 23.4). In the same way as Fig. 23.3, the horizontal axis $\{x_{budget} : 0 \leq x_{budget} \leq 1\}$ denotes the parameter value. Eleven data are distributed for the Budget parameter as shown in Fig. 23.4. Each data is evaluated by giving positive or negative by a user.

(1) Investigate how data distribution is biased in the range of $\{x_{budget} : 0 \leq x_{budget} \leq 1\}$.
(2) Give r (an imaginary range of data) to each data. Then, judge any overlapping of the range r of data with each other.
(3) Count the following amount of data, referring to Fig. 23.4.

m^{++} : *The amount of overlapping positive data with positive data.*

m^{--} : *The amount of overlapping negative data with negative data.*

m^{+-} : *The amount of overlapping positive data with negative data.*

Here, m^{++}_{max}, m^{--}_{max}, and m^{+-}_{max}—the maximum amount of overlapping data—are defined as the amount of them when $r = 1$, respectively.

(4) Calculate the following rates corresponding to the rates of the amount of overlapping data in the maximum amount of overlapping data.

$$p^{++} = m^{++}/m^{++}_{max}$$
$$p^{--} = m^{--}/m^{--}_{max}$$
$$p^{+-} = m^{+-}/m^{+-}_{max}$$

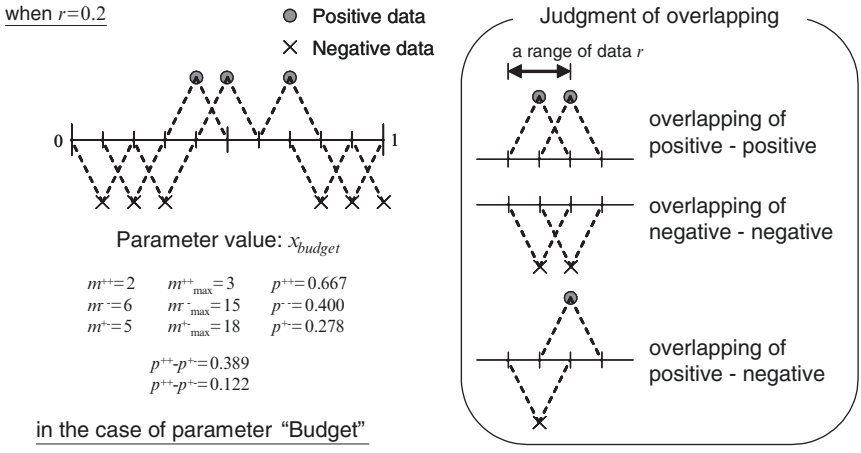


Fig. 23.4 Investigation of how distribution of positive data and negative data are biased

If p^{++} (or p^{--}) is larger while r is small, the distribution of postive data (or negative data) is biased for that. On the other hand, if p^{+-} is larger, both positive data and negative data are mixed for that.

(5) Find r^{+*} (or r^{-*}), which is the range of data when $p^{++} - p^{+-}$ (or $p^{--} - p^{+-}$) is maximized in the range of $\{r : 0 \leq r \leq 1\}$.

These $p^{++} - p^{+-}$ and $p^{--} - p^{+-}$ correspond to the user's weight for the parameter. Then we can put them as follows:

$$\text{When } r = r^{+*}$$

$$w^{+} = \begin{cases} p^{++} - p^{+-} & (\text{When } p^{++} - p^{+-} > 0) \\ 0 & (\text{When } p^{++} - p^{+-} \leq 0) \end{cases}$$

$$\text{When } r = r^{-*}$$

$$w^{-} = \begin{cases} p^{--} - p^{+-} & (\text{When } p^{--} - p^{+-} > 0) \\ 0 & (\text{When } p^{--} - p^{+-} \leq 0) \end{cases}$$

23.3.4 Adjustment of Weight According to Parameter Values

Figure 23.5 shows a process for adjusting weight according to the parameter values shown in Fig. 23.4. We explain the process using Fig. 23.5.

(1) Make clusters consisting of overlapping positive data (or negative data) when $r = r^{+*}$ (or $r = r^{-*}$). In the case of the figure, three clusters ("+1", "−1" and "−2") are made.
(2) Plot normal distribution $N(\mu, \sigma)$ for each cluster on the condition that each data follow the normal distribution. In the case of the figure, three normal distributions $\left(N^{+1}(0.533, 0.153),\ N^{-1}(0.2, 0.1) \text{ and } N^{-1}(0.9, 0.1)\right)$ are plotted. Here, the normal distribution of the cluster by positive data is plotted on the positive side, and then by negative data are plotted on the negative side for convenience.
(3) Normalize the normal distribution by replacing the maximum of normal distribution of the cluster by positive data (or negative data) with w^{+} (or w^{-}).
(4) Compound all distribution.

We define $w(x)$—weight distribution for parameter values—as such compounded distribution. This weight distribution has the following characters:

- $w(x) > 0 \Rightarrow$ The parameter value x influences the user's satisfaction.
- $w(x) < 0 \Rightarrow$ The parameter value x influences the user's dissatisfaction.
- $w(x) = 0 \Rightarrow$ The parameter value x does not influence the user's evaluation.

The example in Fig. 23.5 shows that the user is satisfied with restaurants whose budget is from 3,000 to 8,000 yen, and is dissatisfied with them whose budget is less than 3,000 yen or more than 8,000 yen.

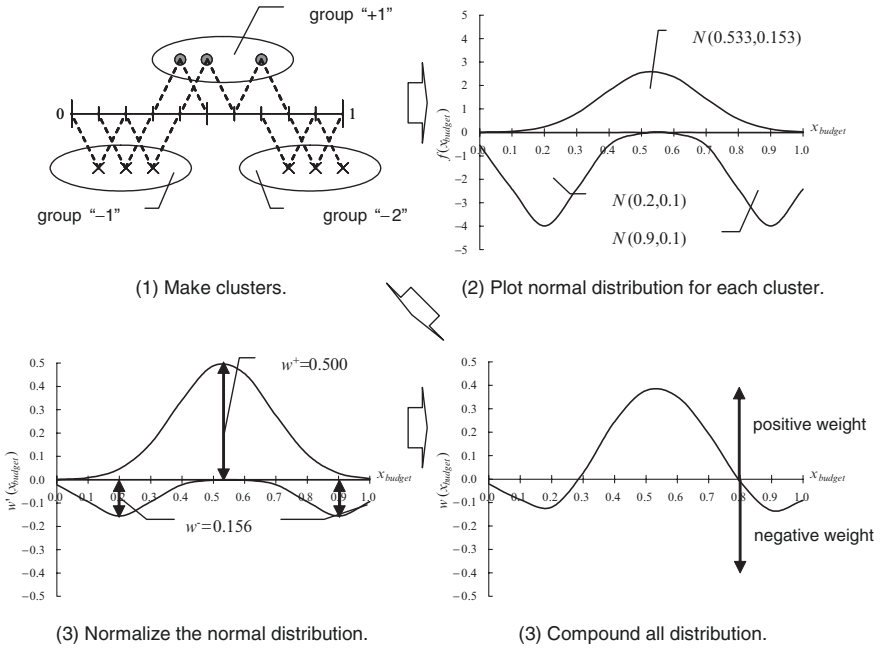


Fig. 23.5 Generation weight distribution of user's preferences

23.3.5 Calculation of Ranking Score Based on User's Weight Distribution

We calculate a ranking score for each restaurant according to the weight distribution described in Sect. 23.3.2.

Now, we focus on a restaurant k. $w_i(x)\{x : 0 \leq x \leq 1\}$ is given as weight distribution for a parameter i. For the restaurant k, $x_{k,i}$ is given as parameter value of the parameter i. Then, the ranking score of the restaurant k is given as the following equation:

$$s_k = \sum_i w_i(x_{k,i}) \tag{23.1}$$

Finally, restaurants are ranked according to the ranking score.

23.4 Verifying SVM's Generalization Performance

It is important that a recommendation system finds profitable contents for users from the enormous number of unknown information sources. Therefore, generalization performance for untrained data in particular is required for classifiers applied to the recommendation method.

In this study, we adopt an SVM that is superior in terms of generalization performance, and in this section we verify the SVM's generalization performance based on comparison experiments with other classifiers' performance.

Besides an SVM [7], a neural network (NN), a *k*-nearest neighbor (*k*NN), a decision tree (DecTree), and a Bayesian filtering (BF) are enumerated as principal classifiers. These five principal classifiers are used for performance comparison.

23.4.1 Procedure to Experiment

The real data sets are based on users' evaluations of restaurant data in advance. The number of training data is 100, which includes user contexts.

We use the model's accuracy as an index of evaluation. The model's accuracy, represented as Eq. (23.2), is an index showing how correctly the decision model can distinguish between the test data sets.

$$\textit{Model's Accuracy} = \frac{\textit{the number of data distinguished correctly}}{\textit{total number of test data}} \times 100(\%) \quad (23.2)$$

In this experiment, we verify a) the model's accuracy for trained data, and b) the model's accuracy for untrained data (namely, generalization performance). We use WEKA [8], which is a tool for data mining, to determine the model's accuracy by each classifier.

23.4.2 Results and Considerations

Figure 23.6 a) illustrates the model's accuracy for trained data, and b) does so for the model's accuracy for untrained data using real data sets. These graphs depict the model's accuracy for each classifier, where each decision model is based on the training data with consideration of user contexts but no consideration of user contexts. Regarding trained data (shown in Fig. 23.6(a)), every classifier has higher model accuracy when considering user contexts than when not doing so. The SVM, the NN, and the *k*NN in particular had high model accuracy by considering user contexts. For untrained data (shown in Fig. 23.6(b)), the SVM had the higher model accuracy than any other classifiers, while the NN and the *k*NN experienced extreme loss of model's accuracy against the model's accuracy for trained data.

As we stated at the beginning of this section, generalization performance for untrained data in particular is required for classifiers applied to the recommendation method. Accordingly, we think that it is appropriate to adopt an SVM for the recommendation method because the SVM has high generalization performance.

Although it is assumed that an NN and a decision tree work more effectively when there is a large amount of training data to learn, we want the classifier to

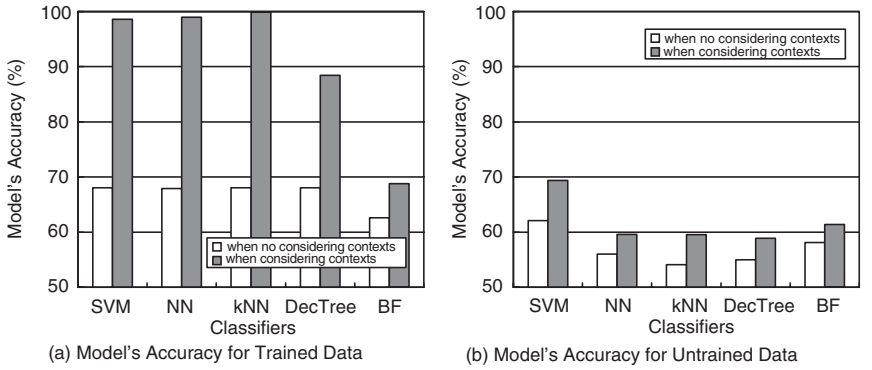


Fig. 23.6 Model's accuracy for trained/untrained data using real data sets

produce high generalization performance even with little training data from the standpoint of practical usage. Thus, an SVM that gives high generalization performance with a small amount of training data would be desirable in practice.

23.5 Experiments Applying C-IF and C-CF to a Restaurant Recommendation System

23.5.1 Experiment 1

It is more effective to use C-IF and C-CF in different ways, depending on the user's situation. However, it is not clear what advantages and disadvantages of the C-IF and C-CF apply to various users' situations. In Experiment 1, we clarified the differences of advantages and disadvantages between C-IF and C-CF in various users' situations.

23.5.1.1 Procedure

In this experiment, we use three restaurant databases: Area A—Shinsaibashi; Area B—Tennoji; and Area C:—Umeda, as shown in Table 23.3. These databases have been extracted from the Japanese gourmet site Yahoo! Gourmet [9]. Each area is an entertainment district located in Osaka, Japan. Areas A and B are both large-scale areas, while C is significantly smaller.

The system shows restaurant data to testees from these restaurant databases in random order. Each testee evaluates the shown restaurant data by assigning a *positive* or *negative* classification. Nine testees participated in the experiment, and were divided into three groups of three (Area A group, Area B group and Area C group).

Table 23.3 Restaurant database areas used for recommendation

Area	Number of Restaurants	Scale of Area
A:'Shinsaibashi'	650	Large
B:'Tennoji'	121	Small
C:'Umeda'	445	Large

Each testee from each group created his/her training data in a different area—i.e., one from each group in Area A, one for Area B, and one for Area C. The number of collected training data is 100, and the restaurant data are shown in ten contexts. Each testee evaluates ten restaurant data in one context. In this way, their training data are produced from their evaluations of 100 restaurant data. Their CP-models are then made using these training data.

The system recommends restaurant data using both C-IF and C-CF to the testees. Restaurant data are recommended in every area, with 50 recommendations given per area for a total of 150 per group. Each testee evaluates the recommended restaurant data by allocating a *positive* or *negative* grade. Thus, we obtain their satisfaction rates for the recommendation using the C-IF and the C-CF. The users' satisfaction rate is represented as Eq. (23.3).

$$\textit{User's Satisfaction Rate} = \frac{\textit{the number of positive data}}{\textit{the number of recommendations}} \times 100(\%) \quad (23.3)$$

23.5.1.2 Results and Considerations

We compare the testees' satisfaction rates in each recommendation area. In this paper, *trained areas* denote areas used to collect testee's training data, and *untrained areas* denote areas other than the trained areas. Taking the testees of the Area A group as an example, Area A is regarded as a trained area while Areas B and C are regarded as untrained areas.

Figure 23.7(a) illustrates the testees' satisfaction rates in each group for trained areas, and Fig. 23.7(b) shows that for untrained areas.

For trained areas (shown in Fig. 23.7(a)), the testees' satisfaction rates for the recommendation using C-IF are higher than using C-CF. On the other hand, for untrained areas (shown in Fig. 23.7(b)), the testees' satisfaction rates for the recommendation using C-CF are higher than using C-IF.

To summarize these results, the advantages of C-IF and C-CF are as follows:

- For trained areas, recommendation using C-IF is more effective than using C-CF.
- For untrained areas, recommendation using C-CF is more effective than using C-IF.

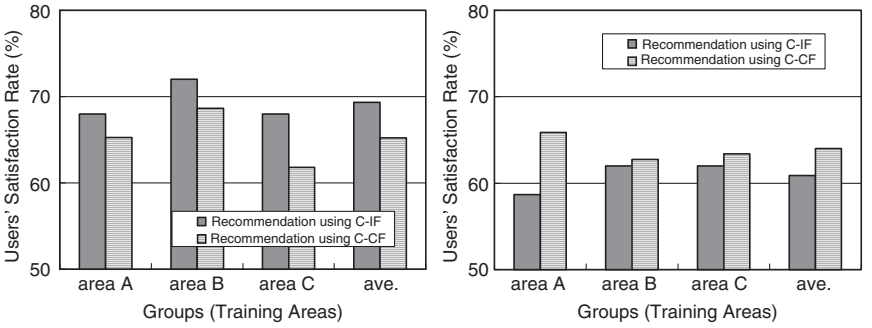


Fig. 23.7 Users' satisfaction rates for recommendation for trained/untrained areas

Thus, it is more effective to use C-IF and C-CF in different ways depending on the user's situation. In future work, we would like to explore how to determine the number of training data required for recommendation using C-IF, and how to choose similar users for the recommendation using C-CF to improve the effectiveness of both methods.

23.5.2 Experiment 2

The parameters of feature vectors used for our recommendation system should be optimized to gain higher generalization performance of the CP-model. In this section, we call the parameters of feature vectors *feature parameters*. In Experiment 2, we investigate how much each parameter influences the generalization performance of the CP-model.

23.5.2.1 Procedure

We focus on the feature parameters shown in Table 23.1 and Table 23.2. The restaurant feature parameters, shown in Table 23.1, are classified into four groups: Genre, Budget, Characteristic, and Atmosphere. The context feature parameters, shown in Table 23.2, are classified into six groups: Month, Day of Week, Time, Weather Info., User Info. and Companions Info. Finally, we analyze the influence on the generalization performance of the C-IF model when each feature parameter group is removed.

By removing a feature parameter group, decreases, the removed feature parameter group is regarded as the valuable feature parameter group if the generalization performance. If the generalization performance increases, the removed feature parameter group is regarded as the negative feature parameter group.

23.5.2.2 Results and Consideration

Figure 23.8 illustrates the accuracy of the CP-model for untrained data when each feature parameter group is removed. The vertical axis in Fig. 23.8 denotes the removed feature parameter group, and the broken line in Fig. 23.8 denotes the generalization performance of the CP-model without removing any feature parameter group (71.7%).

We see from Fig. 23.8 that the valuable feature parameter groups are Genre, Companions Info., Time and User Info., Atmosphere, and Budget, in that order. It seems that the Genre of restaurants is the most valuable feature parameter in this experiment. We also issued the following request to the testees: "List the parameter groups in order of importance to you," and then confirmed that the results of this request were consistent with the results of Experiment 2.

In other circumstances, we see from Fig. 23.8 that the negative feature parameter groups are Characteristic, Month, Weather Info., and Day of Week, in that order. We also obtained the model accuracy of 73.4 percent, 73.8 percent, 74.1 percent, and 74.7 percent, in order when these four feature parameter groups are removed cumulatively.

Accordingly, the model accuracy can be improved by removing such negative feature parameters. However, it is not always the case that these identified negative feature parameters are useless to a practical recommendation system, because these identified feature parameters depend on the experimental environment. Namely, it is necessary to optimize the feature parameters for adjusting the system environment when the system is developed. However, the results suggest that our method of optimizing feature parameters is effective.

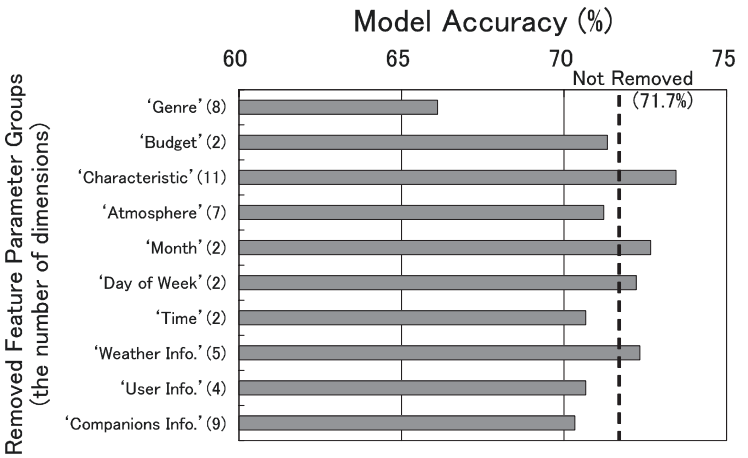


Fig. 23.8 Model's accuracy for untrained data when each feature parameter group is removed

23.5.3 Experiment 3

In this section, we evaluate the effectiveness of a proposal method by experiment. Comparative methods for ranking are shown as follows:

I. A method based on weight distribution for parameter values (proposal method referred to in Sect. 23.3).
II. A method based on classification for *multi-class* case.
III. A method based on euclidean distance from a decision plane of an SVM.

23.5.3.1 Procedure

Data sets for evaluation are shown in Table 23.4. The feature parameter of restaurant data is represented by 28 dimensions. The context parameter is represented by 24 dimensions. Each data has a five-grade satisfaction rate that testees have evaluated in advance. Data having an evaluation value over three are regarded as *positive data*, and the others are regarded as *negative data*. The evaluation value was given for each context pattern imagined by the testee. The number of context patterns is five, and the amount of data per a context is shown in Table 23.4.

A procedure of this experiment is shown as follows:

1. Adjust weight distribution based on the training data set following the procedures in Sect. 23.3.
2. Calculate ranking scores of each data in the recommendation data set following the procedure in Sect. 23.3.
3. Make rankings for recommendation data sets according to the score.

In this experiment, weight distribution is calculated for respective context patterns. Because the weight for parameters is changed according to their contexts. Recommendation data sets are also ranked for respective context pattern. Therefore, from the first to the 30th restaurant, data are listed at each context pattern.

We evaluate the experimental result with a recall-precision curve and *R*-precision. Recall corresponds to the percentage of the amount of recommended data in the all of positive data. Precision corresponds to the percentage of the amount of positive data in the recommended data set. A recall-precision curve is a graph that shows the relation between recall and precision. The perfomance of a method whose curve is plotted upward is better. *R*-precision is precision when the first restaurant to *R*th restaurant are recommended.

Table 23.4 Data sets for evaluation

Data Sets	Amount of Data	Amount of Data per a Context Pattern
training data set	100	20
recommendation data set	150	30

This section compares the performance for the ranking of methods I, II, and III. The number of testees who participated in this experiment was nine.

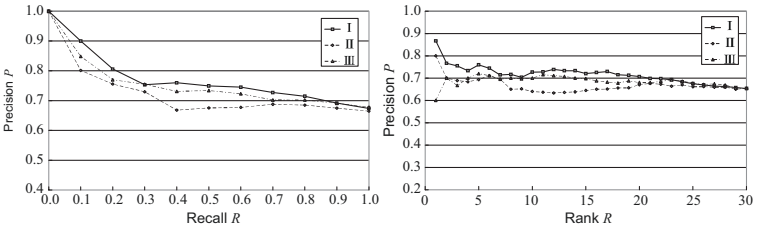
23.5.3.2 Results and Considerations

Recall-precision curves and the *R*-precision of each ranking method are shown in Figs. 23.9 and 23.10, respectively. The results are shown for the following cases:

(a) Data evaluated over three are regarded as positive data.
(b) Data evaluated over four are regarded as positive data.

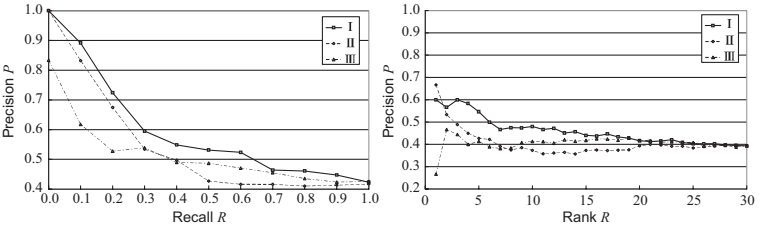
These results show the average of results obtained for five context patterns. In all cases, it is shown that a proposal method is comparatively effective.

Regarding *R*-precision, Fig. 23.10 shows that a user may be satisfied with 70 percent of 10 recommended restaurants if using positive data evaluated over three, and satisfied with 50 percent if using positive data evaluated over four. The result shows that the proposed method works well enough. In the case of using euclidean distance from a decision plane of an SVM, the system cannot distinguish between scores 3, 4 and 5, and regard them as positive data evenly. Thus, it is not always true that data from decision plane are corresponding to data with high satisfaction rate. Though the method using classification for multi-class is better than using the euclidean distance from a decision plane, it is not better than the proposed method.



(a) using positive data evaluated over three. (b) using positive data evaluated over four.

Fig. 23.9 Recall-Precision Curve



(a) using positive data evaluated over three. (b) using positive data evaluated over four.

Fig. 23.10 R Precision

On the other hand, our proposed method has the following advantages:

- It can rank recommended data while considering user's context and preferences appropriately.
- User should rank test data to only two classes, not to multi-classes.

23.6 Conclusions

We investigated a method of developing a context-aware recommendation system on the basis of experiments applying C-IF and C-CF to a restaurant recommendation system. First, we verified that it is appropriate to adopt an SVM to a recommendation method because the SVM has high generalization performance. Second, we gained knowledge of the advantages of both C-IF and C-CF as follows: for trained areas, a recommendation using C-IF is more effective than one using C-CF; and for untrained areas, a recommendation using C-CF is more effective than using C-IF. We also showed that it is possible to optimize parameters of feature vectors by removing negative feature parameters to improve the generalization performance.

We proposed a context-aware ranking method. This method makes ranking for recommended restaurant data while considering users' contexts. We compared the performance of our method and conventional ranking methods by conducting experiments. The experimental results showed that it is effective to use our method for ranking.

In future work, we would like to investigate a recommendation method that considers historical data of users' behavior as well as the relativity of the surrounding environment. Furthermore, we would like to develop a context-aware recommendation system for various types of content (e.g., music and movies, not just restaurants).

Acknowledgements This research was partly supported by MEXT (Grant-in-Aid for Scientific Research on Priority Areas #19024058).

References

1. Chen G and Kotz D (2000) A Survey of Context-Aware Mobile Computing Research, Paper TR2000-381, Department of Computer Science, Dartmouth College
2. Asthana A, Cravatts M and Krzyzanowski P (1994) An Indoor Wireless System for Personalized Shopping Assistance, Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, IEEE Computer Society Press, pp. 69–74
3. Long S, Kooper R, Abowd GD and Atkeson CG (1996) Rapid Prototyping of Mobile Context-Aware Applications: the Cyberguide Case Study, Proceedings of the Second Annual International Conference on Mobile Computing and Networking, White Plains, NY, ACM Press, pp. 97–107

4. Davies N, Cheverst K, Mitchell K and Friday A (1999) Caches in the Air: Disseminating Tourist Information in the GUIDE System, Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, Louisiana, IEEE Computer Society Press
5. Oku K, Nakajima S, Miyazaki J and Uemura S (2006) Context-Aware SVM for Context-Dependent Information Recommendation, Proc. of International Workshop on Future Mobile and Ubiquitous Information Technologies (FMUIT2006), pp. 119–122
6. Oku K, Nakajima S, Miyazaki J and Uemura S (2007) Investigation for Designing of Context-Aware Recommendation System Using SVM, Proc. of The International MultiConference of Engineers and Computer Scientists 2007 (IMECS 2007), March
7. Cortes C and Vapnik V (1995) Support-Vector Networks, Machine Learning, Vol. 20, No. 3, pp. 273–297
8. WEKA: http://www.cs.waikato.ac.nz/ml/weka/ (2006-12-29 confirmed)
9. Yahoo! Gourmet, http://gourmet.yahoo.co.jp/gourmet/ (2006-12-29 confirmed)
10. Chang C-C and Lin C-J (2006) LIBSVM: a Library for Support Vector Machines

Chapter 24
Multiplierless Synthesis of Multiple Constant Multiplications Using Common Subexpression Sharing With Genetic Algorithm

Yuen-Hong Alvin Ho, Chi-Un Lei, and Ngai Wong

In the context of multiple constant multiplications (MCM) design, we propose a novel common subexpression elimination (CSE) algorithm that models synthesis of coefficients into an estimated cost function. Although the proposed algorithm generally does not guarantee an optimum solution, it is capable of finding the minimum/minima of the function in practically sized problems. In our design examples that have known optimal solutions, syntheses of coefficients using the proposed method match the optimal results in a defined search space. We also discover the relationship and propose an improvement search space for optimization that combines all minimal signed digit (MSD) representations, as well as the shifted sum (difference) of coefficients to explore the hidden relationship. In some cases, the proposed feasible solution space further reduces the number of adders/subtractors in the synthesis of MCM from all MSD representations.

24.1 Introduction

Multiple constant multiplications (MCM) are typical operations in digital signal processing (DSP) as well as in the design of finite impulse response (FIR) filters, as shown in Fig. 24.1. Multiplierless MCM is the focus of a lot of research on high-speed and low-power devices. In multiplierless MCM, multipliers are replaced by simpler components such as adders and hard-wired shifts, as shown in Fig. 24.2. (the meaning of *adders* in our chapter includes subtractors as well because their hardware cost is similar).

A commonly used method for minimizing the number of adders in a MCM block is common subexpression elimination (CSE) [1–7]. A conventional CSE algorithm

Yuen-Hong Alvin Ho, Chi-Un Lei, and Ngai Wong
Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong
e-mail: {alvin,culei,nwong}@eee.hku.hk

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

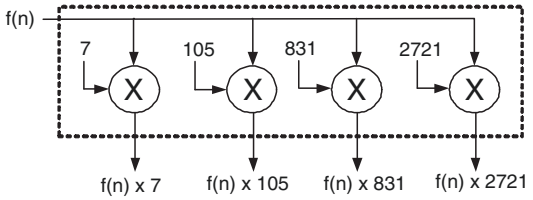


Fig. 24.1 A multiplier-based MCM example

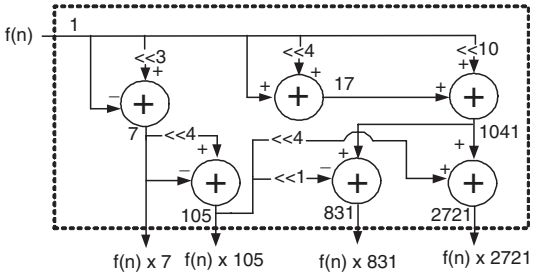


Fig. 24.2 A multiplierless-based MCM example

uses different signed-digit representations and looks for common subexpressions to decompose coefficients. A proposed algorithm by Hartley finds common subexpressions from the CSD representation. Adder reduction of subexpressions is estimated using an exhaustive search. Flores et al. proposed an optimum solution technique [7] that uses a SAT-based 0-1 integer linear programming (ILP) solver. Their results show that the larger search space of all minimal signed digit (MSD) representations is improved over the CSD representation in most cases.

Feasible solution space other than the search space of signed-digit representations may also be used to synthesize MCM. For example, Wang and Roy propose an augmented differential coefficient approach that expands the design space by employing both differences and sums of filter coefficients. Yao et al. also proposes an algorithm that searches for pairs of available subexpressions to synthesize coefficients. The algorithm extracts all subexpressions of coefficients from CSD patterns and treats coefficients themselves as CSD subexpressions of each other.

In this chapter, we propose a CSE algorithm, which determines the syntheses of coefficients by using an estimated cost function. The function accepts syntheses of coefficients as inputs and returns an integer that estimates the adder cost. A genetic algorithm [8, 9] is used to minimize the cost function when the search space is large. It is found that the solutions of the proposed CSE algorithm match the optimum solutions in Flores et al. We also propose a feasible solution space that uses the shifted sum (difference) of coefficients. The expansion of the feasible solution space is applied in our algorithm and has shown improvements over all MSD representations.

Multiplierless MCM can be synthesized without subexpression sharing. An example is the simple CSD method in Hartley and Park et al., in which the adder cost to synthesize a coefficient is (*#nzbit* of the coefficient -1), where *#nzbit* denotes the number of non zero bits in the CSD or MSD representations. Because the simple CSD method synthesizes a multiplierless MCM without sharing any subexpression, the result of the simple CSD method is used as the upper bound of adder cost in the synthesis of multiplierless MCM. On the other hand, the minimum adders required to synthesize a multiplierless MCM from a set of coefficients are not well determined. Because at least one adder is required to synthesize a coefficient, a well-known lower bound of adder cost is the number of coefficients in the MCM. This chapter is organized in the following way. In Sect. 24.2, we introduce the notations in the chapter. In Sect. 24.3, different feasible solution spaces are shown in a lookup table of coefficient decompositions. The way to extract decompositions from the CSD or MSD representations as well as the proposed expansion of feasible solutions is described. In Sect. 24.4, an estimated cost function that uses a lookup table of decompositions is proposed. In Sect. 24.5, we introduce a CSE algorithm that synthesizes a multiplierless MCM using the estimated cost function. In Sect. 24.6, an example is given to numerically illustrate the proposed algorithm. In Sect. 24.7, our results are compared to the representative results in Yao et al. and Flores et al.

24.2 Notation

The following notations are used in this chapter.

- C denotes coefficient
- *#nzbit* denotes the number of non-zero bits in the CSD or MSD representations
- S denotes subexpression
- *Cset* is defined as the set of coefficients that is waiting to be decomposed
- N is the number of coefficients in *Cset*
- n is an integer where $1 \leq n \leq N$
- C_n is the nth coefficient in *Cset*
- LT_n is the decomposition table of C_n
- L_n is the number of decompositions in LT_n
- x_n is an integer variable that is in the range of $1 \leq x_n \leq L_n$. It is also used as an index corresponding to a decomposition in LT_n
- *Dset* is the set of coefficients that have determined synthesis
- *RT* is the result table that stores the synthesis of multiplierless MCM
- *LDB* is the pre-defined logic depth bound that limits the maximum logic depth of the MCM. A logic-depth example is shown in Fig. 24.3.

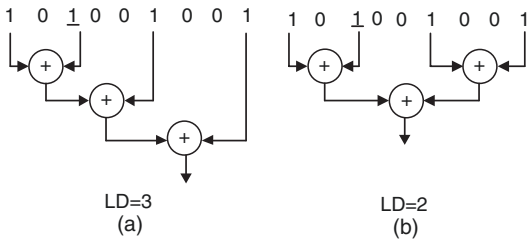


Fig. 24.3 Example of synthesing a coefficient 201: (a) logic depth (LD) = 3; (b) logic depth = 2

24.3 Feasible Solution Space

24.3.1 Introduction

The adder cost minimization problem in the synthesis of a set of coefficients of size N is NP-complete [10]. Therefore, the search space should be bounded to reduce the searching time. In the proposed CSE method, the negative coefficients are converted at the output of a MCM block. Only the absolute values of negative coefficients are considered. Similarly, even coefficients are continuously divided by 2 until they become odd. Original coefficients are implemented by left-shifting the corresponding odd coefficients. Also, duplicated coefficients are synthesized once only because multiple outputs can be wired from a single coefficient. The modified coefficients have reduced the search space. As a result, the coefficients in the synthesis of MCM are unique, positive, and odd.

Equation (24.1) defines a decomposition of a coefficient C_n

$$C_n = \pm S_i \times 2^p \pm S_j \times 2^q \tag{24.1}$$

where $S_i(S_j)$ is a subexpression of C_n that must be positive and odd; p and q are left shifts of S_i and S_j for p and q bits, respectively; and $S_i(S_j) \neq C_n$.

There are infinite decompositions that satisfy Eq. (24.1) in a single coefficient. Some constraints are required to limit the search space. In our CSE algorithm, each coefficient C_n generates a lookup table LT_n that stores the set of feasible decompositions of C_n. Different decompositions and values of S_i, S_j, p, and q of coefficient 831 are shown as an illustrative example.

24.3.2 Table Generation

Step 1: For each coefficient C_n, a lookup table LT_n that stores decompositions of C_n is generated based on all MSD representations of C_n.

During the decomposition extraction from a MSD representation, the decomposition that has the same pair of subexpressions is inserted into LT_n only once. This

Table 24.1 Complete table of $C_2 = 831$

x_2	Decomposition	Search Space	
1	$1 \times 2^{10} - 193$	CSD / MSD	$\boxed{1}\,0\underline{1}01000000\underline{1}$
2	$-1 \times 2^8 + 1087$	CSD / MSD	$10\,\boxed{\underline{1}}\,01000000\underline{1}$
3	$1 \times 2^6 + 767$	CSD / MSD	$10\underline{1}0\,\boxed{1}\,000000\underline{1}$
4	$-1 + 13 \times 2^6$	CSD / MSD	$10\underline{1}0100000\,\boxed{\underline{1}}$
5	$3 \times 2^8 + 63$	CSD / MSD	$\boxed{1}\,0\,\boxed{\underline{1}}\,01000000\underline{1}$
6	$17 \times 2^6 - 257$	CSD / MSD	$\boxed{1}\,0\underline{1}0\,\boxed{1}\,000000\underline{1}$
7	$1023 - 3 \times 2^6$	CSD / MSD	$\boxed{1}\,0\underline{1}0100000\,\boxed{\underline{1}}$
8	$-1 \times 2^7 + 959$	MSD	$100\,\boxed{\underline{1}}\,1000000\underline{1}$
9	$-1 \times 2^6 + 895$	MSD	$100\underline{1}\,\boxed{\underline{1}}\,000000\underline{1}$
10	$7 \times 2^7 - 65$	MSD	$\boxed{1}\,00\,\boxed{\underline{1}}\,1000000\underline{1}$
11	$15 \times 2^6 - 129$	MSD	$\boxed{1}\,00\underline{1}\,\boxed{\underline{1}}\,000000\underline{1}$
12	$1 \times 2^9 + 319$	MSD	$0\,\boxed{1}\,101000000\underline{1}$
13	$1 \times 2^8 + 575$	MSD	$01\,\boxed{1}\,01000000\underline{1}$
14	$9 \times 2^6 + 255$	MSD	$0\,\boxed{1}\,10\,\boxed{1}\,000000\underline{1}$
15	$511 + 5 \times 2^6$	MSD	$0\,\boxed{1}\,10100000\,\boxed{\underline{1}}$
16	$383 + 7 \times 2^6$	Proposed	
17	$1055 - 7 \times 2^5$	Proposed	
18	$1279 - 7 \times 2^6$	Proposed	
19	$-961 + 7 \times 2^8$	Proposed	
20	$1041 - 105 \times 2^1$	Proposed	
21	$-9 + 105 \times 2^3$	Proposed	
22	$111 \times 2^5 - 2721$	Proposed	Invalid

is because the syntheses of different decompositions of a coefficient are equivalent to each other if they have the same pair of subexpressions.

An example of decomposition extractions is shown in Table 24.1. $C_2 = 831$ has three different MSD representations, and they are $10\underline{1}01000000\underline{1}$, $100\underline{11}000000\underline{1}$, and $01101000000\underline{1}$, where $\underline{1}$ represents a -1 digit.

The first MSD of $C_2 = 831$ is, in fact, also the CSD of C_2. Decompositions from $x_2 = 1$ to 7 are generated from the CSD representation. They are extracted from the combinational patterns of the coefficient's signed-digit representation. For example, the decomposition at $x_2 = 5$ $\left(\boxed{1}\,0\,\boxed{\underline{1}}\,01000000\underline{1}\right)$ is extracted as $10\underline{1} \ll 8 + 1000000\underline{1}$ $(831 = 3 \times 2^8 + 63)$. Similarly, the decomposition at $x_2 = 6$ $\left(\boxed{1}\,0\underline{1}0\,\boxed{1}\,000000\underline{1}\right)$ is extracted as $10001 \ll 6 + \underline{1}00000000\underline{1}$ $(831 = 17 \times 2^6 - 257)$.

The rest of MSD representations are also used to expand the number of decompositions. The second and the third MSD representations are $100\underline{11}000000\underline{1}$ and $01101000000\underline{1}$. The additional decompositions are indicated from $x_2 = 8$ to 11, and from 12 to 15, respectively.

CSD and MSD representations have the same minimum number of non zero bits. The decompositions of all MSD representations expand the solution space over the CSD representation. The table size of all MSD representations is about twice the table size of the CSD representation in Table 24.1. The increased search space provides more possible decompositions that may lead to less adder cost in the synthesis of multiplierless MCM.

Because small coefficients often occur in the MCM of FIR filters, the decomposition process in this step is often repeated. To reduce the unnecessary computation, decompositions are pregenerated and stored into the hard disk for coefficients up to 13 bits (i.e., 8192). Decompositions of coefficients that are not pregenerated are generated during execution and saved for future used. The size of our saved data is about 10MB in Matlab 7.

Step 2: Insert decompositions that satisfy Equation 24.2 into LT_n for all n.

Eq. (24.2) is defined as:

$$C_n = \pm S_k \times 2^p \pm S_l \times 2^q \tag{24.2}$$

where Eq. (24.2) is a more constrained version of Eq. (24.1). In Eq. (24.2), $S_k(S_l)$ is a subexpression of C_n that must be positive and odd; p and q are left shifts of S_k and S_l for p and q bits, respectively; $S_k(S_l) \neq C_n$; S_k must be a coefficient in $Cset \cup Dset \cup \{1\}$; and S_l must either be a coefficient in $Cset \cup Dset \cup \{1\}$ or *#nzbit* of $C_n - 1$ *#nzbit* of $S_l \geqslant 1$.

Because C_n, S_k, and S_l are restricted to be odd, one of the $S_k \times 2^p$ or $S_l \times 2^q$ must be odd and the other must be even. Equation (24.2) is then manipulated as follows:

$$C_n = \pm S_k \pm S_l \times 2^q \text{ for } p = 0 \text{ and } q \geqslant 1 \tag{24.3}$$

or

$$C_n = \pm S_k \times 2^p \pm S_l \text{ for } p \geqslant 1 \text{ and } q = 0 \tag{24.4}$$

Also, because both C_n and S_k are in $Cset \cup Dset \cup \{1\}$, C_n, S_k, S_l are all positive, and Eq. (24.3) is manipulated as follows:

$$S_l = (C_n + S_k) \times 2^{-q} \quad \text{or} \tag{24.5}$$

$$S_l = (C_n - S_k) \times 2^{-q} \quad \text{or} \tag{24.6}$$

$$S_l = (-C_n + S_k) \times 2^{-q} \tag{24.7}$$

Equation (24.4) is manipulated as follows:

$$S_l = +C_n + S_k \times 2^p \text{ or} \tag{24.8}$$

$$S_l = +C_n - S_k \times 2^p \text{ or} \tag{24.9}$$

$$S_l = -C_n + S_k \times 2^p \tag{24.10}$$

Decompositions that satisfy Eq. (24.2) are determined based on the shifted sum (difference) of coefficients (C_n and S_k). Exhaustive search are used to search for the

feasible decompositions in Eq. (24.5) to (24.10). Decompositions from $x_2 = 16$ to 22 in Table 24.1 shows the result of 831 when $Cset = \{2721, 831, 105, 7\}$.

Let MCL be the maximum word length of coefficients in the CSD representation. For an estimation of complexity, each coefficient has an upper bound of comparisons of $O(N \times MCL)$. Therefore, the cumulated complexity for N coefficients in $Cset$ is $O(MCL \times N^2)$.

The proposed expansion increases the feasible solution space and, as a result, better solutions may be found. If S_l is in the set $Cset \cup Dset \cup \{1\}$, C_n may be synthesized with one additional adder. Otherwise, the upper bound of C_n synthesis equals *#nzbit* of S_l.

Similar to the decompositions that are generated from the CSD and the MSD representations, decompositions of the shifted sum (difference) of coefficients have the same upper bound adder cost. Because S_l is constrained to have at least one less *#nzbit* than C_n in Eq. (24.2), the synthesis cost of C_n in any decomposition does not exceed the upper bound (the simple CSD method). Decompositions of the shifted sum (difference) of coefficients introduce decompositions that cannot be found from the CSD and the MSD representations. Section 24.6 shows a case of MCM that leads to a lower adder cost from the proposed decompositions numerically.

Step 3: Remove invalid decompositions from LT_n based on RT and LDB.

A decomposition in LT_n is invalid if it immediately causes the critical path of MCM to exceed LDB when it is synthesized. The critical path of MCM is the largest logic depth (LD) of the synthesized coefficients in RT. The synthesized coefficient in RT has a logic depth equal to $\max(\{LD \text{ of } S\}) + 1$, where LD denotes logic depth and S corresponds to the pair of subexpressions of the decomposition in RT. Subexpressions that have not been synthesized in RT are assumed to have the lowest possible logic depth calculated as $\lceil \log_2(\#nzbit \text{ of } S) \rceil$. A decomposition is also invalid if it causes a feedback in the synthesis of MCM after it is inserted into RT.

The decomposition at $x_2 = 22$ in Table 24.1 is $111 \times 2^5 - 2721$. When $LDB = 3$, the decomposition immediately causes the critical path of MCM to exceed LDB. It is found to be invalid as shown below:

$$\begin{aligned} LD \text{ of } 111 &= \lceil \log_2(\#\text{nzbit of } 111) \rceil &&= 2 \\ LD \text{ of } 2721 &= \lceil \log_2(\#\text{nzbit of } 2721) \rceil &&= 3 \\ &\max(\{2,3\}) + 1 &&= 4 > LDB = 3 \end{aligned}$$

24.4 Proposed Estimated Cost Function

We propose a way to estimate the adder cost of MCM using all lookup tables from LT_1 to LT_N. The value of x_n corresponds to a decomposition in LT_n as an example in Table 24.1. We define a MCM estimated cost function as

$$f(x_1,...,x_N) = \#\text{ValidDecomposition} + \sum_{i}^{All}(\#\text{nzbit in } NS_i - 1) \tag{24.11}$$

where f accepts N input variables from x_1 to x_N and returns an integer number. Before the evaluation, NS is an empty set, and #ValidDecomposition is equal to the number of decompositions in RT.

During the evaluation, decompositions are temporarily inserted into RT in a sequential order, from x_1 to x_N. If the decomposition at x_n is invalid, C_n is inserted into NS. Otherwise, the decomposition at x_n is temporarily inserted into RT; #ValidDecomposition is increased by 1; and for the subexpressions at x_n that are not in $Cset \cup Dset \cup 1$, they are inserted into NS.

At the end of all insertions, the value of f is calculated as shown in Eq. (24.11). Decompositions that are temporarily inserted are removed after the evaluation. The adder cost of NS is calculated based on the simple CSD method. As a result, the estimated cost function is an upper bound of the adder cost when the MCM is synthesized according to x_1 to x_N.

In a typical MCM design, f may have 10 input variables and each variable may range from 1 to 10. The search space of f is 10^{10}. The minimization complexity of f that uses a brute force approach is $O(M^N)$ where M is the table size and N is the number of variables.

It is assumed that the decompositions in a table are randomly ordered. However, useful information can still be obtained from the relationship between different input variables. A genetic algorithm is used in this problem because it can give a local optimal solution efficiently even though the search space is complicated. In genetic algorithms, minimization of f is started from a population of randomly chosen input variables. A genetic algorithm searches for a pattern of inputs that is closed to the minimum. The pseudo code of a genetic algorithm is shown below:

1. Choose an initial population.
2. Evaluate the fitness of each individual in the population.
3. Select best-ranking individuals to reproduce.
4. Breed new generations through crossover and mutation (genetic operations) and give birth to offspring.
5. Evaluate the individual fitness of each offspring.
6. Replace worst-ranked part of population with offspring.
7. Repeat Step 3 to Step 6 until <terminating condition>.

A new population can be obtained from randomly exchanging (crossover) variables between two input patterns (chromosomes). An example of a crossover operation is shown in Fig. 24.4. $f(7,2,5,4,5)$ is assumed to be at the minimum of f; $f(1,2,3,4,5)$ and $f(7,6,5,4,3)$ are one of the pairs of chromosomes in a population. The circled input variables are randomly chosen and exchanged. The number of input vairables that matches the input pattern of $f(7,2,5,4,5)$ are shown in Fig. 24.4 to indicate the fitness.

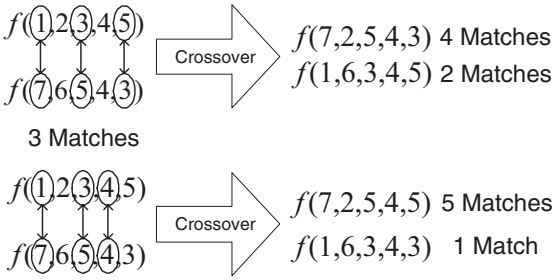


Fig. 24.4 A crossover operation when $f(7,2,5,4,5)$ is the minimum of f

In our CSE algorithm, the minimum of f is unknown. The fitness of a chromosome is determined by the estimated cost f. A good cost function is important because the fitness formulation is related to speed and accuracy. After each crossover operation, the newly created chromosomes are inserted into the population and sorted according to their adder cost. The population has a predefined size. Therefore, chromosomes with a large adder cost are dropped before the next crossover operation. Crossover operations are performed iteratively to minimize f. Although a genetic algorithm generally does not guarantee an optimum solution, examples have shown that the final solution is well-adapted into our CSE algorithm.

24.5 Proposed Algorithm

We are proposing an algorithm to maximize the subexpression sharing using Eq. (24.11). A simplified flowchart of the algorithm is shown in Fig. 24.5. The description of the algorithm is as follows:

1. Reduces all multiplication values of MCM to a set of coefficients that are unique, positive, and odd, as illustrated in Sect. 24.3. Put them into *Cset*.
2. Order the coefficients in *Cset* by their *#nzbit* in descending order. If there are multiple coefficients that have the same *#nzbit*, order them by their integer values in descending order. Create or update all LT_n according to the procedures in Sect. 24.4.
3. Search for valid decompositions that may synthesize a coefficient with one adder in a sequential order, from LT_1 to LT_N. If a coefficient may be synthesized with one adder, synthesize the coefficient and update RT. Move the coefficients from *Cset* to *Dset*. If one or more decompositions may synthesize a coefficient with one adder, choose a decomposition that would result in the lowest logic depth. If a coefficient is synthesized, go to Step 7). Otherwise go to Step 4).
4. Minimize Eq. (24.11) and save the sets of decompositions that have the minimum cost.

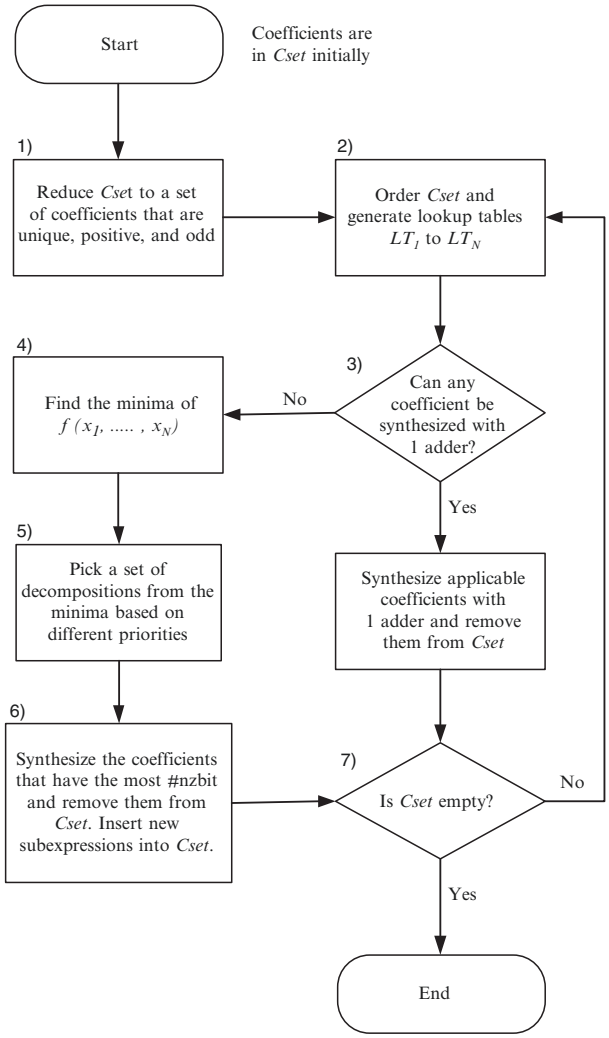


Fig. 24.5 A simplified flowchart of our proposed CSE algorithm

5. If multiple minima are found in Eq. (24.11), pick the set of decompositions according to the following priority:
   i) the lowest average logic depth of the temporarily synthesized coefficients in *Cset*
   ii) the lowest average logic depth of *NS*
   iii) the smallest average value of *NS*
6. Synthesize the coefficient(s) that have the most *#nzbit* into *RT*. Move the coefficient(s) from *Cset* to *Dset*. If there are new subexpressions in *RT* that are not in the set of $Cset \cup Dset \cup \{1\}$, copy them into *Cset*.

7. If *Cset* is not empty, go back to Step 2. Otherwise, the *RT* would contain the full synthesis of MCM.

Analysis of the algorithm and an illustrative example is shown in Sect. 24.6.

24.6 Example

As an example to illustrate our algorithm, we initially set $Cset = \{105, 831, 2721, 896\}$. $LDB = 3$. $N = 4$. $Dset = \{\}$. $RT = \{\}$.

In (1), coefficients are reduced to unique, odd, and positive in *Cset*. The resultant *Cset* is $\{105, 831, 2721, 7\}$ because the actual multiplication of 896 can be implemented at the output of coefficient 7 and left-shift it for 7 bits ($896 = 7 \times 2^7$).

In (2), *Cset* is ordered and labeled as $\{C_1 = 2721, C_2 = 831, C_3 = 105, C_4 = 7\}$. LT_1 to LT_4 is generated accordingly. The complete table of LT_2 is shown in Table 24.1. Coefficients that have a large *#nzbit* generally need more adders. They are also likely to be in the critical path of the MCM. As a result, the order of coefficients in *Cset* is important because coefficients at the beginning of *Cset* have a higher priority to synthesize. C_1 always has the highest priority and, in this case, C_4 has the lowest. Decompositions of coefficients that synthesize first generally have a larger table size because they have fewer constraints from the synthesized coefficients in *RT*. A larger table provides a larger search space that may lead to a smaller adder cost. Note that the decomposition at $x_2 = 22$ is invalid because the decomposition causes the MCM greater than the predefined $LDB = 3$. It is removed from LT_2 before proceeding to the next step. As an intermediate result, $L_1 = 33$; $L_2 = 21$; $L_3 = 19$; and $L_4 = 1$.

In (3), coefficients that require only one adder are synthesized. This is because the decompositions are likely to appear in the minima of f. Synthesizing coefficients in this step reduces the search space of f. In this example, $C_3 = 105$ and $C_4 = 7$ may be synthesized with one adder. As a result, 105 is first implemented with one adder as $7 \times 2^4 - 7$. Then, 7 is implemented with one adder as $1 \times 2^3 - 1$. The intermediate result is $Cset = \{C_1 = 2721, C_2 = 831\}$. $Dset = \{105, 7\}$. $RT = \{105 = 7 \times 2^4 - 7, 7 = 1 \times 2^3 - 1\}$. $N = 2$.

In (4), Eq. (24.11) is generated in the form of $f(x_1, x_2)$. A simplified table that shows pairs of subexpressions for C_1 and C_2 as shown in Table 24.2. To illustrate the estimated cost function f, the set of *NS* for $f(1,3)$ is $\{673, 767\}$. $f(1,3) = 4 + (3+2) = 9$. The set of *NS* for $f(24,20)$ is $\{1041\}$. $f(24,20) = 4 + (2) = 6$.

In (5), the algorithm picks the set of coefficient decompositions that is further away from the *LDB*. In general, coefficients and subexpressions having a lower average logic depth may leave more flexibility for coefficients. It reduces the probability of decompositions that violate the *LDB* constraint. In our example, only one solution is found to have a minimum of six and it is at $f(24,20)$. The algorithm goes to (6) directly.

Table 24.2 Subexpression table of $C_1 = 2721$ and $C_2 = 831$

x_1	1	2	.....	24	.....	32	33
	1	1	.....	1041	.....	4384	111
$C_1 = 2721$	673	2209		105		831	831
x_2	1	2	3	.....	19	20	21
	1	1	1	.....	961	1041	9
$C_2 = 831$	193	1087	767		7	105	105

In (6), coefficients with the most *#nzbit* are synthesized into RT, while coefficients with less *#nzbit* remain in *Cset*. This is because the sizes of the lookup tables of the remaining coefficients may be increased by adding and subtracting the newly inserted coefficients. As a result, the algorithm in the next iteration may find more common subexpressions with each other. In our example, the coefficient that has the most *#nzbit* is $C_1 = 2721$ and it has five non zero bits. Therefore, 2721 is moved from *Cset* to *Dset*. The new subexpressions are inserted into *Cset*. As an intermediate result, $Cset = \{831, 1041\}$. $Dset = \{105, 7, 2721\}$. $RT = \{105 = 7 \times 2^4 - 7, 7 = 1 \times 2^4 - 1, 2721 = 1041 + 105 \times 2^4\}$. $N = 2$.

In (7), because *Cset* is not empty, go back to (2) and the algorithm enters the second iteration. In (2), the lookup table for 831 is updated and another lookup table is generated for 1041. In (3), 831 is synthesized as $1041 - 105 \times 2^1$. Note that 1041 is a subexpression that appears in the proposed expansion of search space at $x_2 = 20$ in Table 24.1. In (6), 1041 is synthesized as $1 \times 2^{10} + 17$. In the third iteration of (3), 17 is synthesized as $1 \times 2^4 + 1$. The final synthesis of MCM is $RT = \{105 = 7 \times 2^4 - 7, 7 = 1 \times 2^4 - 1, 2721 = 1041 + 105 \times 2^4, 831 = 1041 - 105 \times 2^1, 1041 = 1 \times 2^{10} + 17, 17 = 1 \times 2^4 + 1\}$.

24.7 Results

We have implemented the algorithm in Yao et al. and extracted the data results from Flores et al. The results are shown in Table 24.4. The specifications of our filter examples are shown in Table 24.3.

Filter 1 consists of the coefficients described in Sect. 24.6. Filters 2 and 3 are the L1 filter and the Example 2 filter in Yao et al. Filters 4 to 8 are the filters 4 to 8 in Flores et al.

In Table 24.3, *length* stands for the filter length or the number of multiplications of the MCM. *Width* is the maximum binary bit size of coefficients. *#nzbit* is the maximum number of non zero terms in CSD or MSD representations. *#coeff* is the number of coefficients in *Cset* after process (1) in Sect. 24.5. *#coeff* is also the lower bound of the adder cost. *Simple CSD* is the adder cost of the simple CSD method. It is also the upper bound of the adder cost.

In Table 24.4, *LO* is the number of logic operations that equal the total number of adders and subtractors. *LD* is the logic depth. *cputime* is the processing time of our

Table 24.3 Characteristics of the MCM

Filter	length	width	#nzbit	(Lower) #coef	(Upper) CSD
1	4	12	5	4	11
2	25	13	6	13	44
3	121	15	7	51	139
4	81	12	5	28	68
5	121	12	5	34	76
6	60	14	6	20	52
7	60	14	5	29	77
8	60	14	5	28	84

Table 24.4 Result comparison

	Proposed				[4]		[7]	
Filter	LO*	LO	LD	cputime	LO	LD	LO	LD
1	7	6	3	30.078	8	3	NA	
2	21	19	3	1272.515	21	3	NA	
3	56	53	3	827.560	54	3	NA	
4	29	28	3	38.825	28	3	29	3
5	34	34	3	62.875	34	3	34	3
6	22	22	3	119.250	22	3	22	3
7	34	30	3	35.750	30	3	34	3
8	34	32	3	1280.625	32	3	35	3

proposed algorithm in seconds. LO* is the result of the proposed algorithm using the decompositions from all MSD representations alone.

LDB is set to be $\lceil \log_2(\#nzbit \text{ in Table 24.3}) \rceil$ in our proposed algorithm and in Yao et al. because the increase of *LDB* from its minimum depth would increase the delay significantly. Take filter 3 as an example—a logic depth increase from 3 to 4 would increase the critical path by 33 percent but the size of the MCM is only reduced by 2 percent.

Genetic algorithm is used to minimize the estimated cost function. The population size is set to 300 and the genetic algorithm is terminated when no improvement is found in five consecutive iterations. A brute force search is performed when the size of the problem is less than 3,000 because the computation time is comparable or less than that of using genetic algorithm.

The speed of our algorithm is acceptable in practically sized problems. Our program is coded in Matlab 7 and executed on a 2.6 GHz PC with 512MB memory. We have implemented the algorithm in Yao et al. for comparison. The results of Flores et al. are extracted directly from their paper. Although the speed cannot be directly compared, the Flores et al. results are included as a reference. The processing time of the proposed method is longer than those of Flores et al. and Yao et al. in filters 4 to 7. This is expected because the speed of convergence is slow in genetic algorithms and it also requires a longer set-up time. For a more complicated problem in filter 8, the processing time of the proposed algorithm is 1,280 seconds while it is terminated after 3,600 seconds in Flores et al. and it is about 7,200 seconds in Yao et al.

Although genetic algorithms generally do not guarantee an optimum solution, the final solution is adapted well into our CSE algorithm. For our synthesis examples of multiplierless MCM, it is shown that in filters 4 to 7, LO* is the optimal solution in the MSD solution space. Our results match the solution in Flores et al. that searches for an optimal solution using the SAT-based 0-1 ILP. LO* in filter 8 outperforms the result in Flores et al. because their program is terminated after one hour and as a result is not optimal.

The proposed expansion of search space is also exploited. The expansed search space further reduces the size of the MCM in some cases. Adder reductions are seen in filters 4, 7, and 8, where filters 4 and 7 are optimal in the MSD solution space. All logic depth is restricted to be 3. It saves about 58 percent adders than simple CSD and saves about 6 percent adders than MSD. It is also shown in filters 1 to 3 that the proposed algorithm produces a smaller LO than Yao et al. Although the relationship between the coefficients in terms of subexpressions is situtation dependent [7], the proposal search space exploits hidden and significant relationships between coefficients. It saves more adders for cases of larger filters because more hidden relationships are observed. It shows that actual area is saved using the proposed design.

24.8 Conclusion

We have proposed a novel CSE algorithm that models synthesis of coefficients into an estimated cost function. A genetic algorithm has been shown to be capable of minimizing the cost function and give a satifactory (nearly optimal) design in a reasonable time within a finite search space. Numerical examples have verified that the proposed algorithm generates coefficients that match the known optimal solutions. We have also proposed an expansion of differential coefficient solution space that generally further reduces the number of adders in the synthesis of MCM from all MSD representations, which reduces power consumption and silicon area in circuits. Futhermore, an improved version is recently proposed [11] that calculates the number of half adder and full adder for a more accurate (minimium area) synthesis. We are working on this novel concept and will publish in the future.

Acknowledgements The authors would like to thank Prof. P. Flores for sharing the filter coefficients. This work was supported in part by the Hong Kong Research Grants Council and the University Research Committee of The University of Hong Kong.

References

1. Hartley RI (1996) Subexpression sharing in filters using canonic signed digit multipliers. *IEEE Trans. Circuits Syst. II* 43(10):677–688, Oct.
2. Park I-C and Kang H-J (2002) Digital filter synthesis based on an algorithm to generate all minimal signed digit representations. *IEEE Trans. Computer-Aided Design* 21(12): 1525–1529, Dec.

3. Dempster AG and Macleod MD (2004) Digital filter design using subexpression eliminiation and all signed-digit representations. In *Proc. IEEE Int. Symp. on Circuits and Systems* 3: III 169–III 172, May 23–26
4. Yao C-Y, Chen H-H, Lin T-F, Chien C-J and Hsu C-T (2004) A novel common-subexpression-elimination method for synthesizing fixed-point FIR filters. *IEEE Trans. Circuits Syst. I*, 51(11):2215–2221, Nov.
5. Macleod MD and Dempster AG (2005) Multiplierless FIR filter design algorithms. *IEEE Signal Processing Lett.* 12(3):186–189, Mar.
6. Wang Y and Roy K (2005) CSDC: A new complexity reduction technique for multiplierless implementation of digital FIR filters. *IEEE Trans. Circuits Syst. I*, 52(9):1845–1853, Sept.
7. Flores P, Monteiro J and Costa E (2005) An exact algorithm for the maximal sharing of partial terms in multiple constant multiplications. In *Proc. IEEE Intl. Conf. Computer-Aided Design*, San Jose, CA, pp. 13–16, Nov. 6–10
8. Mazumder P and Rudnick EM (1999) *Genetic algorithms for VLSI design, layout & test automation*.Upper Saddle River, NJ: Prentice-Hall
9. Gen M and Cheng R (1997) *Genetic algorithms and engineering design*, Parsaei HR, Ed. New York, NY: John Wiley & Sons
10. Bull DR and Horrocks DH (1991) Primitive operator digital filters. *Proc. Inst. Elec. Eng. G: Circuits, Devices, Syst.* 138(3):401–412
11. Aksoy L, Flores P, Monteiro J, and Costa E (2007) Optimization of area in digital fir filters using gate-level metrics. In *Proc. IEEE Design Automation Conference (DAC)*, San Diego, CA, June 4–8

Chapter 25
Design of Intelligent Controllers Using Online-Trained Fuzzy Neural Networks for the UPFC

Tsao-Tsung Ma

25.1 Introduction

With the trend toward deregulating the power industry, it has been expected that future power systems will need to provide sufficient, secure, high-quality and real-time controllable electric power to various load centers. It is envisaged that flexible AC transmission systems (FACTS) devices [1,2] or similar power flow controllers are going to play a critical role in operating the new type of power systems under such a complex operating environment. Basically, using state-of-the-art communication technologies, high-power converters and properly designed controllers, FACTS devices can offer great control flexibility in power system operations. It has been proved that a number of important control functions and optimization issues, which were unattainable in the past, can now be easily realized by utilizing these high-performance power flow controllers with some properly developed control algorithms. One of the crucial objectives concerning FACTS device applications in modern power systems is real-time load flow control. Of the reported FACTS devices, it has been well accepted that the unified power flow controller (UPFC) is the most versatile and powerful one [3, 4] that can provide effective means for controlling the power flow and improving the transient stability of a power network [5, 6]. However, the UPFC has multiple operating modes and complex system dynamics that require advanced controllers to achieve satisfactory performances. In the literature, a number of conventional controllers have been proposed for the UPFC to perform various power system control functions; however, most of them are linear controllers designed on a linearized model of the controlled power system. To achieve robust control effects and better dynamic control performance some advanced controllers with adaptive features and the ability to learn will be required.

Tsao-Tsung Ma
Department of EE, CEECS, National United University, Taiwan
No. 1, Lien-Da, Kung-Ching Li
Miaoli 36003, Taiwan, R.O.C.
e-mail: tonyma@ieee.org

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

It is well known that adaptive control is able to improve the control of nonlinear and complex systems with fast changing dynamics, such as power systems [7]. This is a result of the concerned dynamics being continually identified by a model. In this aspect, fuzzy and neural networks are very suitable for multivariable applications—especially for the system with unclear and complex dynamics—since they can easily identify the variations of parameters and the interactions between the inputs and outputs of the controllers. The ability to learn and store information about system nonlinearities allows neural networks to be used for modeling and designing intelligent controllers for power systems [8]. Thus, they offer potential alternatives to those applications where traditional linear controllers are not working adequately. Advantages of neural-network-based controllers over conventional controllers are that they can adapt themselves to changes in system operating conditions automatically, unlike conventional controllers whose performance degrades for such changes and which must be retuned to give the desired performance. This chapter describes the design of a fuzzy neural network (FNN) [9–11] online-tuned PI controller for achieving decoupled control of the UPFC and the power system parameters in a two-bus test power system. The design of the series and shunt continually online trained FNN and the proportional plus integral (PI) controllers for the series and the shunt branches of a UPFC are based on the concepts of intelligent and indirect adaptive control schemes. In this chapter, two sets of FNN are used for the complete UPFC control system. A comparative study of the performances of FNN tuned and conventional PI controllers for real-time power flow regulation is presented and discussed in full.

25.2 Hardware Configuration and Its Control Modes

In a UPFC system, the series branch provides the main power flow control functions by injecting a voltage with controllable magnitude and phase angle in series with the line via an insertion transformer. This injected voltage acts essentially like a synchronous AC voltage source. The transmission line current flows through this voltage source resulting in a reactive and active power exchange between itself and the AC system [12]. The basic function of the shunt branch is to generate or absorb the real power demanded by the series branch at the common DC link. The real power demanded by the series branch at the DC link is converted back to the AC side of the shunt branch and fed into the system bus via a shunt-connected transformer. In addition to this, the shunt branch can also generate or absorb a controllable reactive power if desired and thereby provides independent shunt reactive compensation for the system. To have a clear picture of the UPFC system, Fig. 25.1 shows the UPFC hardware configuration, control structure and the related parameters. It is clear that control of real and reactive power in the transmission line can be achieved by injecting a series voltage, V_{pq}, with an appropriate magnitude and angle into the controlled transmission line, and the same control theory can also be applied to the control of the shunt branch. For the purposes of addressing the design procedure of

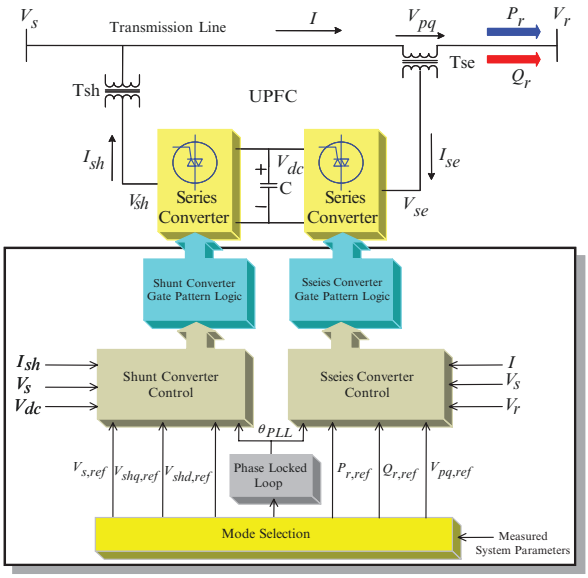


Fig. 25.1 UPFC hardware configuration and its overall control structure

the proposed UPFC series and shunt controllers, the most commonly used UPFC control functions are briefly presented in the following subsections.

25.2.1 UPFC Control Modes (Series Branch)

In normal operations of a UPFC series branch, a number of control functions, e.g., direct system voltage regulation, real-time line impedance and phase angle regulations have been widely used to achieve various power system control objectives. These control functions can be defined as Mode A, B, C and D according to the way of adding the injected voltage, V_{pq}, to the controlled transmission line. These UPFC control modes (Mode A, B, C and D) can be further illustrated by the mathematical formulation of controlled active and reactive power and UPFC parameters.

25.2.1.1 Mode A: Direct Voltage Regulation Mode

In this control mode, the inserted series voltage, V_{pq}, must be controlled either in phase (0 degrees) or in counter phase (180 degrees) with the existing bus voltage, V_S, and thus changing only the magnitude of V_S. The phasor diagram of this control mode is shown in Fig. 25.2. Here, the injected voltage, $V_{pq} = \pm\Delta V$, the transmitted active power, P_r, and reactive power, Q_r, at the receiving end, can be expressed as Eqs. (25.1) and (25.2).

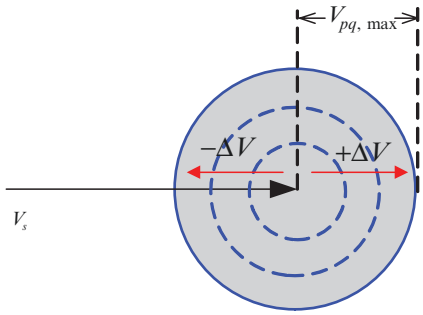


Fig. 25.2 Voltage Regulation Mode

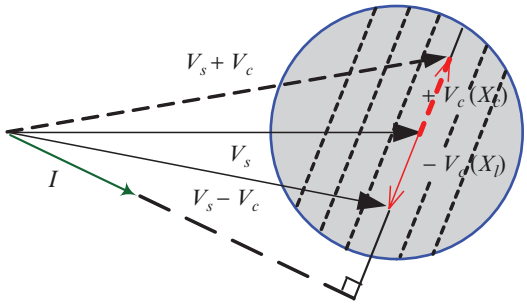


Fig. 25.3 Line Impedance Compensation Mode

$$P_r = \frac{(V_s \pm \Delta V) \cdot V_r}{X} \sin \delta \tag{25.1}$$

$$Q_r = \frac{(V_s \pm \Delta V) \cdot V_r}{X} (1 - \cos \delta) \tag{25.2}$$

In Eqs. (25.1) and (25.2), V_s and V_r respectively represent the sending end and receiving end voltage. δ represents the phase angle between V_s and V_r. X represents transmission line impedance.

25.2.1.2 Mode B: Line Impedance Compensation Mode

In this control mode, the phase angle of the inserted voltage, V_{pq}, must be kept at either in 90 degrees leading or lagging the transmission line current, I, to respectively achieve the capacitive or inductive impedance compensation. In Fig. 25.3, the voltage, V_c, decreases or increases the effective voltage drop across the line segment impedance, X, according to whether V_c lags or leads I, as illustrated in Fig. 25.3. Here, $V_{pq} = V_c$, is injected in quadrature with the line current phasor.

The transmitted power flow, P_r and Q_r, can be expressed as:

$$P_r = \frac{V_S \cdot V_R}{X\,(1-k)} \sin \delta \tag{25.3}$$

$$Q_r = \frac{V_S \cdot V_R}{X\,(1-k)} (1 - \cos \delta) \tag{25.4}$$

where $k = \frac{X_{comp}}{X}$, X_{comp} represents the equivalent UPFC series compensation impedance, $X_{comp} = \frac{\pm V_c}{I}$.

25.2.1.3 Mode C: Phases Shifting Mode

In this control mode, the inserted series voltage, V_{pq}, must be controlled either in 90 degrees leading or lagging the existing bus voltage, V_s, to achieve the control of transmission phase angle, $\Delta\delta$. The phasor diagram of this control mode is shown in Fig. 25.4, where, $V_{pq} = V_{\Delta\delta}$, is injected to the system with an angular relationship of 90 degrees leading or lagging to V_s that achieves the desired $\Delta\delta$ phase shift (advance or retard) and keeps the voltage magnitude unchanged.

In this control case, the transmitted power flow, P_r and Q_r, can be expressed as:

$$P_r = \frac{V_s \cdot V_r}{X} \sin(\delta + \Delta\delta) \tag{25.5}$$

$$Q_r = \frac{V_s \cdot V_r}{X} [1 - \cos(\delta + \Delta\delta)] \tag{25.6}$$

25.2.1.4 Mode D: Unified Control Mode

The Mode D is a unified control of the above three modes. It is executed by simultaneously injecting a resultant voltage phasor combining the terminal voltage regulation, series line impedance compensation and phase shifting voltage

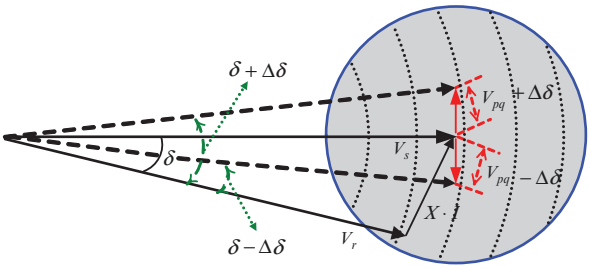


Fig. 25.4 Phases Shifting Mode

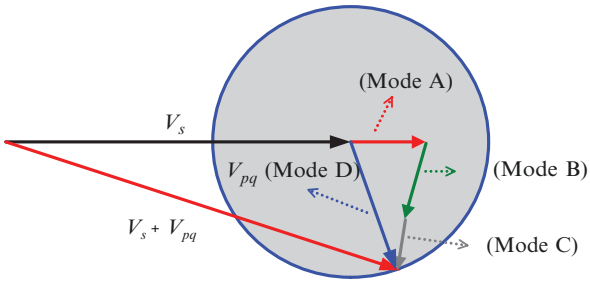


Fig. 25.5 Unified Control Mode

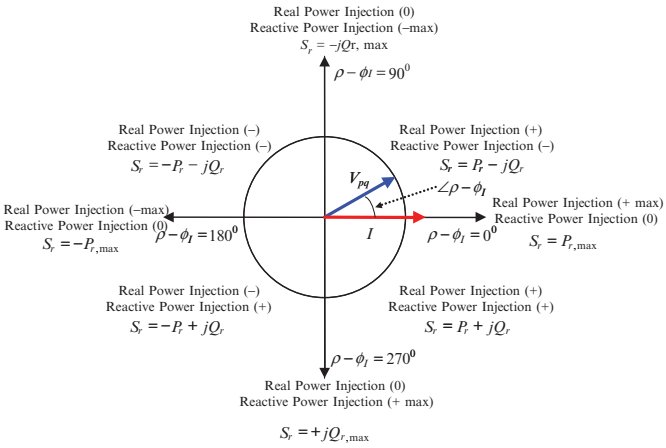


Fig. 25.6 The active and reactive power exchange between UPFC and the power system

components. As shown in Fig. 25.5, $V_{pq} = V_{pq}(\Delta V) + V_{pq}(X) + V_{pq}(\Delta\delta)$. Basically, in this control mode the UPFC can be controlled to work as any combinations of mode A, B, and C as desired. In other words, the inserted voltage, V_{pq}, can be controlled to have both its magnitude and phase angle varying freely in all four quadrants.

Here, the transmitted power flow, P_r and Q_r, can be expressed as:

$$P_r = \frac{(V_s + V_{pq}) \cdot V_r}{X} \sin\delta \tag{25.7}$$

$$Q_r = \frac{(V_s + V_{pq}) \cdot V_r}{X} (1 - \cos\delta) \tag{25.8}$$

To have a clear picture of the control modes provided by the series branch of a UPFC, Fig. 25.6 shows the feasible active and reactive power exchanges between UPFC and the controlled power system under different operating modes and system conditions.

25.2.2 UPFC Control Modes (Shunt Branch)

The UPFC shunt converter is operated to draw a desired current from the transmission line. One component of this current is automatically determined by the requirement of balancing the real power with the series inverter. The remaining current component is reactive and can be set to any desired reference level (inductive or capacitive) within the capability of the inverter. The reactive compensation control mode of the shunt inverter is very similar to those commonly employed conventional static var compensators.

For presenting the operating principles of the UPFC shunt converter, Fig. 25.7 represents only the shunt converter connected to a bus. When the fundamental com-

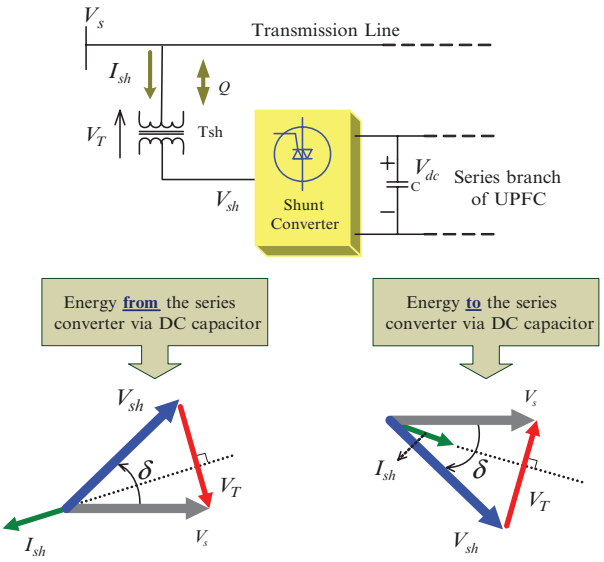


Fig. 25.7 Control of active and reactive power generation in the shunt converter of UPFC

ponent of the shunt converter voltage is controlled greater than the bus voltage, a leading reactive current is drawn from the bus, and the shunt converter behaves as a capacitor.

When the fundamental component of the shunt converter voltage is controlled smaller than the bus voltage, a lagging reactive current is drawn from the bus and the converter behaves as an inductor. In addition, the shunt converter can cause a temporary flow of active power, sufficient to charge or discharge the DC capacitor and to achieve a constant of the DC voltage. This is done by slightly lagging or leading a small angle (δ) of the converter output voltage with respect to the bus voltage, as shown in Fig. 25.7.

The leading (capacitive) reactive current flowing into the shunt converter increases the voltage amplitude of the controlled bus, while the inductive current decreases the bus voltage. To force the shunt converter to draw more capacitive or inductive current, the active power flow flowing into the shunt converter can be controlled to charge (or discharge) the DC capacitor voltage.

25.2.2.1 Reactive Power Control Mode

In reactive power control mode, the reference input is an inductive or capacitive var demand. The shunt converter controller translates the var reference into a corresponding shunt current demand and adjusts the gating pattern of the converter to institute the desired current. The control uses current feedback signals obtained from the output current of the shunt converter as the current reference. A feedback signal representing the DC-link voltage is also required.

25.2.2.2 Automatic Voltage Control Mode

In automatic voltage control mode, the reactive current is automatically regulated to maintain the transmission line voltage at the point of connection to a reference value, with a defined droop characteristic. The droop factor defines the per unit voltage error and the per unit reactive current within the current range of the converter. The automatic voltage controller uses voltage feedback signals obtained from the controlled bus.

25.3 Design of Fuzzy Neural Network Controllers

25.3.1 UPFC P-Q Controllers

As addressed previously, the active power (P) and reactive power (Q) in the transmission line are controlled by the inserted series synchronous voltage, V_{pq}, while the control of bus voltage and the voltage of the DC-interface is achieved by adjusting the output voltage of the shunt inverter, V_{sh}. The basic control scheme for the UPFC series and shunt inverters can be developed by using PQ-decoupled PI control means [13]. The outputs of these control systems are the corresponding modulation indices and phase-shift signals of the internal converters. In this chapter, Figs. 25.8 and 25.9 respectively show the conceptual control block diagram of the proposed FNN based PI controllers for the UPFC to perform the shunt reactive compensation and Mode A, B and C controls provided by the UPFC series branch. The SW1∼4 shown in Fig. 25.8 and Fig. 25.9 is designed to select the type of control scheme to be investigated (the FNN tuned PI control or simply PI control scheme). In Figs. 25.8 and 25.9, the reference model is designed as a standard second order system, $\frac{\omega_n^2}{S^2+2\rho\omega_n+\omega_n^2}$, where the rise time is set to be $t_s = \frac{4.5\rho}{\omega_n} = 0.1$ second, ρ represents the damping ratio, designed to be unity. The design of FNN controllers is explained in the next subsection.

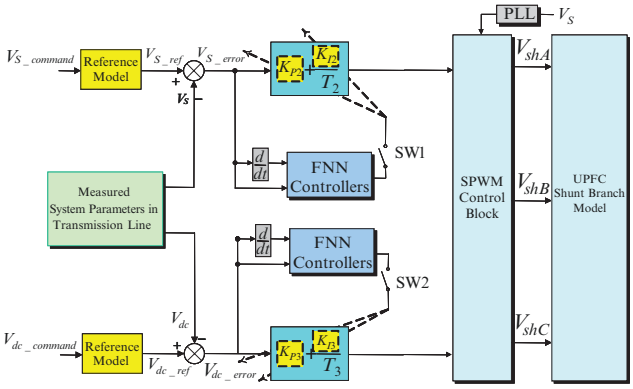


Fig. 25.8 FNN based voltage controllers for the UPFC shunt branch

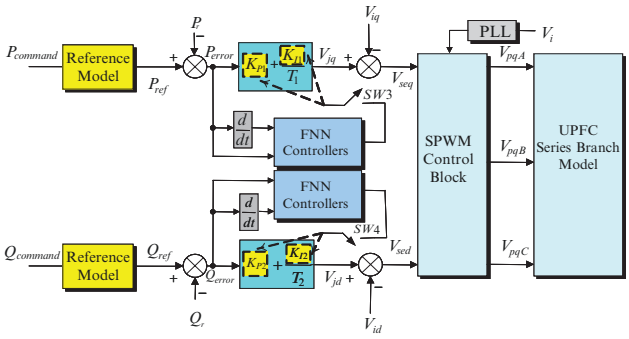


Fig. 25.9 FNN based P-Q controllers for the UPFC series branch

25.3.2 Design of Fuzzy Neural Network Controllers

As stated in the previous sections, the main purpose of this study is to design two sets of FNN controllers—one for the series inverter and the other for the shunt inverter—to achieve on-line tuning two sets of pre-designed UPFC PI controllers for eliminating the cross interferences of power flow control parameters and adapting the changing hybrid dynamics of the UPFC and the power system conditions.

25.3.2.1 Structure of the FNN

In this chapter, the multi-layered FNN shown in Fig. 25.10 is utilized to design the proposed UPFC controllers. The FNN has four layers: input layer, membership layer, rule layer, and output layer. In this application case, the FNN structure is organized into 2 input variables (the control error and its derivative), 3-term nodes for each input variable, 1 output node (Kp or Ki in this case), and 9 rule nodes. Layer 1 accepts input variables. Its nodes represent input linguistic variables, i.e., Negative (N), Zero (Z) and Positive (P). Layer 2 is used to calculate Gaussian membership values. Nodes in this layer represent the terms of the respective linguistic variables. Layer 3 forms the fuzzy rule base representing fuzzy rules. Links before Layer 3 represent the preconditions of the rules, and the links after Layer 3 represent the consequences of the rule nodes. Layer 4 is the output layer, where each node is for an individual output of the system. The links between Layer 3 and Layer 4 are connected by the weighting values to be on-line trained.

25.3.2.2 Layered Operation of the FNN

To have a clear insight of FNN principles, this subsection presents the details regarding the signal propagation and the operation functions of the nodes in each

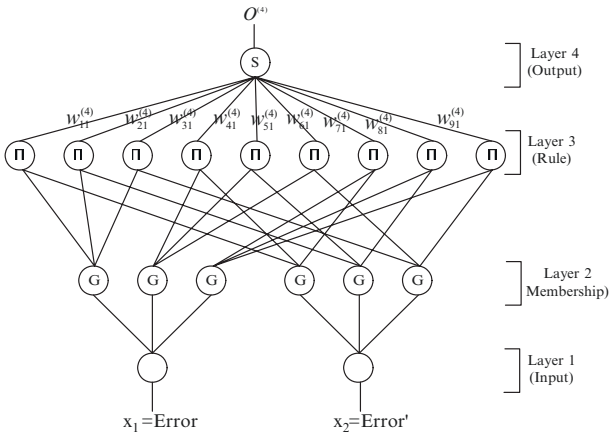


Fig. 25.10 The multi-layered structure of the proposed FNN

FNN layer. In the following description, $u_i^{(k)}$ denotes the ith input of a node in the layer k; $O_i^{(k)}$ denotes the ith node output of layer k.

Layer 1: Input Layer: The nodes in this layer only transmit input values to the next layer directly. The node input and the node output are represented as:

$$\text{node input}: \; u_i^{(1)} = x_i$$
$$\text{node output}: \; O_i^{(1)} = u_i^{(1)}$$

where, $i = \{1,2\}$ in this case and the link weight at layer 1, $w_i^{(1)}$, is unity.

Layer 2: Membership Layer: In this layer, each node performs a membership function. The Gaussian function is used here as a membership function.

$$\text{node input}: \; u_{ij}^{(2)} = -\frac{(u_i^{(1)} - m_{ij})^2}{(\sigma_{ij})^2}$$
$$\text{node output}: \; O_{ij}^{(2)} = \exp(u_{ij}^{(2)})$$

where, m_{ij} and σ_{ij} are respectively the center and the standard deviation of the Gaussian membership function. The subscript ij indicates the jth term of the ith input x_i, where, $j = \{1.2.3\}$. Each node in this layer has two adjustable parameters, that is m_{ij} and σ_{ij}.

Layer 3: Rule Layer: The nodes in this layer are called rule nodes. The following AND operation is applied to each rule node.

$$\text{node input}: \; u_{ij}^{(3)} = O_{ij}^{(2)} \cdot w_{ij}^{(3)}$$
$$\text{node output}: \; O_k^{(3)} = \prod_{i=1}^{2} u_{ij}^{(3)}$$

here, $k = \{1,2\ldots,9\}$. The output of a rule node represents the "firing strength" of its corresponding rule. The $w_{ij}^{(3)}$ is assumed to be unity.

Layer 4: Output Layer: Each node in this layer is called an output linguistic node. The node output is a linear combination of the consequences obtained from each rule. That is:

$$\text{node input}: \; u_q^{(4)} = O_k^{(3)} \cdot w_{kq}^{(4)}$$

$$\text{node output}: \; O^{(4)} = \sum_{k=1}^{9} u_q^{(4)}$$

Here, $q = \{1\}$, the link weight $w_{kq}^{(4)}$ is the output action strength of the qth output associated with the kth rule. The $w_{kq}^{(4)}$ are the tuning factors of this layer, and $O^{(4)} = \Delta K_P$ or ΔK_I.

On-Line Learning Algorithm

To apply the on-line learning algorithm to the proposed FNN, the gradient descent method is adopted from [14]. Let's take the control of real power as an example. First, the energy function can be defined as:

$$E(k) = \frac{1}{2}(P_{command} - P_r)^2 = \frac{1}{2}\Delta P_{error}(k)^2 \tag{25.9}$$

where $P_{command}$ and P_r represent the power flow command and the actual power flow in the transmission line; $\Delta P_{error}(k)$ denotes the output error between the power flow command and the actual power flow. Then, the learning algorithm based on back propagation is described in the following.

Layer 4: The error term to be propagated is given by

$$\delta_q^{(4)} = -\frac{\partial E}{\partial u_q^{(4)}} = -\frac{\partial E}{\partial e} \cdot \frac{\partial e}{\partial P_{line}} \cdot \frac{\partial P_{line}}{\partial O^{(4)}} \cdot \frac{\partial O^{(4)}}{\partial u_q^{(4)}} \tag{25.10}$$

and the update rule of $w_{kq}^{(4)}$

$$w_{kq}^{(4)}(k) = w_{kq}^{(4)}(k-1) + \Delta w_{kq}^{(4)}(k) + \alpha(w_{kq}^{(4)}(k-1) - w_{kq}^{(4)}(k-2)) \tag{25.11}$$

where, $\Delta w_{kq} = -\eta \cdot \frac{\partial E}{\partial w_{kq}} = \eta \cdot \delta_q^{(4)} \cdot u_q^{(4)}$

η is the learning rate of the FNN weights and α is the dynamic factor.

Layer 3: Since the weights in this layer are unity, only the error term needs to be calculated and propagated:

$$\delta_k^{(3)} = -\frac{\partial E}{\partial u_k^{(3)}} = -\frac{\partial E}{\partial O^{(4)}} \cdot \frac{\partial O^{(4)}}{\partial u_q^{(4)}} \cdot \frac{\partial u_q^{(4)}}{\partial O_k^{(3)}} \cdot \frac{\partial O_k^{(3)}}{\partial u_k^{(3)}} = \delta_q^{(4)} w_{kq}^{(4)} \tag{25.12}$$

Layer 2: The multiplication operation is done in this layer. The error term is computed as follows:

$$\begin{aligned}\delta_j^{(2)} &= -\frac{\partial E}{\partial u_{ij}^{(2)}} = -\frac{\partial E}{\partial O_q^{(4)}} \cdot \frac{\partial O_q^{(4)}}{\partial u_q^{(4)}} \cdot \frac{\partial u_q^{(4)}}{\partial O_k^{(3)}} \cdot \frac{\partial O_k^{(3)}}{\partial u_k^{(3)}} \cdot \frac{\partial u_k^{(3)}}{\partial O_{ij}^{(2)}} \cdot \frac{\partial O_{ij}^{(2)}}{\partial u_{ij}^{(2)}} \\ &= \sum_k \delta_k^{(3)} O_k^{(3)}\end{aligned} \tag{25.13}$$

Similarly, the update laws of m_{ij} and σ_{ij} are

$$m_{ij}(k) = m_{ij}(k-1) + \Delta m_{ij} + \alpha(m_{ij}(k-1) - m_{ij}(k-2)) \tag{25.14}$$

$$\sigma_{ij}(k) = \sigma_{ij}(k-1) + \Delta\sigma_{ij} + \alpha(\sigma_{ij}(k-1) - \sigma_{ij}(k-2)) \tag{25.15}$$

where,

$$\Delta m_{ij} = -\eta_m \frac{\partial E}{\partial m_{ij}} = -\eta_m \frac{\partial E}{\partial O_{ij}^{(2)}} \cdot \frac{\partial O_{ij}^{(2)}}{\partial u_{ij}^{(2)}} \cdot \frac{\partial u_{ij}^{(2)}}{\partial m_{ij}} = \eta_m \delta_m^{(2)} \frac{2\left(x_i^2 - m_{ij}\right)}{\left(\sigma_{ij}\right)^2} \tag{25.16}$$

$$\Delta\sigma_{ij} = -\eta_\sigma \frac{\partial E}{\partial \sigma_{ij}} = -\eta_\sigma \frac{\partial E}{\partial O_{ij}^{(2)}} \cdot \frac{\partial O_{ij}^{(2)}}{\partial u_{ij}^{(2)}} \cdot \frac{\partial u_{ij}^{(2)}}{\partial \sigma_{ij}} = \eta_\sigma \delta_m^{(2)} \frac{2\left(x_i^2 - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^3} \tag{25.17}$$

η_m and η_σ are respectively the learning-rate parameters of the mean and the standard deviation of the Gaussian function for the FNN.

25.4 Case Studies and Simulation Results

For identifying the detailed dynamics of the UPFC, the two-bus power system embedded with a UPFC as shown in Fig. 25.11 is simulated in a PSCAD/EMTDC environment while the FNN controllers are implemented in a MATLAB program module. PSCAD/EMTDC is an electromagnetic transient simulator for complex electric networks with the capability of modeling power electronics, controls and nonlinear power networks. It also has a flexible interface for exchanging parameters with other simulation programs. The studied power system shown in Fig. 25.11 comprises two buses that are connected by two sections of standard transmission lines.

The UPFC is placed between the two sections of the transmission lines. The detailed system data is given in each of the case studies. The simulation cases chosen for evaluating the dynamic power flow control performances of the UPFC controllers are based on two different control schemes, i.e., conventional PI controllers

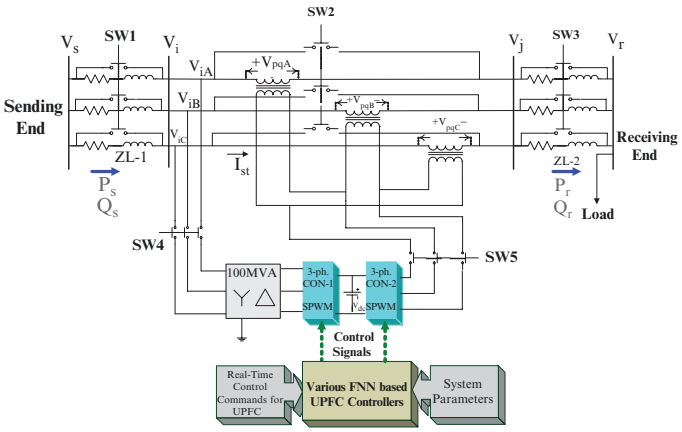


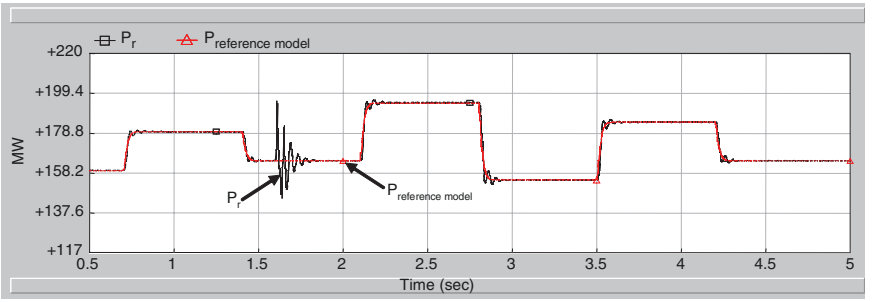
Fig. 25.11 The test power system with a UPFC

without considering the coupling effects of controlled parameters and the proposed FNN based decoupled control schemes. Based on a given set of system conditions, typical simulation results concerning the DC interface parameters and the real and reactive power flow control objectives, respectively performed by the shunt and series branches of the UPFC (Mode A), are presented in this section with some discussions.

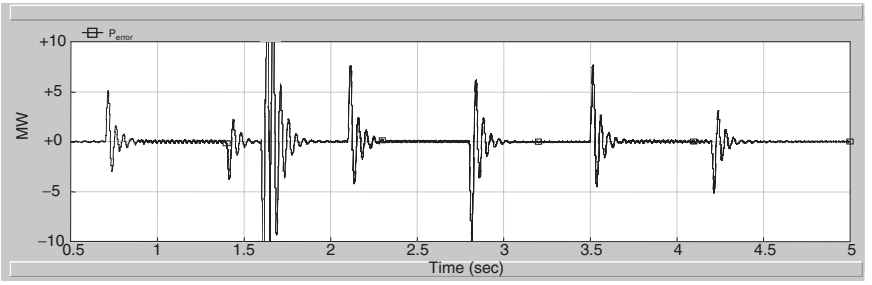
25.4.1 Simulation Results of the UPFC Series Control (Mode A)

25.4.1.1 Conventional Proportional-Integral Controllers

As shown in Figs. 25.8 and 25.9, two sets of conventional PI based UPFC power flow controllers can be constructed with SW1, SW2, SW3 and SW4 off. The initial power system parameters and the parameters used in the PI controllers are chosen as follows: generator voltage (at the sending end): 230KV$\angle 80°$, system frequency: 60 Hz, line impedance: 5.29 + j52.9, DC-link voltage of the UPFC: 43kV, DC capacitor: 4000μF, PI controller parameters: $K_{P1} = 1$, $K_{I1} = 0.25$, $K_{P2} = 5$, $K_{I2} = 1$, $K_{P3} = 0.5$, $K_{I3} = 1$. In this study, the location of UPFC is assigned to be at the left-side of the controlled transmission line and a fault is simulated at 1.6 second of the simulation time for a duration of 50ms. For this simulation case, Fig. 25.12(a) shows the tracking control results of the active power. Fig. 25.12(b) shows the corresponding control error. The corresponding series converter modulation index is shown in Fig. 25.13. For shunt branch control loops, Figs. 25.14(a) and (b) respectively show the DC-link voltage and its control error between actual voltage and reference voltage of 43KV. As can be seen from the results of DC-link voltage control error, the DC-link voltage varies greatly during the transient period of power flow control.



(a)



(b)

Fig. 25.12 The results of controlled active power using conventional PI controller (Mode A): **(a)** The active power and reference model, **(b)** The error between the active power and the reference model

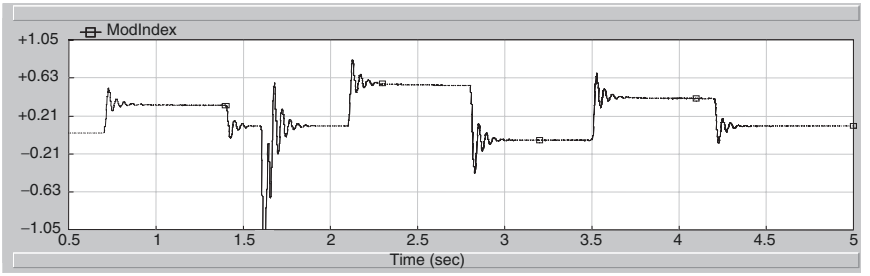


Fig. 25.13 The modulation index of UPFC series converter using conventional PI controller (Mode A)

25.4.1.2 Online Gain Tuning Proportional-Integral Controllers Using FNN

In this subsection, two sets of separate FNN tuned PI controllers as shown in Figs. 25.8 and 25.9 are used for controlling the UPFC by setting SW1 on, SW2 on, SW3 on, SW4 on. The simulation results of the active power tracking control are shown in Fig. 25.15(a). Fig. 25.15(b) shows the corresponding control error. It can be clearly seen that the UPFC using FNN tuned PI controller can achieve much better control performance than that presented in the previous subsection, during

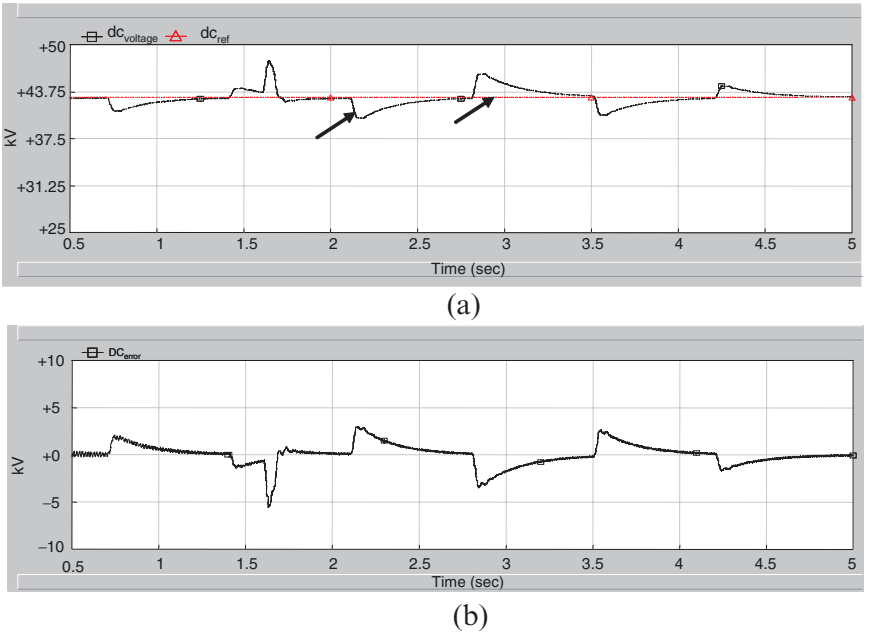


Fig. 25.14 The DC link voltage control using conventional PI controller (Mode A): **(a)** The DC link voltage and reference model, **(b)** The error between DC link voltage and the reference model

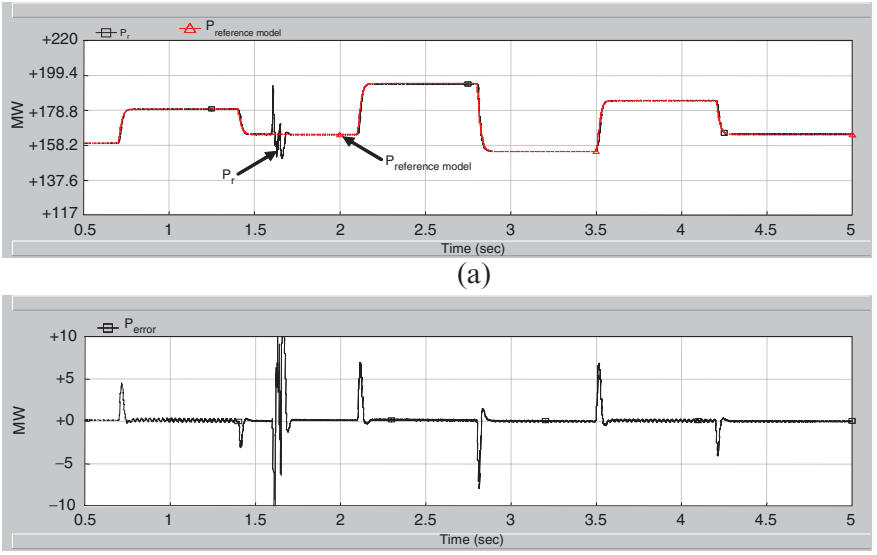


Fig. 25.15 The results of controlled active power using FNN tuned PI controller (Mode A): **(a)** The active power and reference model, **(b)** The error between the active power and the reference model

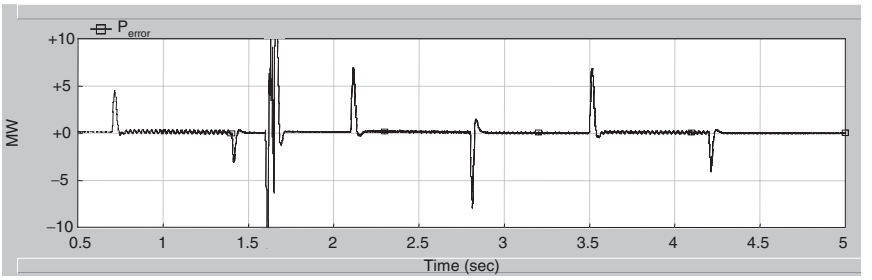


Fig. 25.16 The modulation index of UPFC series converter using FNN tuned PI controller (Mode A)

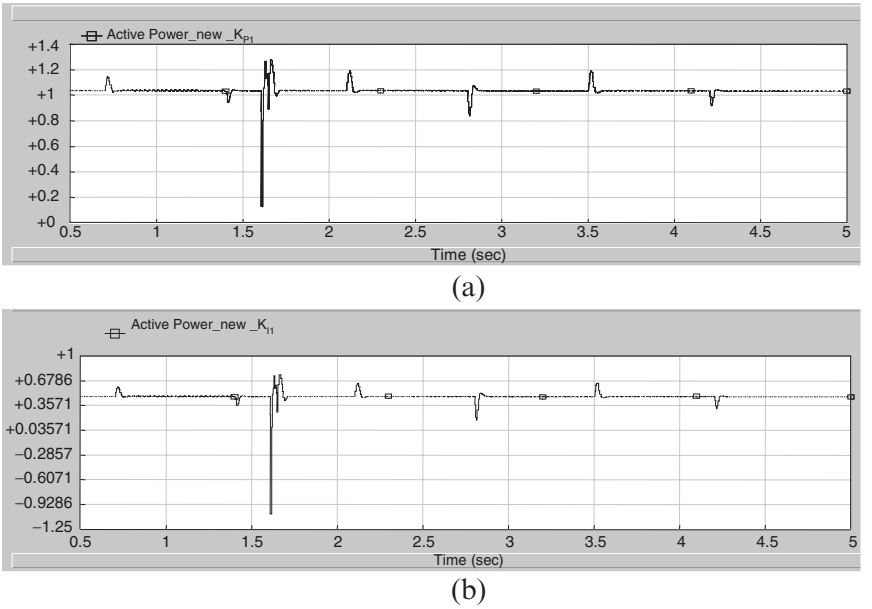
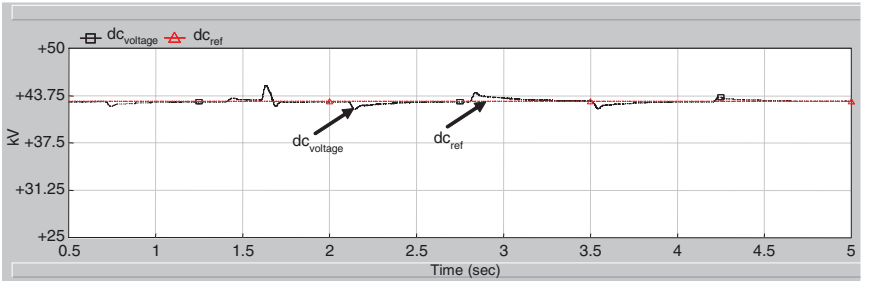
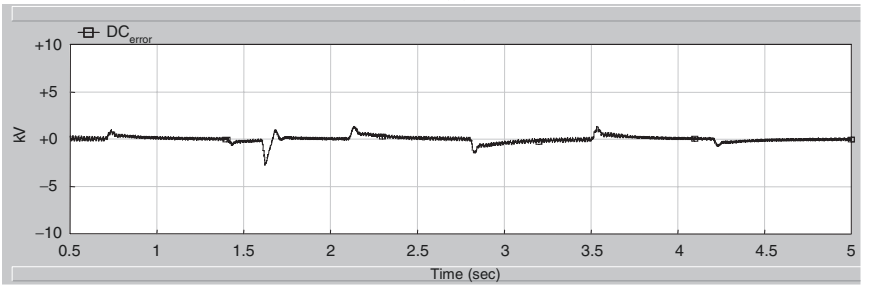


Fig. 25.17 The parameters of FNN tuned PI controller for the active power control: **(a)** The FNN tuned PI controller (parameter K_{P1}), **(b)** The FNN tuned PI controller (parameter K_{I1})

both steady and transient states. The corresponding series converter modulation index is shown in Fig. 25.16. Figs. 25.17(a) and (b), respectively, show the results of FNN tuned PI controller parameters, K_{P1} and K_{I1}. Obviously, when the reference value for the active power is changed, the FNN controllers can real-time produce the compensating signals for the K_{P1} and K_{I1} of the PI controllers. For shunt branch, Fig. 25.18 shows the control results of the DC-link voltage. Due to the FNN based controllers having better dynamic responses, when the interference occurs in the control of active power, the DC-link voltage can respond fast enough to reduce the transient effects. The corresponding modifications on the parameters of the FNN tuned PI controller, K_{P3} and K_{I3} to achieve the above control effects, are respectively shown in Figs. 25.19(a) and (b).

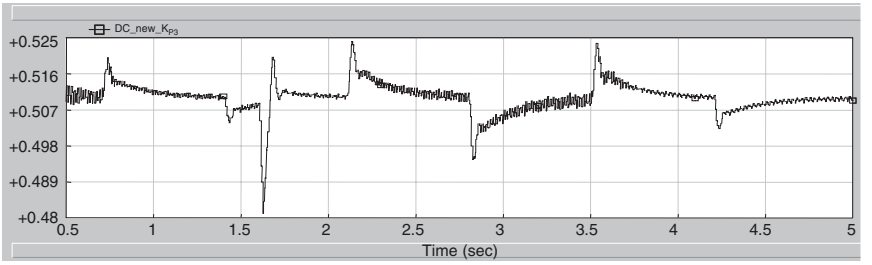


(a)

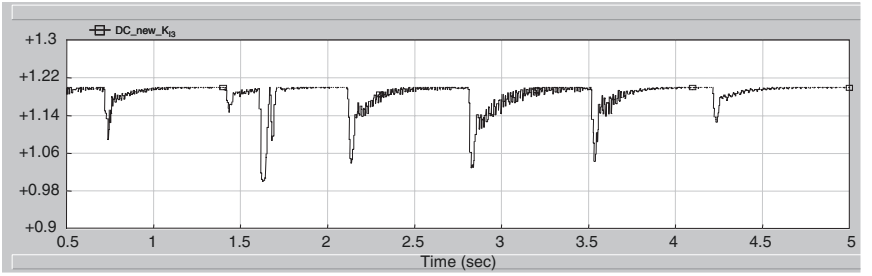


(b)

Fig. 25.18 The DC link voltage control using FNN tuned PI controller (Mode A): **(a)** The DC link voltage and reference model, **(b)** The error between DC link voltage and the reference model



(a)



(b)

Fig. 25.19 The parameters of FNN tuned PI controller for DC link voltage control: **(a)** The FNN tuned PI controller (parameter K_{P3}), **(b)** The FNN tuned PI controller (parameter K_{I3})

25.5 Conclusion

This chapter has presented the design of a novel power flow control scheme for the unified power flow controller (UPFC) to improve the dynamic control performance. Two sets of continually online trained FNN controllers are designed to provide adaptive and decoupled control of the series and shunt inverters of the UPFC according to the power flow and voltage control commands. A set of comprehensive simulation results has been presented in this chapter. It has been shown that the proposed FNN tuned PI controllers are able to successfully adapt themselves to the hybrid complex dynamics of a UPFC and the controlled power system over a variety of control modes and a wide range of system operating conditions. Based on the simulation results, it is obvious that the proposed FNN aided PI controllers are able to control the UPFC much better than the conventional PI controllers in all control modes. The superior performance of the new approach over conventional PI controllers can be explained as a result of the online, nonstop training of the FNN. In addition, as addressed in this chapter the FNN with the inherent feature of dynamic mapping can demonstrate good control performance in the presence of uncertainties. In the existing UPFC P-Q control schemes the PI controllers are the most commonly used, which tend to cause large interactions between the active and reactive power flows, while the proposed FNN based control schemes are able to overcome the inevitable problems of P-Q interactions especially when the operating point of the controlled power system changes greatly. Future work concerning this topic will be directed to extending the proposed control strategy to transient stability control of a large power system with multiple FACTS devices.

Acknowledgements The author would like to acknowledge the financial support of the National Science Council of Taiwan, R.O.C. for this research through the contract number: NSC-95-2221-239-050.

References

1. Hingorani NG (1993) Flexible AC transmission. IEEE Spectrum 1:40–45
2. Edris A-A (1997) Proposed terms and definitions for flexible AC transmission systems (FACTS). IEEE Trans. on Power Delivery 12(4): 1848–1853
3. Gyugyi L (1992) Unified power-flow control concept for flexible AC transmission system. IEEE Proceedings-C, 139 (4):323–33l
4. Schauder CD, Hamai DM, Edris A (1998) Operation of the unified power flow controller (UPFC) under practical constraints. IEEE Trans on Power Delivery 13(2): 630–639
5. Lo KL, Ma TT (2001) UPFC damping control strategy based on transient energy function. The International Journal of Electric Power Systems Research (EPSR) 56:195–203
6. Lo KL, Ma TT (2000) A robust UPFC damping scheme using pi and ann based adaptive controllers. The International Journal for Computation and Mathematics in Electrical Engineering (COMPEL) (19)3:878–902
7. Wu B, Malik OP (2006) Multivariable adaptive control of synchronous machines in a multimachine power system. IEEE Trans. on Power Systems. 21(4):1772–1781

8. Amjady N (2006) Generation adequacy assessment of power systems by time series and fuzzy neural network. IEEE Trans. on Power Systems. 21(3):1340–1349
9. Narendra KS, Parthasarathy K (1990) Identification and control of dynamical system using neural networks. IEEE Trans. Neural Networks 1:4–27
10. Venayagamoorthy GK, Harley RG (2001) A continually online trained neurocontroller for excitation and turbine control of a turbo generator. IEEE Trans. on Energy Conversion 16(3):261–269
11. Heinzinger G, Fenwick D, Paden B, Miyazaki F (2001) Stability of Takagi-Sugeno recurrent fuzzy neural networks for identification and control of dynamic systems. In: Proc. IEEE Int. Conf. on Fuzzy Systems, Melbourne, Australia, pp 537–540
12. Lo KL, Ma TT, Trecat J, Crappe M (1998) Detailed real-time simulation and performance analysis of UPFC using electromagnetic transients program (EMTP). In: Proc. of POWERCON '98, Beijing, China, 1, pp 889–894
13. Dong LY, Zhang L, Crow ML (2002) A new control strategy for the unified power flow controller. In: Proc. of IEEE-PES Winter Meeting, 1, pp 562–566
14. Leu YG, Lee TT, Wang WY (1997) On-line tuning of fuzzy-neural network for adaptive control of nonlinear dynamical systems. IEEE Trans. Syst. Man Cybern. 1: pp 1034–1043

Chapter 26
Use of Soft Computing Components to Improve Transient/Steady-State of Control Systems

PenChen Chou, TsiChian Hwang, and TsiChow Chang

26.1 Introduction

In recent years, soft computing techniques have been widely applied to the controller design of a variety of control systems in which fuzzy-logic controllers are used for unknown plant or mathematical model that is highly nonlinear or hardly derived. Knowledge engineers translate the idea of control discipline from expert operators to applicable fuzzy inference systems (FIS). Secondly, artificial neural networks (NN) are wisely used for pattern recognition, classification, function approximations, fault diagnosis, prediction of stock markets, and neural controls. The necessary requirement for neural control relies on the possibility of identifying result as well as the learning algorithm suitable for the specified problem. Both fuzzy-logic controllers (FLC) and neural controllers (NC) are nonlinear types; however, NC shows smoother variation than does FLC. Besides, ranges of input/output fuzzy sets are dominant selection factors. With wrong selection or no inclusion of corresponding scaling factors for scaling the input/output range, even an FLC, without knowing the characteristics of the plant, would not work properly. This restriction is apparently relieved by NC control. As to NC design, identification of dynamic systems becomes important; otherwise, there is no information from which the NC can learn. If an NC is used in an open-loop control for the convenience of learning an easy to apply, robustness is not generally achieved, whether on parameter variation or noise immunity capability. On the other hand, if NC is used in a closed-loop way, availability of weights becomes difficult because the learning algorithm can no longer be applied. Therefore, smart optimization algorithms such as genetic algorithms (GA) or particle swarm intelligence (PSO) can be candidates for finding the best values of weights for NC.

PenChen Chou, TsiChian Hwang, and TsiChow Chang
EE Department, DaYeh University, DaTrin village, ChungHwa County, Taiwan
e-mail: choup@tcts.seed.net.tw

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

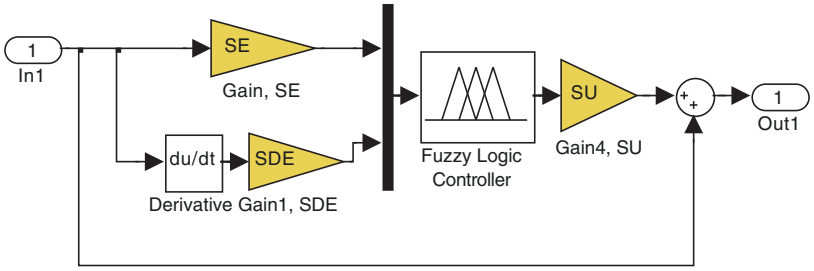


Fig. 26.1 1+ fuzzy structure

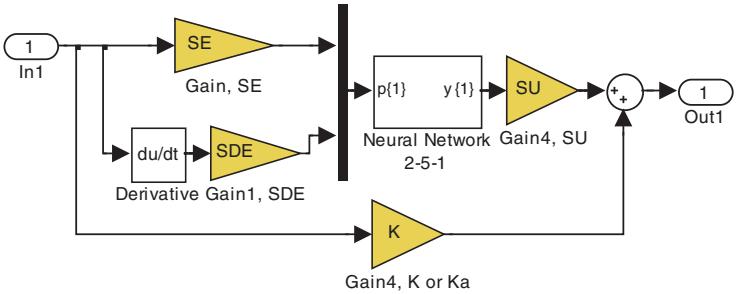


Fig. 26.2 The original idea of naming a K + NN structure

Among many different approaches for the control design, the first-time use of K + NN (K stands for gains) structure in this paper stems from the idea of 1+ fuzzy [1, 2]. 1+ fuzzy is an FIS connected in parallel with a direct-through link as shown in Fig. 26.1. Several investigations that compared the capability of an FIS-based control and that of an NN-based control [3, 4] have found that NN has more flexibility for use as a controller in practice due to its simple calculation of numerals rather than executing the complicated FIS algorithm [5–7]. First, as can be seen in Fig. 26.2, 26.1 can be replaced by a different gain constant such as K. This is more useful to split a quick signal from the error signal for NN use. SE, SDE, and SU are conventional scaling factors for the range scaling as used in a general FIS structure. Applications of K + NN structures are examined in [4–8]. From them, we can see that the K + NN structure is in reality almost a universal controller or assistor to other controllers. Furthermore, If K + NN is used for modifying the original plant structure such that the performance of the new plant is faster than the original one, the controller designed is much easier than before [25]. This is a good viewpoint.

In Sect. 2, the concept of soft computing is quickly reviewed, and the components used are introduced briefly. Plausible examples and simulation results are introduced in Sect. 3. Finally, a summary is provided as the conclusion in the final section.

26.2 Soft Computing and K + NN Structure

26.2.1 Soft Computing Components

Many algorithms for optimization have been used over the years. Error back propagation (EBP or BP in short) can be one of the algorithms used for finding the optimum of some fitness functions. The shortcoming with BP is the result can be trapped into a local optimum. GA is a better candidate for optimization solving. It involves evolution processes such as best-selection, crossover of the chromosomes, and the mutation process as well as the randomness nature used in the algorithm. Ant colony optimization and bird flocking of PSO [9–11] are alternatives for solving optimization with satisfactory achievements since it is difficult to find parameters for controllesr as well as K + NN structures in real-time mode. The second reason for using PSO for parameter finding is that BP can not be easily used to find weights for NN learning. Not surprisingly, difficulty with NN learning occurs frequently.

Two out-of-soft computing components are utilized as the cornerstone of the K + NN structure. The first is the simple structure of a 2-5-1 multi-layer neural network. The transfer function of the hidden layer is the saturated linear (satlins [12, 13]), while the output uses a pure linear (purelin). To simplify calculations used for NN implementation, only 5 neurons are used. Bias is not required here, whereas the best possible weights are listed in the following [5–7]:

$$\text{net.IW}\{1,1\} =$$
$$\begin{matrix} -0.239764000000000 & 0.079984000000000 \\ -0.014845000000000 & -0.095267000000000 \\ -0.027509000000000 & 0.025377000000000 \\ 0.015372000000000 & 0.011187000000000 \\ 0.026170000000000 & -0.026874000000000 \end{matrix}$$

$$\text{net.LW}\{2,1\} = [14, -5, 36, 194, -37]$$

The second component is the optimization algorithm of PSO, which is used for finding those parameters used both in K + NN and the conventional controller (predefined). The coefficients c1 and c2 used in the algorithm (see Eq. (26.1) and (26.2)) are set to 2 respectively.

$$\text{vel}[] = \text{vel}[] + \text{c1} * \text{rand}() * (\text{pbest}[] - \text{particle}[]) + \text{c2} * \text{rand}() * (\text{gbest}[] - \text{particle}[]) \tag{26.1}$$

$$\text{particle}[] = \text{particle}[] + \text{vel}[] \tag{26.2}$$

where

vel: velocity of the particle
particle: variables to be optimized
rand: uniform random number (0 to 1)

c1, c2: learning factors
pbest: the best particle record in a generation
gbest: the global best particle

26.2.2 K + NN Structure

The final structure used in this paper is the K + NN one shown in Fig. 26.3. K (or Ka) is in parallel with the NN, and the combination output is in series with a Kp or Kb. In general, the pre-defined controller of the original system comes after the Kb block.

The block diagram of a conventional control system may include some of three different kinds of controllers, e.g., the feedforward compensator or filter, the cascade controller in the loop, and the feedback controller in the loop as shown in Fig. 26.4.

The role of K + NN structure is indicated in the central part of Fig. 26.5, and, AK + NN structure is a controller to the plant is shown in Fig. 26.6.

If the cascade controller is not available, then the K + NN structure can be a temporary controller as shown in Ex. 1, Sect. 3. From this point of view, K + NN is almost like a universal controller.

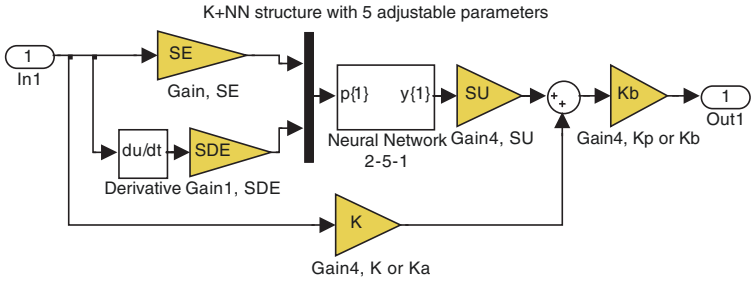


Fig. 26.3 The final K + NN structure with an additional gain Kb in series with K + NN

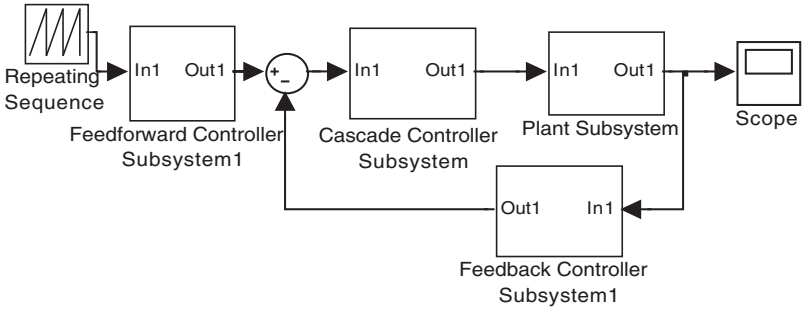


Fig. 26.4 A complicated controller consists of a cascade controller, a feedback controller and a feedforward controller

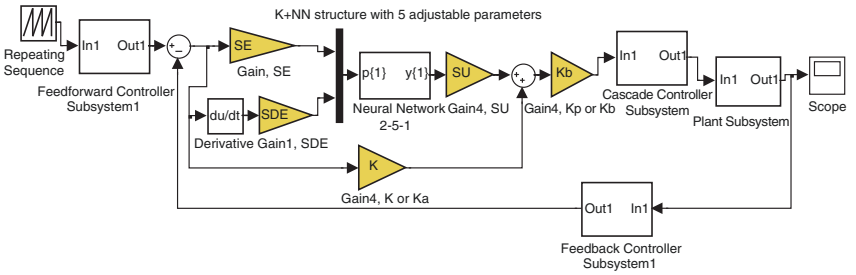


Fig. 26.5 A K + NN assistor is in cascade with the cascade controller of the above complicated controller

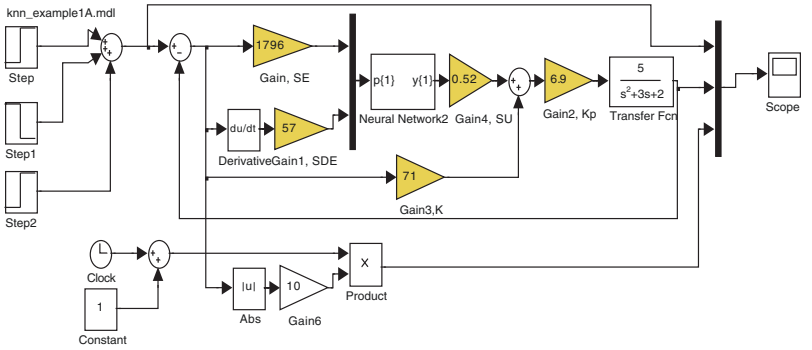


Fig. 26.6 A K + NN structure is a controller to the plant $5/(s^2 + 3s + 2)$

26.3 Applicable Examples and Simulations

In order to demonstrate that K + NN structures can be used to enhance the transient behavior of different tracking control systems, examples and simulation results are provided below:

Example 1: K + NN Only as a Universal Controller

The plant transfer function is given in Eq. (26.3).

$$\mathrm{Gp(s)} = \frac{5}{s^2 + 3s + 2} \tag{26.3}$$

Without limits of control efforts, these parameters are

$$\mathrm{Kp} = 6.9,\ \mathrm{Ka} = 71,\ \mathrm{SE} = 1796,\ \mathrm{SDE} = 57,\ \mathrm{SU} = 0.52.$$

The simulation result is shown in Fig. 26.7. The transient response is fast enough with good tracking response. The settling time is just over 0.1 second. The K + NN can be seen as a proportional, integral, and derivative (PID) implementation. PID compensators are widely used in industrial control systems because of their simplicity and robustness with regard to plant parameter variations. Since NN uses

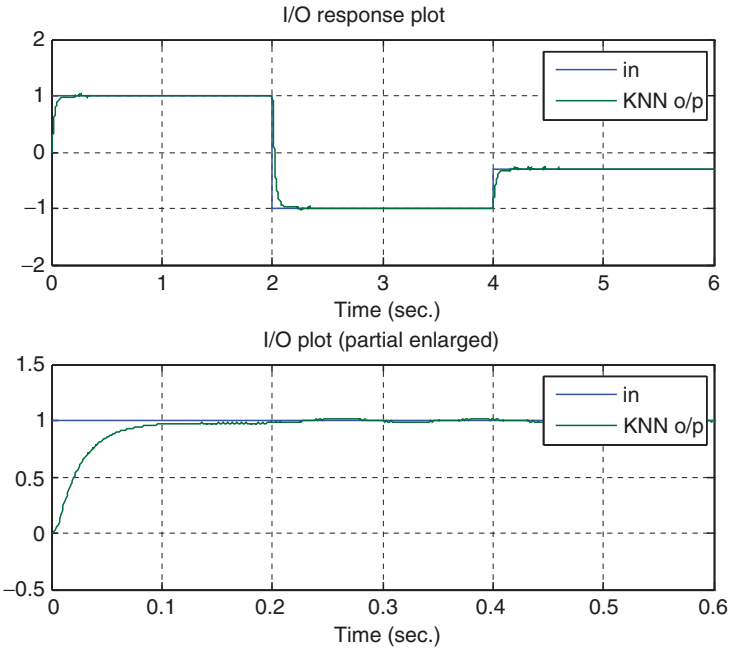


Fig. 26.7 Time responses of input and tracking output under the K + NN control

both the error signal and the derivative of the error signal, it can be modeled as a PD controller after a fashion. Also K + NN can do the function as well as a PID controller does.

Example 2: K + NN Only, with Limitations on Plant Input

With the same plant as before, the limit of the plant input is restricted to ±100. Parameters are found to be

$$\text{Kp} = 38.3,\ \text{Ka} = 155.9,\ \text{SE} = 6904,\ \text{SDE} = 267.84,\ \text{SU} = 1.266.$$

As seen in Fig. 26.8, appending a limiter ahead of the plant will slow down the original response of the system markedly (compared to Fig. 26.7). In this case, large parameter values are required to overcome the appended limitation of the applied control effort.

Example 3: Use of K + NN to Simulate a Given PID Controller with the Plant of Eq. (26.3)

The pre-designed PID controller for the plant has parameter values of:

$$\text{Kp} = 6539,\ \text{Ki} = 23.643,\ \text{Kd} = 55.74.$$

Using K + NN to simulate this PID controller, the adjusted parameters become

$$\text{Kp} = 628.52,\ \text{Ka} = 9.42,\ \text{SE} = 0.047558,\ \text{SDE} = 0.223155,\ \text{SU} = 0.068826.$$

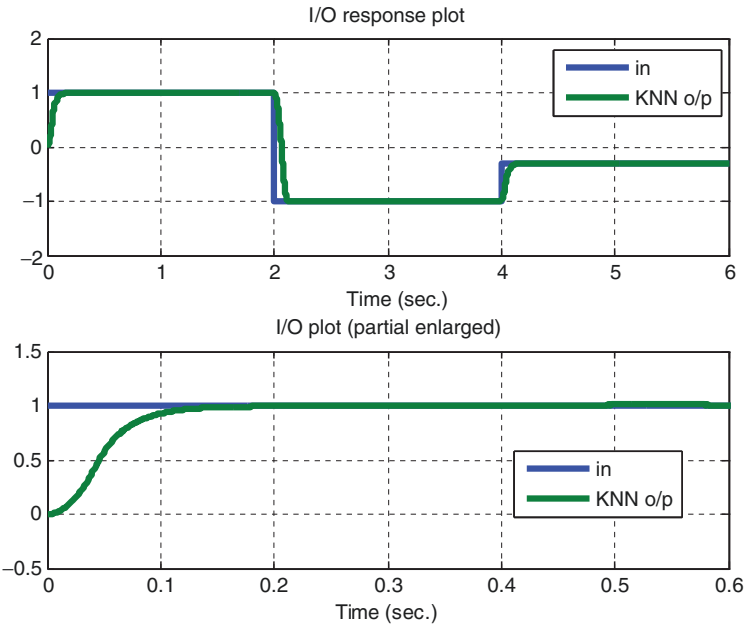


Fig. 26.8 Time responses of input and tracking output under the K + NN control with a limiter on the plant input

In this example, K + NN can be utilized as a PID controller as well Although they are different in structure, they are similar in function. K + NN animates the PID function with good results, as shown in Fig. 26.9.

Example 4: A Given PID Controller is Replaced by a K + NN in a Sluggish Plant (Eq. 26.4) [8]

$$\mathrm{Gp(s)} = \frac{18.5s^2 + 6.93s + 18.2}{s^5 + 17.47s^4 + 67.52s^3 + 64.86s^2 + 43.3s + 14.16} \tag{26.4}$$

With PID given parameters,

$$\mathrm{Kp} = 0.41,\ \mathrm{Ki} = 0.15,\ \mathrm{Kd} = 0.778.$$

The found parameters for K + NN are

$$\mathrm{Kp} = 0.903,\ \mathrm{Ki} = 0.3275,\ \mathrm{Kd} = 1.6784,\ \mathrm{K} = 0.472,$$
$$\mathrm{SE} = 2.1736,\ \mathrm{SDE} = 0.541,\ \mathrm{SU} = 0.002517.$$

Again, this example shows K + NN can simulate a given controller with different parameters used. They have almost the same response as shown in Fig. 26.10.

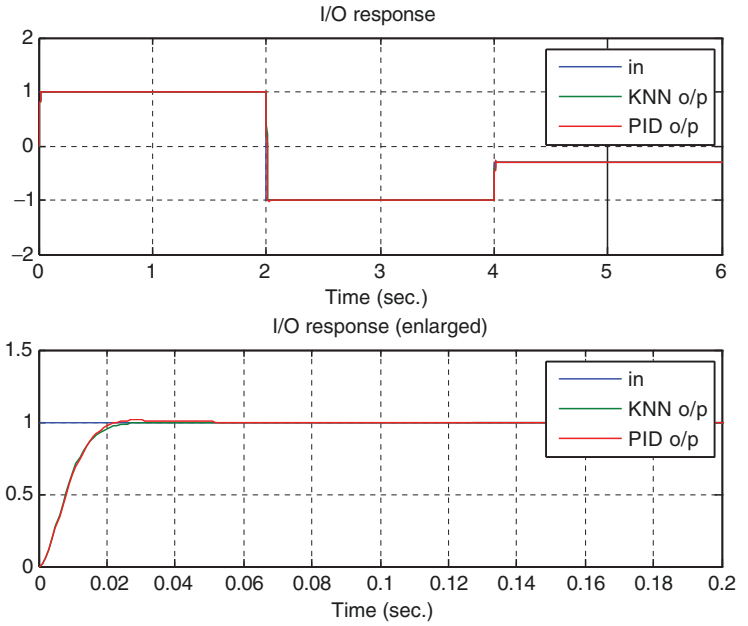


Fig. 26.9 Animate a K + NN to a given PID controller. Input/output response

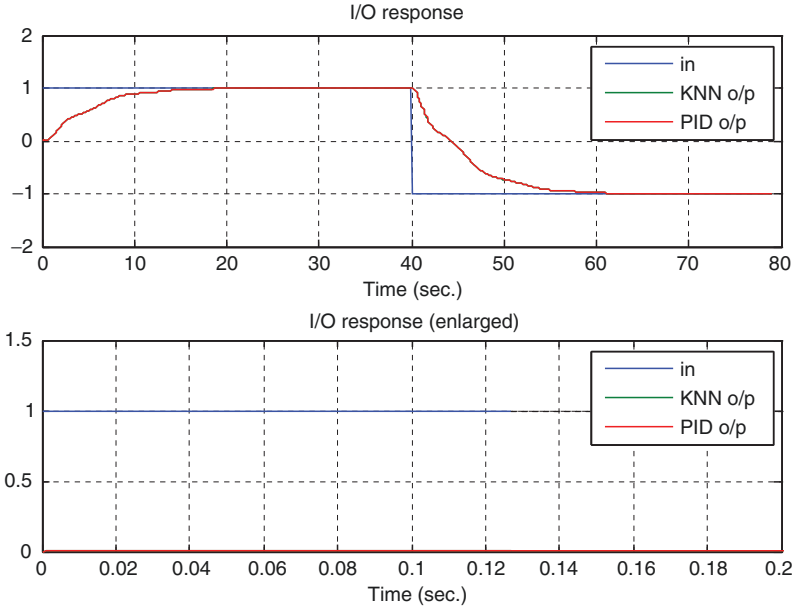


Fig. 26.10 K + NN perfomrs the same function as the PID

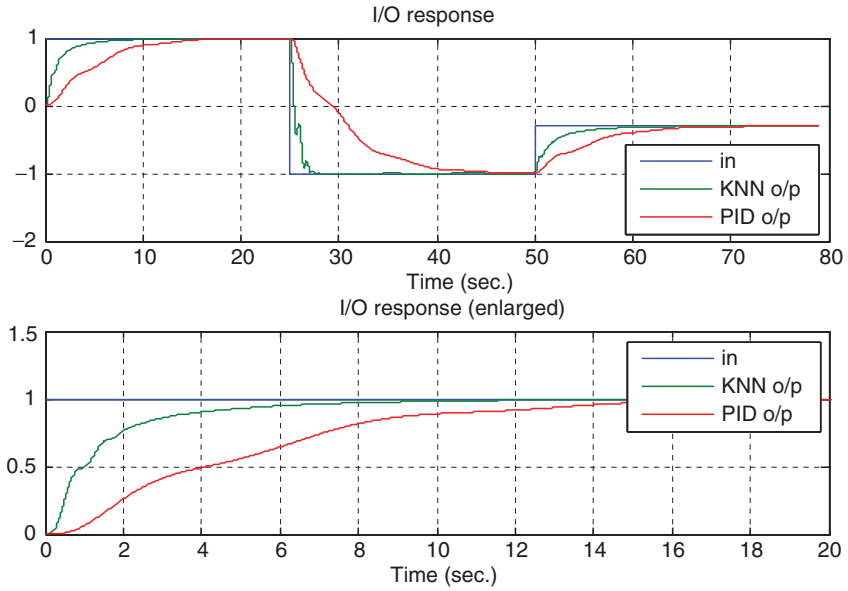


Fig. 26.11 Fast transient under K + NN control

Example 5: K + NN to Enhance the Transient of the Same Sluggish Plant Shown in Example 4

This time the capability of using K + NN to improve the transient of the original control system is apparent as shown in Fig. 26.11. The reading of parameters is

$$\text{Kp} = 2.172,\ \text{Ki} = 4.97,\ \text{Kd} = 1.874,\ \text{K} = 7.877, \text{SE} = 28.42,$$
$$\text{SDE} = 74.05,\ \text{SU} = 0.0432.$$

It is very difficult for us to find a better PID to reduce the transient time. In general, a more complicated compensator is required to improve the transient further. However, use of a K + NN structure is really a good choice to do the job. Of course, nonlinearity of the K + NN assistor is obviously seen in the figure. No blame on it.

Example 6: K + NN to Improve a Fuzzy-Logic Controller with the Plant Transfer Function of Eq. (26.5)

$$\text{Gp(s)} = \frac{1}{s^3 + 3s^2 + 3s + 1} \tag{26.5}$$

The block diagram for the simulation is shown in Fig. 26.12. Discussion of the fuzzy-logic controller (FLC) [14] is omitted here for simplicity. In the block diagram, the K + NN structure consists of Ka, Kb (or Kp), SE, SDE, SU as shown in the figure. Those parameters have the following values:

$$\text{Kp} = 0.267543,\ \text{Ka} = 0.834,\ \text{SE} = -3.4,\ \text{SDE} = 2.405,$$
$$\text{SU} = 0.6841.$$

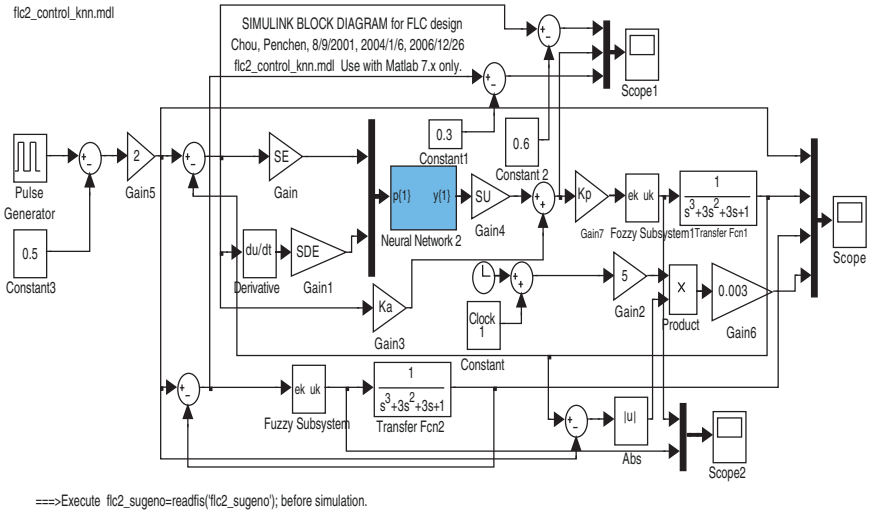


Fig. 26.12 A K + NN is inserted (upper part) before the fuzzy subsystem

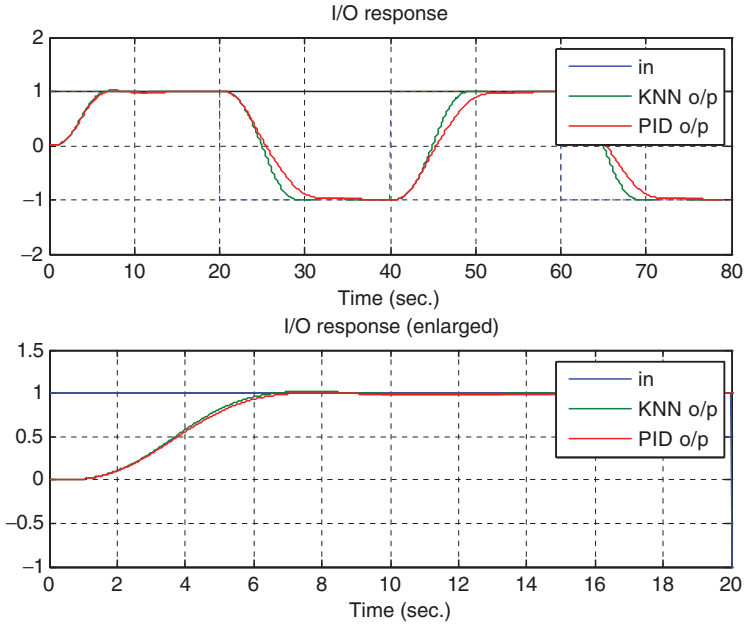


Fig. 26.13 Further transient improvement on the output under K + NN assistance

The improvement in transient in this case is not as obvious as before. This can be deduced from Fig. 26.13. However, K + NN self-proves that it works for transient improvements. More trials on the search of parameters becomes necessary.

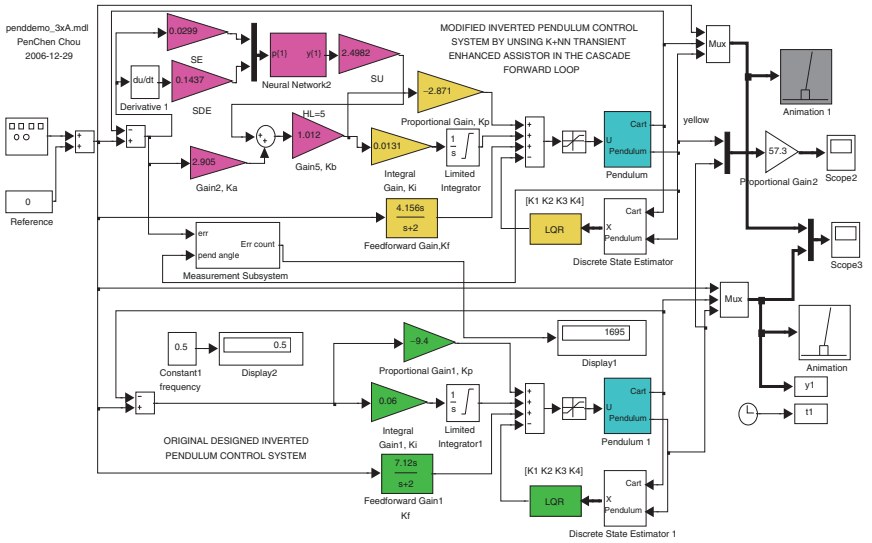


Fig. 26.14 K + NN is added to a complicated inverted pendulum controller (upper part)

Example 7: K + NN to Improve the Transient of an Inverted Pendulum System with a Composition of Different Controllers

The Simulink diagram for this problem [13, 15–24] is shown in Fig. 26.12. Parameters for the K + NN structure are

$$\text{Kp} = 0.267543,\ \text{Ka} = 0.834,\ \text{SE} = -3.4,\ \text{SDE} = 2.405,$$
$$\text{SU} = 0.6841.$$

It is interesting to note that in order to control the pendulum in the vertical angle without falling down, and to move from position to position in a short period of time, PI, feedforward filter and state-feedback control are all used together to overcome the tendency of the pendulum to fall if it is not in a good control mode. Thus the original controller is a composition of several different control forms. The transient can be further improved by using the K + NN structure. In the long run, as we see in Fig. 26.15, a better transient with quicker response to position changes accordingly reduces steady-state error. This is a real challenge. By decling the pendulum angle up to 50 degrees, the controller moves at top speed to prevent the pendulum from falling.

Example 8: K + NN to Improve Transient of a PID Control System with A Slow-Varying Plant of Eq. (26.6)

The transfer function of the plant [8] is

$$\text{Gp(s)} = \frac{25.2s^2 + 21.2s + 3}{s^5 + 16.58s^4 + 25.41\text{s}^3 + 17.18s^2 + 11.7s + 1} \tag{26.6}$$

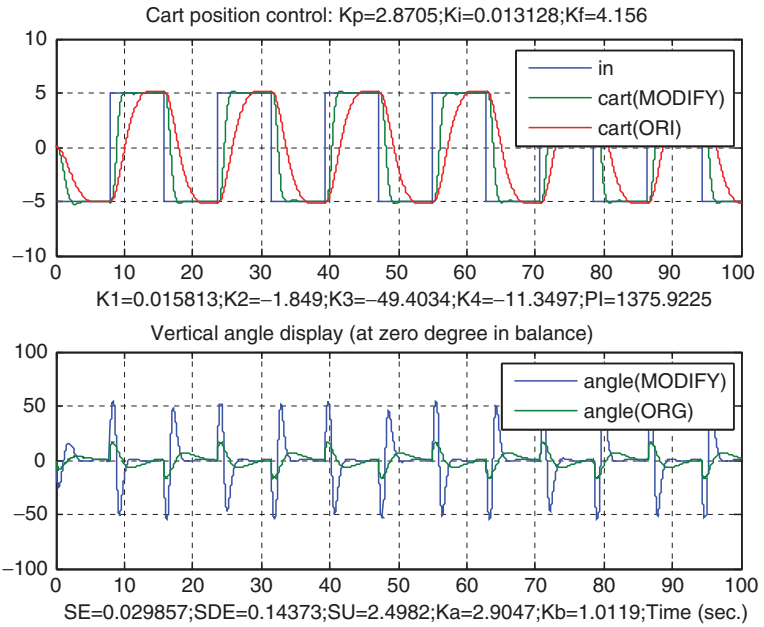


Fig. 26.15 Fast transient and good tracking on position control with the aid of a K + NN

The original designed PID has parameters of

$$\text{Kp} = 0.4258,\ \text{Ki} = 0.3281,\ \text{Kd} = 0.3333.$$

Using K + NN for this system, the parameters (shown in yellow blocks) are found to be

$$\text{Kp} = 7.888,\ \text{Ki} = 0.396,\ \text{Kd} = 0.265,\ \text{Ka} = 5.602,$$
$$\text{SE} = 1.0813,\ \text{SDE} = 1.1832,\ \text{SU} = 0.109,\ \text{Kb} = 5.407.$$

Figure 26.14 shows the connection of K + NN in the original control system. The plant is a tough one in general for the control engineers designing its controller. K + NN is added (in the upper part) to the original PID controller for a slow varying plant in Fig. 26.16. With maximum effort, one PID (data shown above) still has a sluggish tracking response, as shown in Fig. 26.17. However, with the incorporation of a K + NN universal assistor, the response is fascinating. This shows the magic of using K + NN as the assistor to the original controller. It is believed that K + NN can be used as the partner of a conventional controller as an add-in option.

Example 9: One K + NN to Improve the Original Plant and another K + NN to Improve the Controller

The transfer function of the plant [25] is

$$\text{Gp(s)} = \frac{5}{s^2 + 3s + 2} \tag{26.7}$$

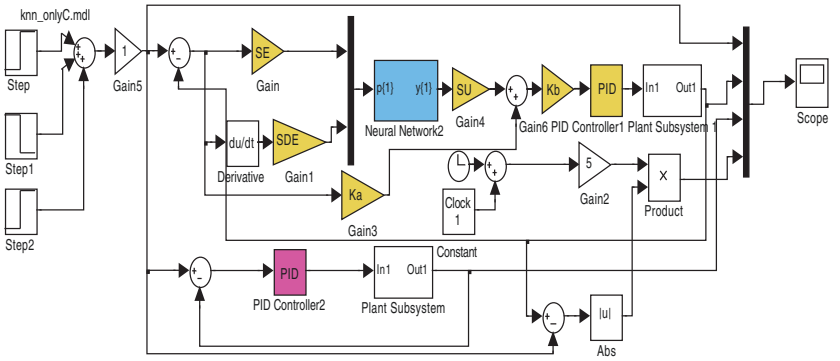


Fig. 26.16 K + NN is added (in the upper part) to the original PID controller for a slow-varying plant

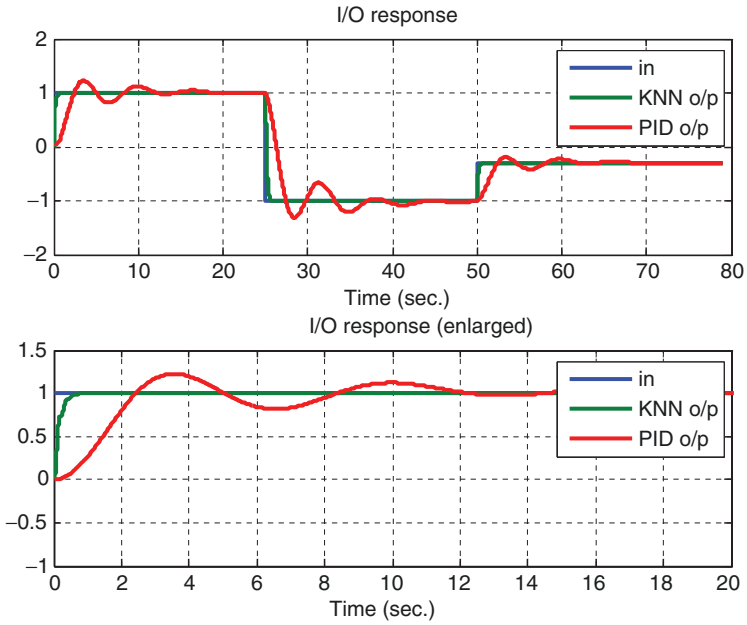


Fig. 26.17 K + NN can improve the transient further in the output response

With the K + NN assistor to the plant in Eq. (26.7) with adjustable parameters (Ka1, Kb1, SE1, SDE1, and SU1), another K + NN assistor to the PID controller2 is appended as shown in Fig. 26.18. A KNN-modified plant subsystem is drawn in Fig. 26.19. There are Kp, Ki, Kd, Ka, Kb, SE, SDE, SU parameters for the control part, and Ka1, Kb1, SE1, SDE1, SU1 in the plant modification. All 13 parameters are tuned by PSO to be

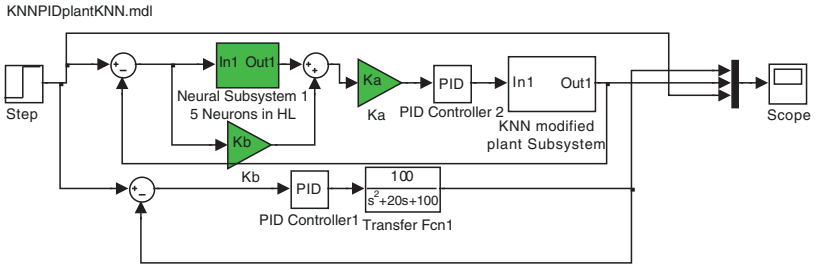


Fig. 26.18 K + NN + PID control to the K + NN modified plant. Kp, Ki, Kd are used in PID controller2 and Ka, Kb, SE, SDE, and SU are used in this block. (SE, SDE, SU are in Neural Subsystem1)

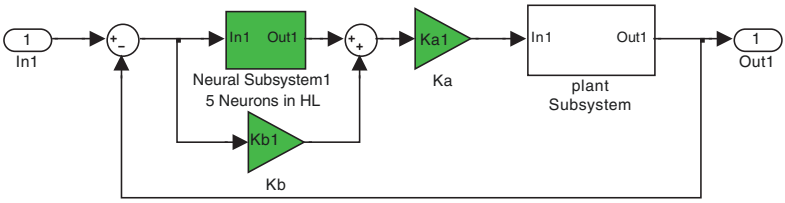


Fig. 26.19 KNN modified plant subsystem is expanded here. Ka1, Kb1, SE1, SDE1, SU1 are used in this block diagram

```
para (1)=        707.024224;  % Kp
para (2)=        864.392268;  % Ki
para (3)=          0.079051;  % Kd
para (4)=          1.354039;  % Ka
para (5)=         10.930904;  % Kb
para (6)=          0.626722;  % SE
para (7)=          0.139216;  % SDE
para (8)=          0.147756;  % SU
para (9)=          0.403021;  % Ka1
para(10)=          2.759857;  % Kb1
para(11)=          1.783194;  % SE1
para(12)=          1.509223;  % SDE1
para(13)=          0.003021;  % SU1
```

The corresponding I/O plot for the system in Fig. 26.18 is shown in Fig. 26.20. The reference model is based on the plant transfer function of

$$G_{ref}(s) = \frac{100}{s^2 + 20s + 100} \tag{26.8}$$

Values for the reference PID controller are 13.61 for Kp, 30.12 for Ki, and 0.3445 for Kd. The performance for the reference model has a fast settling time and a small overshoot as shown in Fig. 26.20. However, with 2 K + NN structures involved for the same control system, 13 parameters are used instead of 3 PIG gains mentioned above, Better performance can be obtained compared to the reference model, that is, shorter settling time and no overshoot.

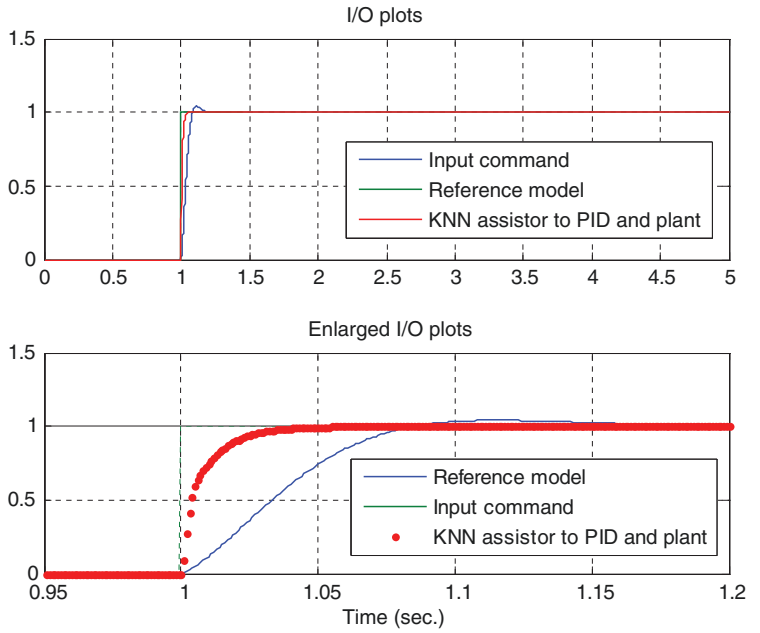


Fig. 26.20 I/O response of K + NN + PID control to the K + NN modified plant

26.4 Conclusion and Future Work

Using soft computing techniques such as NN and PSO to find the necessary parameters in advance (off-line style) before the actual control simulation, it is shown that further improvement on the transient as well as the steady-state error become possible. The basic K + NN structure consists of a given constant gain K (or Ka in some examples) in parallel with a simple NN with three appropriate scaling factors, SE, SDE, SU. Kb is followed at the composition of K and NN. The block diagram is shown in Fig. 26.3. If there is no compensator available for a plant, the simple gain Kb at the end of K + NN structure can be treated as the proportional control. If there exists a given controller after the error signal terminal for a tracking system, use K + NN in cascade with the given controller. K + NN is inserted then between the error signal and the predefined controller. In this case, Kb can be an option. Furthermore, it is better to readjust accordingly those parameters used in the given controller besides K, Kb, SE, SDE, SU (the later 3 parameters are scaling factors for NN) parameter-finding.

From the previous examples, it is shown that K + NN can be either a stand-alone controller or an assistor to a well and pre-defined controller for improving transient/steady-state. Further more, K + NN can be used to modify the plant for easy later controller design. Therefore, the K + NN structure is highly applicable for use in many control systems in the future.

How to investigate for the behavior of K + NN compared to the usual conventional linear/nonlinear controller is one topic to be studied in the future. More applicable examples on different control types must be observed as well.

References

1. Hung LC, Liu YC, Chung HY (2002) An associate design of fuzzy logic with grey-neural prediction in PID controller. In: 2002-ROC Automatic Control Symposium, Taipei, Taiwan
2. Lai LY, Lee MY (1993) Fuzzy tuning of integrator outputs of PID controllers for a dc motor system. In: Chung-Yuan J, Vol. XXII: 126–137
3. Basilio JC, Matos SR (2002) Design of PI and PID controllers with transient performance specifications. In: IEEE Transactions on Education 4(45):364–370
4. Chou PC (2003) Design of a robust neural controller for a specified plant using genetic algorithms approach. In: IEEE International symposium on computational in telligence for measurement systems and applications, pp 233–235
5. Chou PC, Dzen AT, Hsieh SC (2005) Neural assisted PID controllers design for control systems. In: Proc. of the Fourth IASTED International Conference Computational Intelligence, July 4–6, Calgary, Alberta, Canada, pp 199–204
6. Chou PC, Hsieh SC (2005) Neural assisted PI/PID controller design for a motor control system. In: IEEE CIMSA2005, July 20–22, Sicily, Italy
7. Chou PC, Chang TC, Hwang TC (2006) The experience of using a neural assistor to enhance the transient characteristics of well-defined control systems. In: IEEE-CIMSA 2006, Spain
8. Chou PC, Huwang JJ (2004) Design of PID controllers using genetic algorithms approach for low-damping, slow-varying plants. In: IEEE International symposium on neural networks (ISNN 2004), Dalian, China
9. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: From natural to artificial systems. In: ISBN 0-19-513159-2
10. Kennedy J, Eberhart RC (2004) Swarm intelligence. In: ISBN 1-55860-595-9
11. Parsopoulos KE, Vrahatis MN (2002) Recent approaches to global optimization problems through particle swarm optimization. In: Natural Computing 1 (2–3):235–306
12. MathWorks, inc. (2002) Matlab User's Guide
13. MathWorks, inc. (2002) Simulink User's Guide
14. Chou PC (2005) Genetic Algorithms, Principles and Applications. ChuanHwa book Company (In Chinese)
15. Kumar M, Garg D (2004) Intelligent learning of fuzzy logic controllers via neural networks and genetic algorithm. In: Proceedings of 2004 JUSFA
16. Kawada K, Yamamoto T, Mada Y (2003) A design of evolutionary recurrent neural-net based controllers for inverted pendulum, vol. 3, pp 1419–1422
17. Kawamoto S (2002) Nonlinear control and rigorous stability analysis based on fuzzy system for inverted pendulum. Osaka Prefecture University
18. Lin Y, Qian J, Xue A, Wang J (2002) Simple multi-pd control algorithm of double inverted pendulum. In: Proc. of IEEE TENCON'02
19. Omatsu S, Deris S (1996) Stabilization of inverted pendulum by the genetic algorithm. In: IEEE-1996
20. Callinan T, (2003) Artificial neural network identification and control of the inverted pendulum, School of Electronic Engineering, Dublin City University, Dublin
21. Haupt RL, Haupt SE (2004) Practical genetic algorithms, 2nd edition. In: Hoboken, New Jersey: Wiley-Interscience
22. Huang TC, Chou PC (2004) The study of PID controllers design to slow-varying, low-damping plants. In: ROC Automatic Control, Changhua, Taiwan 2004

23. Lee TW, Chou PC, Huang CT (2004) Modern control technique on position and pendulum control of an inverted pendulum system. In: ROC Automatic Control, Changhua, Taiwan 2004
24. Lee TW, Chou PC (2004) Use of genetic algorithms to improve fuzzy logic controller. In: ROC Automatic Control, Changhua, Taiwan 2004
25. Chou PC, ET Tsai (2007) Smart assistor for controllers and plants in control systems. In: Submitted to ASM 2007 Conference

Chapter 27
Design and Implementation of UHF Antennas for UAV Radio Link

Firouzeh ZH, Safari-Hajat-Aghaei A, and Mirmohammad-Sadeghi H

27.1 Introduction

Unmanned Aerial Vehicle (UAV) is a remotely piloted or self-piloted aircraft that can carry cameras, sensors, communication equipment or other payloads. They commonly are used in military operations and scientific research, where their powerful intelligence collection capability greatly reduces human risk and operation cost. Their autonomous flight and powerful sensing capabilities further facilitate their operation beyond the visual range. UAVs can be controlled by *Ground Station (GS)* or *Satellite Station (SS)* to perform their missions [9, 16].

GS antennas should have appropriate frequency bandwidth, polarization, gain, and radiation patterns on the basis of radio link specifications. It is necessary to use directional antennas to cover long ranges because of their high gain. The antenna polarization of the UAV is vertical, and the direction of its installed antenna most often changes due to maneuvering rapidly around the GS; therefore, it is necessary to choose circular or slant $(+45/-45)$ polarization for GS antennas. Meanwhile, it uses spread spectrum methods such as frequency hopping to increase link security, so the antennas should be relatively wideband. Since the desired system is based on Ground Control Station (GCS), i.e., it can find the location of the UAV by Global Positioning System (GPS) or received RF signals; thus, it can direct the antenna to the UAV and track. However, the fast maneuvering of the UAV around the GS, especially during takeoff and landing should be regarded; consequently, GCS cannot follow the UAV simultaneously and quickly. Moreover, the UAV can be controlled visually, so an omni-directional antenna can be utilized in the short ranges—less than 5 km.

The GS monopole antenna, which is a vertical-polarized omni-directional antenna, is designed, simulated by a Fidelity simulator, and optimized. The *Helix*

Firouzeh ZH, Safari-Hajat-Aghaei A, and Mirmohammad-Sadeghi H
Information and Communication Technology Institute (ICTI), Isfahan University of Technology (IUT), Isfahan, Iran

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

antenna is designed and matched to 50Ω in frequency bandwidth of 100 MHz by a new stub-loaded matching method. The dimensions and position of optimum stub matching is obtained by an IE3D simulator based on Method of Moment. Also, helix antenna specifications such as radiation pattern, gain, HPBW, VSWR, and axial ratio (AR) are calculated. The implemented antennas will be tested and measured. In the end, the appropriate height for installing the antennas will be proposed in regard to LOS (line-of-sight) relations and multipath problems to cover the maximum range of 100 km.

In addition, UAV antenna type, number of antennas, and location of the antennas on the UAV are very important for the efficient range. The influence of the UAV metallic body on the radiation pattern of antenna and multipath fading cause some degradation of the radio link performance. Firstly, appropriate locations will be recommended for UAV antenna installation, and necessary simulations will be performed by Fidelity software based on the FDTD method. After finding optimum antenna locations and measuring the received power at the GS, multipath effect and UAV body will be considered in order to select the proper antenna toward or away from the GS. *Switched antenna diversity* is a simple and appropriate method to establish a reliable radio link between UAV and GS.

27.2 Design and Simulation of GS Antennas

27.2.1 The Monopole Antenna

It is necessary to design a quite wideband, vertical-polarized, omni-directional antenna that covers 360° around the GS at the maximum range of 5 km. A *thick-rod monopole antenna* with bent ground plane has the desired characteristics. The dimensions have been shown in Fig. 27.1. The $\lambda/4$ monopole antenna is simulated at the center frequency of 830 MHz by Fidelity simulator.

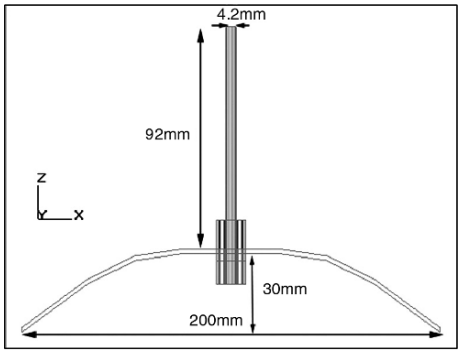


Fig. 27.1 The $\lambda/4$ thick-rod monopole antenna with bent ground plane

27.2.2 Helix

Helix antenna has many applications to establish any radio links between UAV and GS. Circularly polarized axial-mode helical antenna prepares the GS antenna to be able to receive signals in the different maneuvering conditions and to solve the wave propagation problems. The UAV has a vertical-polarized antenna, so the helix antenna will receive the signals with a loss of 3dB in all directions [2, 12].

Geometry of a helix has been depicted in Fig. 27.4. Helix radiates in two modes—normal and axial—on base of the ratio of circumference to wavelength. The axial mode has the maximum radiation at the direction of antenna axis. When the circumference of the helix is about one wavelength, the paraxial radiation is circularly polarized. Firstly, Kraus introduced a few relations to calculate gain and HPBW of the helix, but the expected values were more than the real ones [12]. In 1980, King and Wong recommended Eq. (27.1) to compute the gain according to the measurement [10].

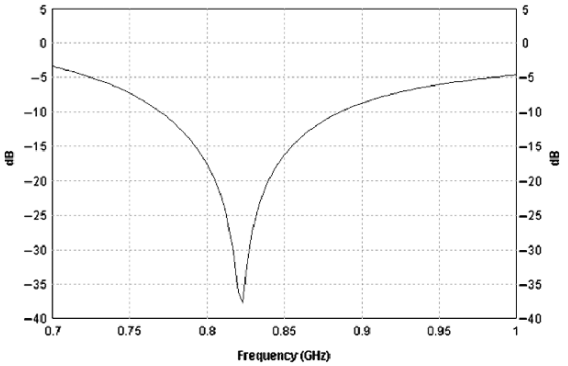


Fig. 27.2 Simulated VSWR of the monopole antenna

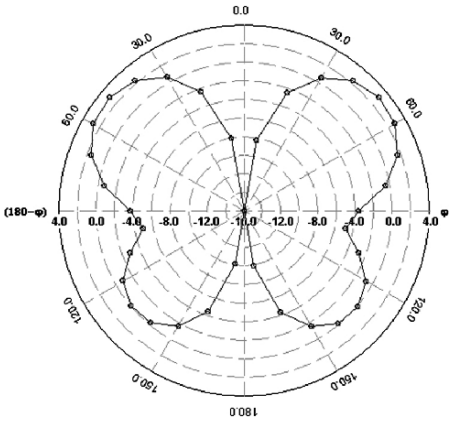


Fig. 27.3 Radiation pattern of the monopole antenna

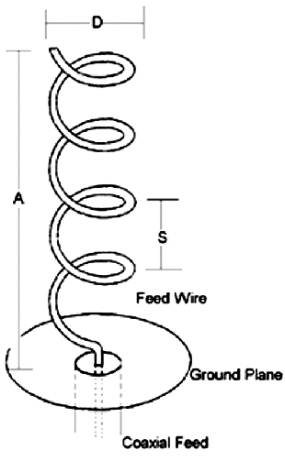


Fig. 27.4 Geometry of a helix

$$G_P(dB) = 8.3\left(\frac{\pi D}{\lambda_P}\right)^{\sqrt{N+2}-1}\left(\frac{NS}{\lambda_P}\right)^{0.8}\left(\frac{\tan 12.5^o}{\tan\alpha}\right)^{\sqrt{N}/2} \tag{27.1}$$

The parameters are N: number of turns, S: turn spacing, D: diameter of helix circumference, α: the pitch, and λ_P: the wavelength corresponding to the maximum gain. In 1995, Emerson proposed Eq. (27.2) to predict the maximum gain for axial mode on basis of numerical modeling [3].

$$G_{\max}(dB) = 10.25 + 1.22\bar{A} - 0.0726\overrightarrow{A}^2 \tag{27.2}$$

$$\bar{A} = {}^{A}/_{\lambda}$$

$\bar{A}$ is the normalized antenna length to the wavelength. It has the good accuracy to predict the gain, hence; it will be used to design the helix. It is required to design the helix as a GS antenna at the center frequency of 830 MHz for the radio link. It should have VSWR < 2 (50Ω system) and AR $<$ 2dB in a frequency bandwidth of 100 MHz. LOS link calculation shows that a GS antenna with the minimum gain of 12dB throughout the bandwidth is required to cover the 100 km range. According to [3, 12] references, the number of turns and the dimensions of the desired helix antenna are as follows:

$$N = 13,\ S = 9.0\,cm, \quad D = 12.6\,cm \tag{27.3}$$

Kraus's relations predict the gain of 16.5dB and the HPBW of 29°, but King & Wong's relations expect the gain of 14.4dB and the HPBW of 37° [10, 12]. It is calculated 13.4dB gain based on Emerson's formulas [2].

The helical conductor diameter does not significantly affect the radiation properties of the helix and may vary from 0.006λ and 0.05λ. The diameter conductor is presumed 2.7 mm. The helix may be supported by a few radial insulators mounted

on an axial dielectric or tube whose diameter is a few hundredths of wavelength, by one or more longitudinal dielectric rods mounted peripherally or by a thin-wall dielectric tube on which the helix is wound. We use a thin-wall plastic cylinder to support the helix. The ground plane may be flat (either circular or square) or cup-shaped, forming a shallow cavity, or replaced by loops. The diameter or side dimension of the ground plane is calculated at 0.5λ at the low frequency of the bandwidth. If the diameter of the ground plane is lengthened, the level of the side lobes will be increased. We use a flat 320 mm-diameter circular ground plane for the helix. The antenna parameters of the helix are not sensitive to the calculated dimensions, and it works well in a wide bandwidth [4].

The input impedance of an axial-mode helical antenna is between 140Ω and 150Ω. With a suitable matching section, the terminal impedance can be made any desired value from less than 50Ω to more than 150Ω. Thus, by bringing the last $1/4$ turn of the helix parallel to the ground plane in a gradual manner, a tapered transition between the 140- or 150-Ω helix impedance and a 50Ω coaxial line can be readily accomplished. This can be done with either axially or peripherally fed helices but is more convenient with a peripheral feed [11, 18]. A new *stub-loaded* matching is proposed for a helix in the frequency bandwidth of 100 MHz. This method is easy, flexible, tunable and low-loss. Figure 27.5 shows the feed rod is passed through the circular stub. The diameter and thickness of the copper-circular stub are 50 mm and 1 mm, respectively. The position of stub is changeable and tunable. Meanwhile, the input impedance of the helix is dependent on the thickness of the feed rod and the angle of the first loop relating to the ground plane. The plastic-cylinder structure is connected to the ground plane by Teflon flange.

The designed helix is simulated by IE3D software to optimize the dimensions of the helix and the place of the stub. The geometry of the helix is depicted for simulation in Fig. 27.6. Simulation results show that the helix without stub-matching has the directivity of 12.8dB and VSWR of 2.5; consequently, the antenna efficiency and gain will be 80 percent and 11.8dB, respectively. After simulation of the helix with a 50 mm-diameter circular stub, the most favorable height of the stub is obtained 30 mm above the ground plane. In addition, the angle of the first loop relation to the ground plane is 45°. Figures 27.7 and 27.8 illustrate far-field radiation patterns in an elevation plane $\varphi = 0°$ (XOZ) and azimuth plane $\varphi = 90°$ (YOZ) at a frequency of 830 MHz, correspondingly. Both of the radiation patterns are similar; therefore, elevation and azimuth HPBWs are equal to 31°. The gain is computed at 12.7dB. Figure 27.9 shows the antenna gain from 11.8dB at 780 MHz has been increased to 13.8dB at 880 MHz. As a result, the beamwidth is increased at low frequencies and

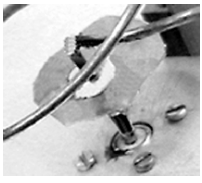


Fig. 27.5 The position of the circular stub

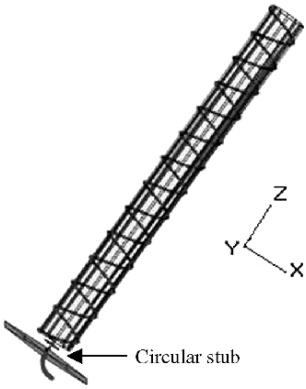


Fig. 27.6 Helix geometry simulated by IE3D software

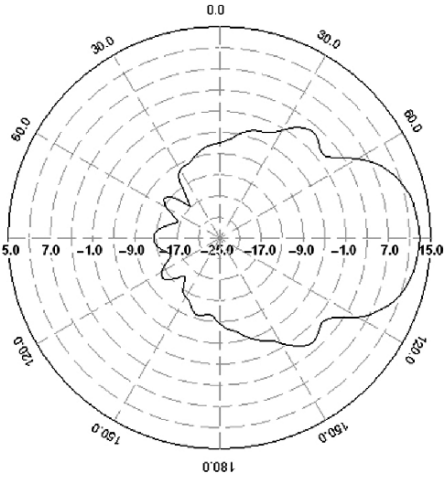


Fig. 27.7 Elevation-plane radiation pattern at 830 MHz

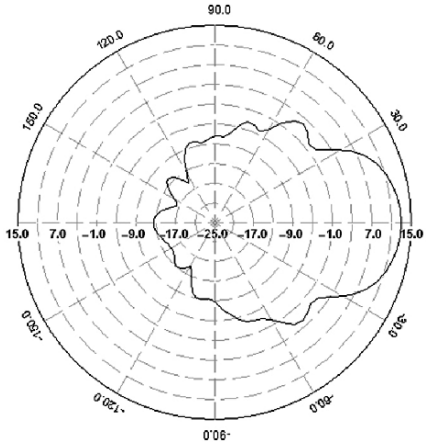


Fig. 27.8 Azimuth-plane radiation pattern at 830 MHz

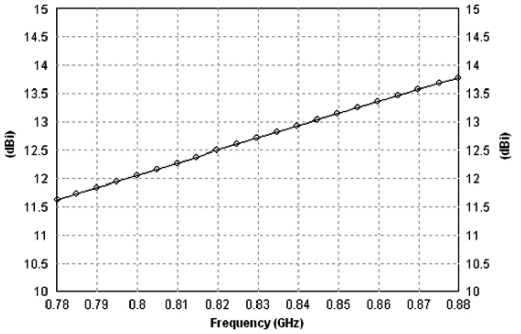


Fig. 27.9 Helix gain in frequency bandwidth of 100 MHz

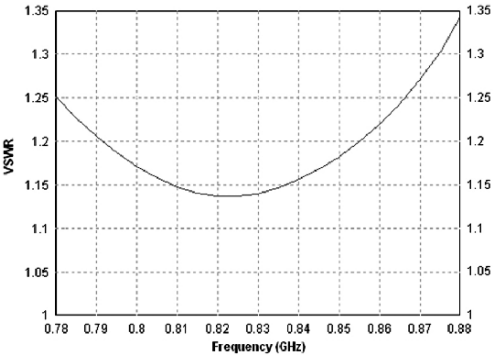


Fig. 27.10 Helix VSWR in frequency bandwidth of 100 MHz

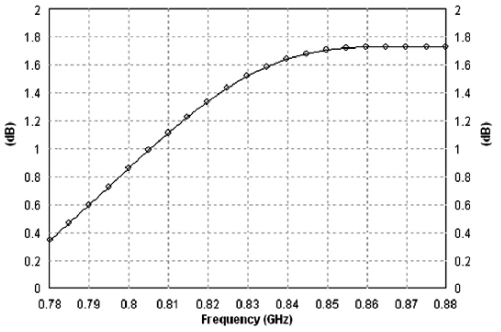


Fig. 27.11 Axial Ratio in the direction of helix axis

vice versa. As shown in Fig. 27.10, VSWR is 1.14 at 830 MHz and less than 1.35 throughout the frequency bandwidth. It is found that the circular stub can realize a very good matching with a loss of around 0.15dB. The axial ratio is the values simulated in the direction of the helix axis. Its depicted AR is less than 1.8dB as shown in Fig. 27.11. The loss of plastic-cylinder structure is expected about 0.1dB.

Pattern, polarization and impedance characteristics represent remarkably good performance over a wide frequency range for a circularly polarized beam antenna.

27.2.3 Implementation and Measurement results

After manufacture of a monopole antenna as shown in Fig. 27.12, VSWR and radiation patterns were measured. Figure 27.13 shows that the monopole antenna has a $VSWR \cong 1.13$ at 830 MHz. The monopole antenna is omni-directional in the azimuth plane. According to Fig. 27.14, the measured elevation radiation pattern is presented at 830 MHz. HPBW of the vertical plane is nearly 65°. The gain of the monopole antenna is 2.5dB at an elevation angle of 40°, and the loss is 2dB in the horizontal plane.

The helix was implemented according to the optimized dimensions as shown in Fig. 27.15. The angle of the first loop to connect to the feed rod relation to the ground plane is 45°, which is shown in Fig. 27.16. VSWR was measured by a network analyzer. Figure 27.17 indicates VSWR < 2 in a bandwidth of 100 MHz, and VSWR is 1.6 at 830 MHz. The radiation pattern of the helix is endfire with a gain of 12.5 dB in the direction of helix axis. HPBW in both the elevation and azimuth planes are quite 32°. It is found that there is good agreement between Emerson's relations and measurement. The antennas have mechanical resistance against vibration and wind.

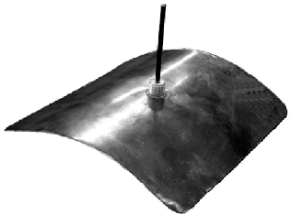


Fig. 27.12 The $\lambda/4$ thick-rod monopole antenna

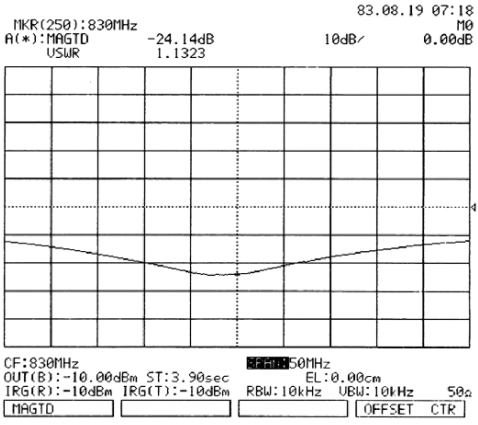


Fig. 27.13 VSWR of the monopole antenna

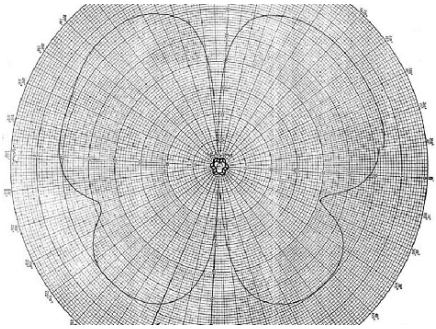


Fig. 27.14 Elevation-plane radiation pattern

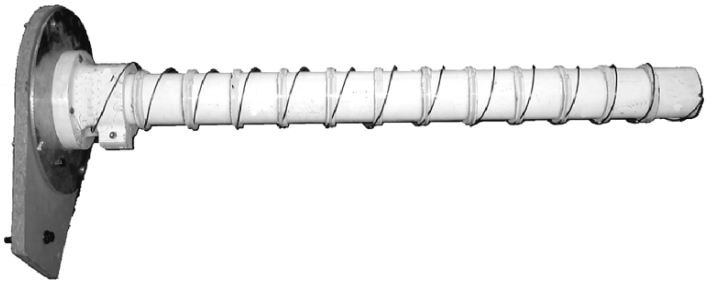


Fig. 27.15 Helix antenna

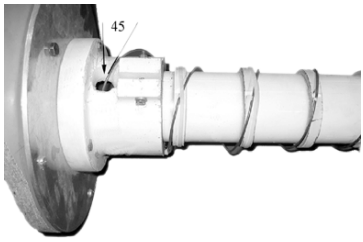


Fig. 27.16 Angle of 45° to connect the first loop to the feed rod

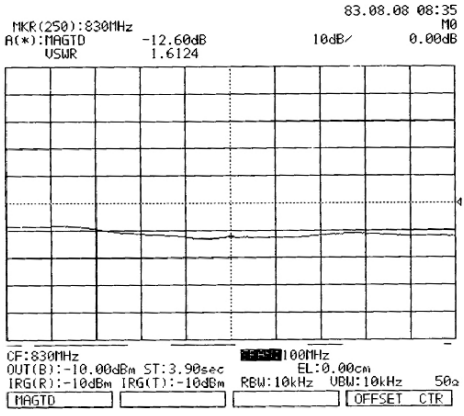


Fig. 27.17 VSWR of helix antenna

27.3 UAV Specifications and Antenna Design Procedure

We would like to design a quite wideband vertical-polarized omni-directional UAV antenna that covers 360° around the UAV at a maximum range of 100 km. Also, it should reduce the multipath effect and have minimum loss at the horizon. Figure 27.18 shows the geometry of the UAV.

- The length of the UAV nose: L1 = 0.6 m
- The length of the UAV body: L2 = 3 m
- The diameter of the UAV body: D = 0.3 m
- The length of short the UAV wings: W1 = 1.1 m
- The length of long UAV wings: W1 = 2.5 m
- The length of the UAV fin: W3 = 1.2 m

The UAV body is metallic but the nose is composite. Because of the metallic UAV body, a single antenna cannot cover 360° around the UAV. For instance, if the antenna is installed on the UAV nose, a deep null will be found in the rear of UAV radiation pattern. Many flight tests have been done to find the best locations for the antenna installation on the UAV. Consequently, two antennas are necessary to cover 360° around the UAV. One antenna is installed on the nose and the other one is put on the fin to cover the fore and the rear half-space, respectively. Both antennas are vertical-polarized omni-directional, which should have a $VSWR < 2$ at the frequency of 830 MHz.

After simulation and several tests, a $\lambda/4$ tapered-blade monopole antenna with a small ground plane on the fin and a $\lambda/4$ thick-rod monopole antenna with a large bent ground plane on the nose were used. Both antennas are aerodynamically conformal to the UAV body to reduce wind resistance. The *nose antenna* is set up near the metallic body; consequently, the reflections from the UAV body influence radiation

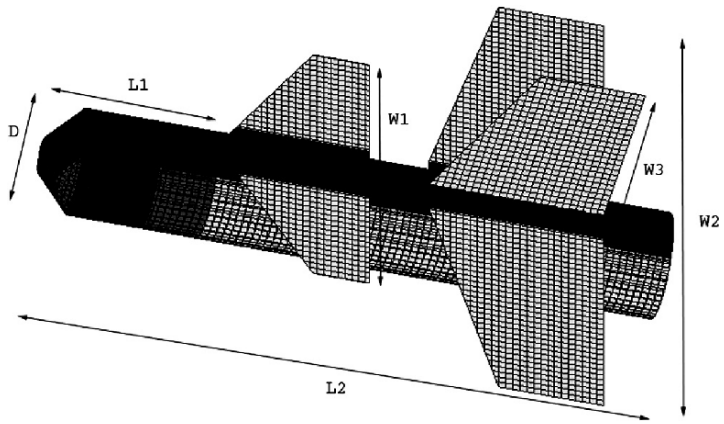


Fig. 27.18 The geometry of the UAV

performance. As a result, it is necessary to choose a large ground plane to decrease the induced current until the antenna has stable impedance characteristics [1]. Meanwhile, the distance between the nose antenna and the metallic body is essential because the radiation pattern changes intensely and asymmetrically when the nose antenna is mounted near the body. According to the simulations, measurements and the nose dimensions, the nose antenna should be set up at a distance of $^3/_4\lambda$ from the metallic circular disk. In this case, the loss is very low at the horizon.

27.3.1 Simulation of the UAV Antennas

Both of the antennas have been designed at 830 MHz and have been implemented on the basis of Figs. 27.12, 27.19. Simulation and measurement result in both antennas having the omni-directional radiation pattern in the azimuth plane. The tapered-blade monopole antenna has a loss of 1.5 dB in the horizon plane, but the other one has no loss—Figs. 27.20, 27.21.

The $^\lambda/_4$ tapered-blade monopole antenna on the fin and the $^\lambda/_4$ thick-rod monopole antenna on the nose are installed. The radiation pattern of the omni-directional antennas are influenced by the UAV body. Diffraction of the UAV body and the wings should be especially regarded [5]. The radiation pattern of the installed antennas on the UAV is simulated by Fidelity software. The simulated results have been depicted in Figs. 27.22, 27.23. The thick-rod monopole antenna has been mounted $^3/_4\lambda$ from the metallic circular disk on the nose as shown in Fig. 27.22a. There is a deep null in the back of the azimuth-plane radiation pattern. The radiation pattern is symmetrical around the UAV body axis. In addition, the gain is 3.5 dB in front of the UAV, and there is a good beamwidth to cover the fore half-space.

The $^\lambda/_4$ tapered-blade monopole antenna has been installed on the fin and simulated by Fidelity software as shown in Fig. 27.23a. Azimuth-plane and elevation-plane radiation pattern have been shown in Figs. 27.23b and 27.23c, respectively. The antenna has a gain of 2 dB in the rear of the UAV, and there is a good beamwidth to cover the back half-space.

Although the *fin antenna* can cover 360° around the UAV, but the loss at $\phi = 0°$—perpendicular to the UAV body axis—is more than the previous case. Moreover,

Fig. 27.19 The $^\lambda/_4$ tapered-blade monopole antenna at 830 MHz as the fin antenna

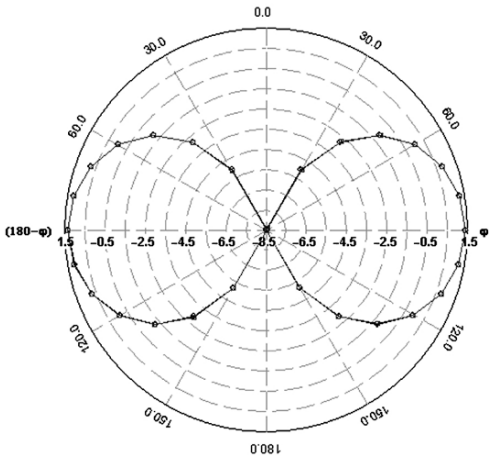


Fig. 27.20 Elevation-plane radiation pattern of the $\lambda/4$ tapered-blade monopole antenna

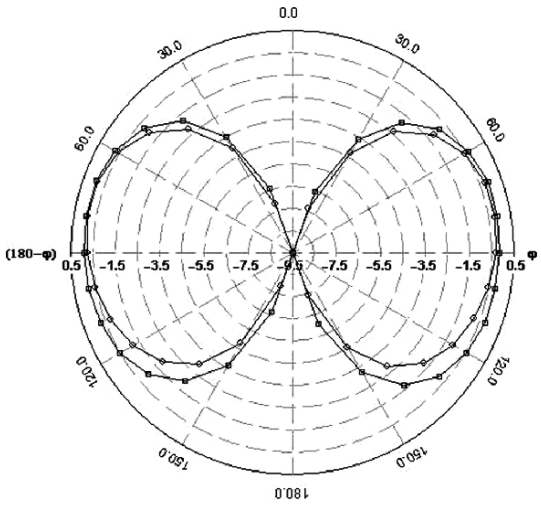


Fig. 27.21 Elevation-plane radiation pattern of the $\lambda/4$ thick-rod monopole antenna

the measurement results showed we cannot have a reliable radio link in both the toward and far from GS paths with just the fin antenna. Therefore, we should use two antennas for the radio link.

27.4 Short and Long Ranges Coverage

LOS Loss between UAV and GS is similar to free space loss if the distance is limited to radio horizon. The maximum LOS range in regard to the earth curvature is calculated as follows:

$$R = 4.12\,(\sqrt{h_a} + \sqrt{h_b}) \tag{27.4}$$

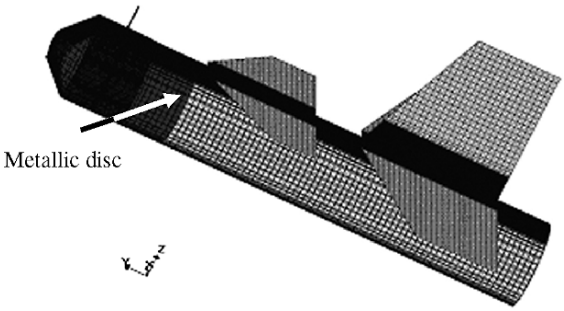


Fig. 27.22a The $\lambda/4$ thick-rod monopole antenna on the nose

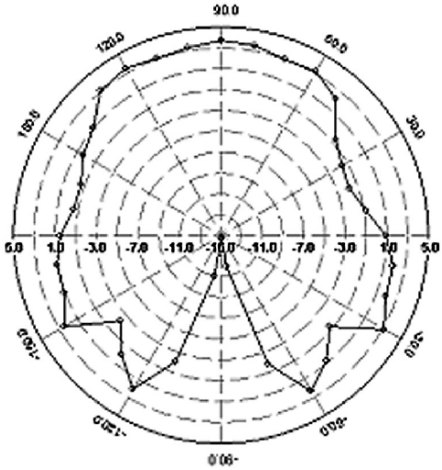


Fig. 27.22b Azimuth-plane radiation pattern of the nose antenna

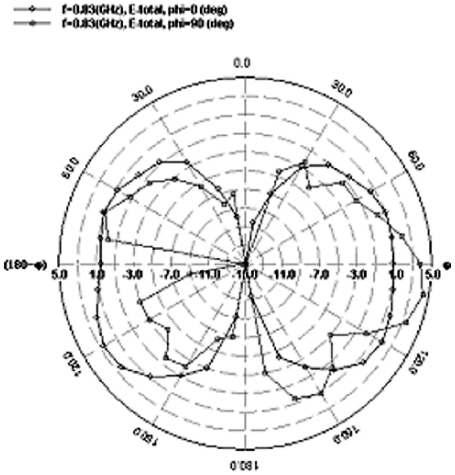


Fig. 27.22c Elevation-plane radiation pattern of the nose antenna at $\phi = 0^\circ$ (perpendicular to the UAV body axis) and $\phi = 90^\circ$ (UAV body axis) planes

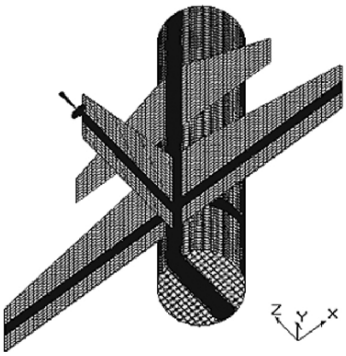


Fig. 27.23a The $\lambda/4$ tapered-blade monopole antenna on the fin

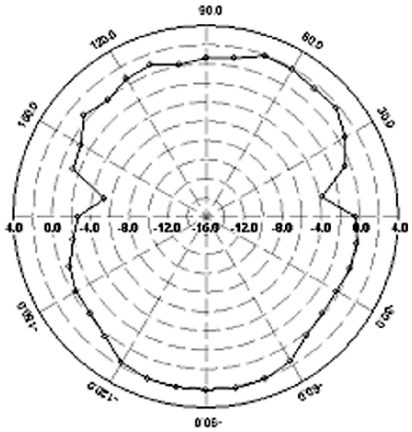


Fig. 27.23b Azimuth-plane radiation pattern of the fin antenna

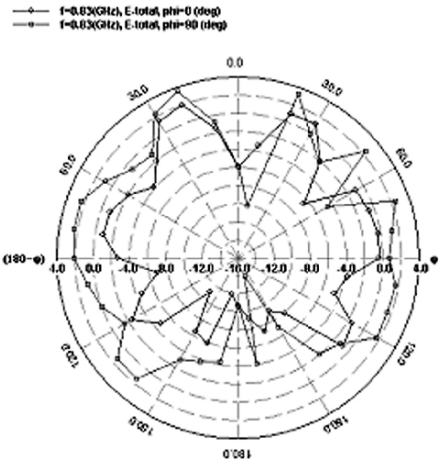


Fig. 27.23c Elevation-plane radiation pattern of the fin antenna at $\phi = 0^\circ$ (perpendicular to the UAV body axis) and $\phi = 90^\circ$ (UAV body axis) planes

$R(km)$ is the maximum LOS range, $h_a(m)$ is the height of the flying UAV, and $h_b(m)$ is the height of the installed GS antenna. The effect of the *Fresnel zone* is very important in the LOS region because direct waves and reflected waves from the ground can be added or subtracted; therefore, the received power can be changed, which is known as the *multipath fading* effect. This effect is the most important problem in mobile wireless systems that are working in radio frequencies less than 10GHz since it causes the received power to change from 0 to 10 dB in comparison to the free space loss [14]. It is observed that the power fluctuation of vertical polarization is less than that of horizontal polarization. One of the useful methods to reduce the multipath fading effect is a switched antenna diversity method applyed to the UAV antennas [17]. The more distance in terms of wavelength between the two UAV antennas, the less correlation there is between the received signals. It is noticeable that installation of antennas in the UAV body axis direction is a more efficient way to reduce the multipath fading effect than antennas perpendicular to it [13].

One oscillator with output power of 22 dB at 830 MHz is set up on the UAV to fly on a low high mountains region at the range of 60 km and the height of 1000 m. Firstly, the thick-rod monopole antenna is installed on the nose at a distance of $3/4\lambda$ from the circular metallic disk. The power received by the helical antenna has been measured in both toward GS and away from GS paths as shown in Fig. 27.24a. Secondly, the tapered-blade antenna has been set up on the fin, and the received power curve in both of the paths is depicted in Fig. 27.24b.

The multipath fading effect is *frequency selective*; hence, the received power can be increased by the GS antenna, which leads to the improvement of radio link performance. The transceiver system has been designed based on spread spectrum—frequency hopping—to enhance the radio link performance against the multipath fading effect. The UAV antenna selection algorithm of switched antenna diversity method is very simple. The main condition is if one of the antennas does not have good conditions to use in the radio link, the other one may have good conditions. This algorithm has been chosen because of the radiation pattern of the UAV antennas and the reduction of system complexity. The selection factor is based on bit error rate (BER). If the BER is more than the threshold, the antenna will be changed [6].

With respect to the UAV flying height, it is recommended that GS antennas—especially helix antennas—be installed at a height of 4 m. The first break point—the furthest break point away from GS—is at the range of 21 km [15] as shown in Fig. 24. The multipath loss can not degrade the data radio link in this range because of the high level of received power. It is necessary to elevate the helix axis up about 4° to compensate for the ground effect on the radiation pattern. Flight tests confirmed the preceding recommendation [7, 8].

The desired system is based on GCS, i.e., it can find the location of UAV by GPS or received RF signals; therefore, it can direct the helix to the UAV and track. In the other hand, the fast maneuvering of the UAV around the GS, especially during takeoff and landing, should be regarded; consequently, GCS can not follow the UAV simultaneously and quickly. Moreover, the UAV can be controlled visually, so omni-directional antennas can be utilized in the short ranges—less than 5 km.

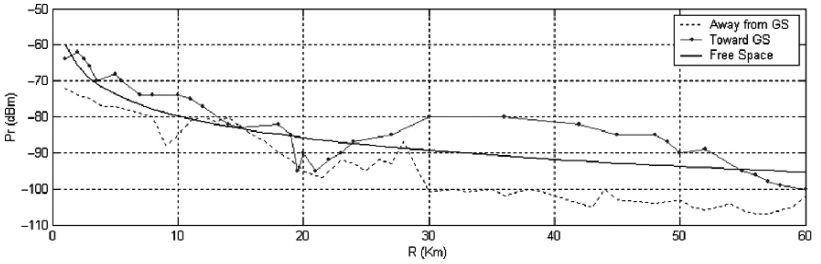


Fig. 27.24a Received power by the $\lambda/4$ thick-rod monopole antenna on the Nose

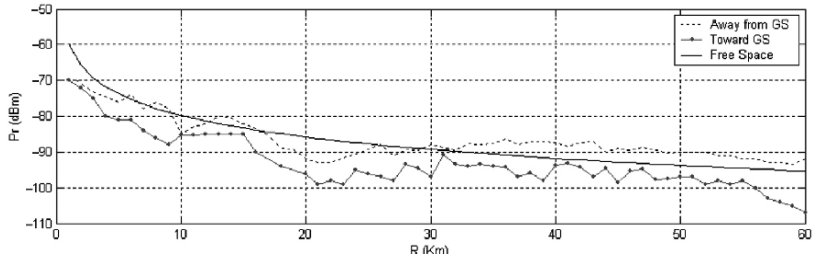


Fig. 27.24b Received power by the $\lambda/4$ tapered-blade monopole antenna on the fin

An RF switch can be utilized in the GS Transceiver to select the desired antenna electronically.

According to the various flight tests, when the thick-rod monopole antenna and the tapered-blade monopole antenna have been installed on the nose and the fin, respectively, BER is better than 4×10^{-5} for 32 dBm output power at the range of 60 km. Similarly, in some flight tests accomplished at the range of 100 km, the data radio link had good performance.

27.5 Conclusion

Design and implementation of GS antennas have been presented to control UAV by use of data radio link between the UAV and the GS based on a spread spectrum bandwidth of 780–880 MHz at the maximum range of 100 km. Helix and omni-directional antennas have been designed and implemented to cover short and long ranges with high reliability. A monopole antenna has been used for short ranges—less than 5 km. It has been designed, simulated and implemented. Measurements indicate the monopole antenna has a VSWR < 2 in the 780–880 MHz frequency band. Vertical-plane HPBW is about 65°. The gain of the monopole antenna is 2.5 dB at an elevation angle of 40°, and the loss is 2 dB in the horizontal plane.

A helix antenna has been used to cover ranges more than 5 km to establish a radio data link. It has been designed, simulated and matched to 50Ω in a frequency bandwidth of 100 MHz by a new stub-loaded matching method. It has a VSWR < 2

in a bandwidth of 100 MHz. The radiation pattern of the helix is endfire with a gain of 12.5 dB in the direction of helix axis. HPBW in both the elevation and azimuth planes is quite 32°. There is good agreement between Emerson's relations and measurement.

Many flight tests have been done to find the best locations for the antenna installation on the UAV. Two antennas are necessary to cover 360° around the UAV. A $\lambda/4$ thick-rod monopole antenna has been installed on the nose, and a $\lambda/4$ tapered-blade monopole antenna has been put on the fin to cover the fore and the rear half-space, respectively. Both antennas are vertical-polarized omni-directional, and have a $VSWR < 2$ at a frequency of 830 MHz. Both antennas have the omni-directional radiation pattern in the azimuth plane. The tapered-blade monopole antenna has a loss of 1.5 dB in the horizon plane, but the other one has no loss. The gain is 3.5 dB in front of the UAV, and there is a good beamwidth to cover the fore half-space. The antenna has a gain of 2 dB in the rear of the UAV, and there is a good beamwidth to cover the back half-space.

According to the multipath effect and LOS relations, GS antennas should be installed at a height of 4 m in regard to the UAV flying height. It is necessary to elevate the helix axis about 4° to compensate for the ground effect on the radiation pattern. The UAV antenna selection algorithm is a switched antenna diversity method based on BER as the selection factor. When the thick-rod monopole antenna and the tapered-blade monopole antenna have been installed on the nose and the fin, respectively, BER is better than 4×10^{-5} for 32 dBm output power at a range of 60 km. Similarly, some flight tests have been accomplished at the range of 100 km; the data radio link has good performance.

Acknowledgements The authors would like to thank the staff of the Information and Communication Technology Institute (ICTI), Isfahan University of Technology (IUT), Iran, for their cooperation and support for this work.

References

1. Arai H (2001) Measurement of Mobile Antenna Systems, Artech House, Norwood, MA 02062 USA
2. Balanis CA (1997) Antenna Theory, Analysis and Design, Hoboken, NJ, John Wiley & Sons
3. Emerson D, AA4FV (1995) The gain of an axial-mode helical antenna. The ARRL Antenna Compendium, vol. 4, USA
4. Elliott RS (2003) Antenna Theory and Design. rev. ed. Hoboken, NJ, John Wiley & Sons
5. Fujimoto K, James JR (2000) Mobile Antenna Systems Handbook, 2nd ed. Artech House, Norwood, MA 02062 USA
6. Firouzeh ZH, Haji-Hashemi MR, Moosavian SM, MirMohammad-Sadeghi H (2005) Reduction of UAV body effect and multipath fading in the high range of UHF radio link. The 13th ICEE, Zanjan University, Iran, pp 377–382
7. Firouzeh ZH, Moradian M, Safari-Hajat-Aghaei A, Mirmohammad-Sadeghi H (2006) Design & Implementation of ground system antennas for UAV data radio link in UHF band. 2nd IEEE International Conf. on Information & Communication Technologies; from Theory to Applications, Syria

8. Firouzeh ZH, Safari-Hajat-Aghaei A, and Mirmohammad-Sadeghi H (2007) Design and Implementation of Stub-Loaded Helical Antenna for UAV data radio link. International Multiconference of Engineers and Computer Scientists, Hong Kong, pp 1797–1801
9. http://airtraffic1.stanford.edu/∼uav/
10. King HE, Wong JL (1993) Helical antennas. In: Antenna Engineering Handbook. 3rd ed. L. Johnson, McGraw-Hill, USA
11. Kraus JD (1977) A 50-ohm input impedance for helical beam antennas. IEEE Trans. Antenna Propagation ap-25(6)913
12. Kraus JD (1988) Antennas. 2nd ed. New York, Columbus, OH, McGraw-Hill
13. Lee William CY (1989) Mobile Cellular Telecommunications Systems. Columbus, OH, McGraw-Hill
14. Lee William CY (1993) Mobile Communications Design Fundamentals. 2nd ed. Hoboken, NJ, John-Wiley
15. Lee William CY (1997) Mobile Communications Engineering. 2nd ed. Columbus, OH, McGraw-Hill
16. Munson K (1988) World Unmanned Aircraft. JANE'S, New York, pp. 7–10
17. Manning T (1999) Microwave radio Transmission Design Guide. Artech House, Norwood, MA 02062 USA
18. The American Radio Relay League (1988) The ARRL Antenna Book, 15th ed, USA

Chapter 28
Efficient 2D Linear-Phase IIR Filter Design and Application in Image Processing

Chi-Un Lei, Chung-Man Cheung, and Ngai Wong

We present an efficient and novel procedure to design two-dimensional (2D) linear-phase IIR filters with less hardware resource. A 2D linear-phase FIR filter prototype is first designed using semidefinite programming (SDP). The prototype filter is then decomposed into modular structures via Schur decomposition method (SDM). Each section is reduced into IIR structures using a novel digital system identification technique called the discrete-time vector fitting (VFz). Examples with image processing application show that the algorithm exhibits fast convergence and produces low hardware cost and accurate filters.

28.1 Introduction

Two-dimensional (2D) filters are widely used in image processing [1], medical imaging [2], face recognition [3], etc. These applications often require high-order filters with accurate magnitude response and linear phase in the passband. For instance, a 63 pixel $\times$ 63 pixel kernel filter is used in medical imaging [2]. However, hardware resources are usually restricted due to limited multipliers and memory in ASICs and FPGAs [4]. Therefore, 2D IIR filters are generally used to reduce the hardware cost. However, to date there is no optimal algorithm for 2D IIR filter design in terms of computation and the resultant hardware cost. Direct optimization of an IIR filter gives excellent accuracy but the computation complexity is high [5]. In the singular value decomposition (SVD) approach, the frequency response of a 2D FIR prototype filter is replaced with parallel sections of 1D cascaded subfilters [6]. The problem is then reduced from a 2D design problem into several 1D design tasks, thus producing less complicated implementation due to parallelization and

Chi-Un Lei, Chung-Man Cheung, and Ngai Wong
Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong
e-mail: {culei,edmondcheung,nwong}@eee.hku.hk

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

modularization of filter sections [7]. By neglecting sections associated with small singular values, the decomposed filter can be simplified with only slight error in the filter response [6]. Moreover, subfilters can be reduced by IIR approximation using model order reduction methods to further reduce hardware costs. However, this method is complicated when the subfilter sizes and the number of sections are large. To this end, we present a novel design flow for designing 2D (near-)linear-phase IIR filters with low computational complexity and hardware cost. First, semi-definite programming (SDP) is used to design a 2D FIR filter prototype, followed by the Schur decomposition method (SDM) that decomposes the prototype filter into sections of cascaded 1D FIR subfilters. A novel digital system identification technique, called the discrete-time vector fitting (VFz), is then used to reduce the 1D FIR subfilters into 1D IIR subfilters. It is shown that VFz gives efficient and accurate IIR approximation over conventional schemes. A practical image processing example demonstrates that the integrated SDP/SDM/VFz design flow produces excellent 2D IIR approximants with good passband phase linearity and low hardware costs.

28.2 Design Methodology

28.2.1 FIR prototype Design via Semidefinite Programming

The transfer function of a 2D FIR filter of odd order (N_1, N_2) is characterized by

$$H(z_1,z_2) = \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} h_{ij} z_1^{-i} z_2^{-j} = z_1^T \hat{H} z_2, \tag{28.1}$$

where $z_i = \left[\, 1 \; z_i^{-1} \; \ldots \; z_i^{-(N_i-1)} \,\right]^T$ for i = 1 and 2, and $\hat{H} \in R^{N_1 \times N_2}$ is impulse response. This kind of filter with phase linearity becomes octagonal-symmetric [7]. Therefore, $\hat{H}$ can be partitioned as

$$\hat{H} = \begin{bmatrix} H_{11} & h_{12} & H_{13} \\ h_{21}^T & h_{22} & h_{23}^T \\ H_{31} & h_{32} & H_{33} \end{bmatrix} \tag{28.2}$$

where $H_{11}, H_{13}, H_{31}, H_{33} \in R^{n_1 \times n_2}, h_{12}, h_{32} \in R^{n_1 \times 1}, h_{21}, h_{23} \in R^{n_2} \times 1, h_{22} \in R$ and $n_1 = (N_1 - 1)/2, n_2 = (N_2 - 1)/2$. Moreover, the frequency response is given by

$$H(\omega_1,\omega_2) = e^{-j(n_1\omega_1 + n_2\omega_2)} c_1^T(\omega_1) H c_2(\omega_2) \tag{28.3}$$

where $c_i(\omega_i) = [\, 1 \; \cos\omega_i \; \ldots \; \cos n_i\omega_i \,]^T$, for i = 1 and 2, and $H = \begin{bmatrix} h_{22} & 2h_{23}^T \\ 2h_{32} & 4H_{33} \end{bmatrix}$.

The design of the prototype FIR filter is a minimax problem of error of H in (ω_1, ω_2) [5]. It can be formulated as

$$\begin{array}{c} \min \qquad c^T x \\ \text{subject to}: F(x) \geq 0 \end{array} \tag{28.4}$$

where $c = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T$, and $x = \begin{bmatrix} \delta & h^T \end{bmatrix}^T$, in which $F(x) = diag\{\Gamma(\omega_1^{(1)}, \omega_2^{(2)}), \ldots, \Gamma(\omega_1^{(M)}, \omega_2^{(M)})\} \geq 0$. Here $\geq$ denotes matrix positive semidefiniteness. A_{dw} is the weighted desired magnitude response. By stacking from the first to last columns of H and c_{kn_1+m}, h and c_ω are column vectors, respectively. $c_{kn_1+m}(\omega_1, \omega_2) = \cos(m\omega_1)\cos(k\omega_2)$ for $0 \leq m \leq n_1$, and $0 \leq k \leq n_2$. This is a semidefinite programming (SDP) problem. SDP is an optimization framework wherein a linear or convex objective function is minimized subject to linear matrix inequality (LMI)-type constraints.

28.2.2 Schur Decomposition of 2D FIR Filters

The octagonal-symmetric linear phase 2D FIR filters can be decomposed into 1D subfilters by Schur decomposition method (SDM) [7]. SDM is superior to SVD in computational complexity by exploitation of filter response symmetry. The idea of SDM is to decompose the 2D transfer function as

$$H(z_1, z_2) = \sum_{l=1}^{k} F_l(z_1) G_l(z_2), \tag{28.5}$$

where $F_l(z_1) = \sum_{i=0}^{N_1-1} f_l(i) z_1^{-i}$ and $G_l(z_2) = \sum_{j=0}^{N_2-1} g_l(j) z_2^{-j}$ are the transfer functions of two cascaded 1D subfilters, $f_l(i)$ and $g_l(j)$ are the corresponding impulse responses, and k is the number of parallel sections. k is chosen regarding the tradeoff between computation and approximation accuracy. With the octagonal symmetry, $\hat{H}$ in (28.2) can be rewritten as

$$\begin{aligned} \hat{H} &= \begin{bmatrix} I_L & 0 \\ \hat{I} & I_M \end{bmatrix} \begin{bmatrix} H_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_L & \hat{I}^T \\ 0 & I_M \end{bmatrix} \\ &= \begin{bmatrix} H_1 & H_1 \hat{I}^T \\ \hat{I} H_1 & \hat{I} H_1 I^T \end{bmatrix} \end{aligned} \tag{28.6}$$

where $\hat{I} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 1 & 0 & 0 \\ \vdots & \iddots & & \vdots & \vdots \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$ and $L = n_i + 1$. $H_1 \in R^{L \times L}$ contains all the impulse response information of the 2D FIR filter [7]. H_1 is then decomposed by means of SDM

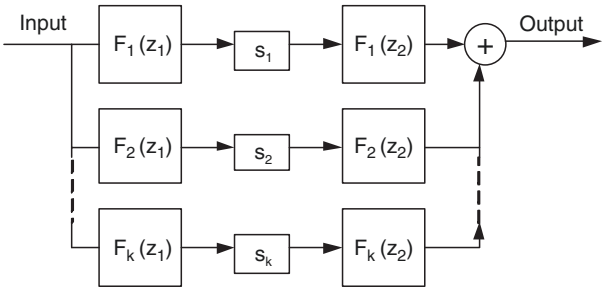


Fig. 28.1 Decomposed FIR prototype filter

$$U^T H_1 U = \sum \tag{28.7}$$

where $U^T U = UU^T = I_L$ and $\sum = diag(\overline{\lambda}_1, \overline{\lambda}_2, ..., \overline{\lambda}_L)$. Retaining the k most significant eigenvalues in $\sum$, namely, $\sum_1 = \sum(1:k, 1:k)$, the approximated 2D FIR filter becomes

$$\tilde{H}_d = W\,|\Sigma_1|\,SW^T = FSG = FSF^T \tag{28.8}$$

where $W = \begin{bmatrix} U(:, 1:k) \\ \hat{I}U(:, 1:k) \end{bmatrix}$, $F = W\sqrt{|\sum_1|} \in R^{N\times k}$, $G = F^T$, and $S = diag\left(s_1, s_2, ..., s_k\right)$, in which $s_l = sign\left(\bar{\lambda}_l\right)$ is the sign weight for interconnection between subfilters. Each column of F is an FIR linear phase 1D subfilter with its own frequency response. Therefore, Eq. (28.8) becomes

$$\tilde{H}_d = \sum_{l=1}^{k} F_l s_l F_l^T \tag{28.9}$$

$\tilde{H}_d$ preserves phase linearity of the 2D FIR filter. The architecture of the decomposed filter is shown in Fig. 28.1.

28.2.3 IIR Filter Approximation of 1D FIR Subfilters

For the sake of hardware savings, and consequently power consumption, each of the 1D FIR subfilters (F_1, F_2,..., F_k) is approximated by IIR structures

$$\sum_{n=0}^{L} h_n z^{-n} = F_l(z) \approx \widehat{f}(z) = \frac{P(z)}{Q(z)} = \frac{\sum_{n=0}^{N} p_n z^{-n}}{\sum_{m=0}^{M} q_m z^{-m}}. \tag{28.10}$$

where $\widehat{f}(z)$ is the IIR approximant. We aim at locating a set of p_n and q_m with N, M $\ll$ L to form a stable and causal IIR filter with a good approximation subject to constraints like magnitude and phase response, and low algorithmic complexity.

28.3 Discrete-Time Vector Fitting

Vector fitting (VF) [8] is a popular technique for fitting a continuous-time (s-domain) frequency-dependent vector/matrix with rational function approximations. VF starts with multiplying a scaling function σ(s) to the desired response f(s). The poles on both sides of the equality are set to be equal:

$$\underbrace{\left(\sum_{n=1}^{N}\frac{c_n}{s-\alpha_n}\right)+d+se}_{(\sigma f)(s)}\approx\underbrace{\left(\left(\sum_{n=1}^{N}\frac{\gamma_n}{s-\alpha_n}\right)+1\right)}_{\sigma(s)}f(s) \tag{28.11}$$

The basis of partial function ensures well-conditioned arithmetic. The poles (α_n) and residues (γ_n) are either real or exist in complex conjugate pairs. The variables c_n, d, e, and γ_n are solved by evaluating Eq. (28.11) at multiple frequency points. In Eq. (28.11), the set of poles of $(\sigma f)(s)$ and $\sigma(s) f(s)$ are the same. Therefore, the original poles of f(s) cancel the zeros of $\sigma(s)$, which are assigned as the next set of known poles to Eq. (28.11). This iteration process continues until the poles are refined to the exact system poles. In general, it only takes a few iterations. VF is readily applicable to digital domain (z-domain), called discrete-time vector fitting (VFz), for IIR approximation of FIR filters [9]. In the VFz approach, an initial set of stable poles $\left\{\alpha_n^{(0)}\right\}$ is first assigned to be refined. Analogous to VF, the desired 1D FIR filter response is fitted with a rational function:

$$\underbrace{\left(\sum_{n=1}^{N}\frac{c_n}{z^{-1}-\alpha_n^{(i)}}\right)+d}_{(\sigma f)(z)}\approx\underbrace{\left(\left(\sum_{n=1}^{N}\frac{\gamma_n}{z^{-1}-\alpha_n^{(i)}}\right)+1\right)}_{\sigma(z)}f(z) \tag{28.12}$$

In digital systems, it is required that $|\alpha_n|>1$ because stable poles are inside the unit circle. For N_s frequency points at z = z_m $(m=1,2,\ldots N_s)$ and $N_s \gg 2N+1$, Eq. (28.12) is presented in an overdetermined linear equation

$$\left(\sum_{n=1}^{N}\frac{c_n}{z_m^{-1}-\alpha_n^{(i)}}\right)+d-\left(\sum_{n=1}^{N}\frac{\gamma_n f(z_m)}{z_m^{-1}-\alpha_n^{(i)}}\right)\approx f(z_m), \tag{28.13}$$

where i is the ith iteration. It can be solved by

$$Ax=b \tag{28.14}$$

where the mth row in A, A_m, and entries in the column vector b, b_m, and x are $A_m = \left[\frac{1}{z_m^{-1}-\alpha_1^{(i)}} \cdots \frac{1}{z_m^{-1}-\alpha_N^{(i)}} \; 1 \; \frac{-f(z_m)}{z_m^{-1}-\alpha_1^{(i)}} \cdots \frac{-f(z_m)}{z_m^{-1}-\alpha_N^{(i)}} \right]$, $x = \left[c_1 \ldots c_N \; d \; \gamma_1 \ldots \gamma_N \right]^T$, and $b_m = f(z_m)$.

To determine the new poles (the reciprocals of zeros of $\sigma(s)$) for the next iteration, the poles are computed by the eigenvalues of the following function:

$$\Psi = \begin{bmatrix} \alpha_1^{(i)} & & & \\ & \alpha_2^{(i)} & & \\ & & \ddots & \\ & & & \alpha_N^{(i)} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} \gamma_1 \cdots \gamma_N \end{bmatrix} \tag{28.15}$$

To ensure stability, $\left|\alpha_n^{(i+1)}\right|$ must be greater than 1. Unstable poles are relocated by flipping their reciprocals, $1/\alpha_n^{(i+1)}$, into the unit circle. This is possibly done by multiplying both sides of Eq. (28.12) by an allpass filter $\frac{z^{-1}-\alpha_n^{(i+1)}}{1-\overline{\alpha}_n^{(i+1)} z^{-1}}$. Here, only the phase is changed. When all the poles converge, $\sigma(z) \approx 1$. It turns out that the IIR denominator part is determined by

$$\tilde{f}(z_m) = \left(\sum_{n=1}^{N} \frac{c_n}{z_m^{-1} - \alpha_n^{(N_T)}} \right) + d \approx f(z_m) \tag{28.16}$$

VFz is also extended to handle complex conjugate poles commonly found in digital filters. The accuracy and computational complexity of VFz are dependent on the order of the 1D subfilters, the number of iterations and frequency-sampling points. In short, VFz improves the approximation accuracy successively by using deterministic pole relocation techniques. It is shown by experiments that its accuracy is comparable to, if not better than, that of model reduction techniques [10, 11], but with much less computational complexity. The saving is even more significant when the number of subfilter sections is large. This result is remarkable. Besides magnitude approximation, VFz simultaneously performs accurate phase approximation, whose linearity is particularly important in image processing.

28.4 Numerical Example

We would like to verify the performance using two numerical examples. The proposed design methodology is illustrated with a practical lowpass filter example similar to that in Lu. A diamond-shape linear-phase FIR filter with order = (37, 37) is used whose specification is

$$W(\omega_1, \omega_2) = \begin{cases} 0\text{dB}, & \text{for} |\omega_1| + |\omega_2| \leq 0.8\pi \\ -40\text{dB}, & \text{for} |\omega_1| + |\omega_2| > \pi \end{cases} \tag{28.17}$$

Uniformly distributed grid points of 47 × 47 are used. The algorithm is coded in Matlab m-script file and run under a Matlab 7.2 environment on a 1G RAM 3.4GHz computer. The filter specification is first converted into an SDP problem containing 5,651 equations. The prototype filter is decomposed into five sections (k = 5) by SDM, which is the most important subsection (sections with the five largest eigenvalues proportion, shown in Fig. 28.5). Figures 28.2 and 28.3 show the filter response magnitude after using SDP and SDM, respectively. Each section is then approximated by a 1D IIR subfilter using VFz with orders 19, 20, 19, 20, and 21, respectively. The numerator and denominator in each 1D filter are of the same order. One hundred thirty sampling points and five iterations are used in VFz. Figure 28.4 shows the frequency response of the final 2D IIR filter. The normalized rms errors of the IIR filter approximant are 0.4 percent and 0.6 percent in the passband and stopband, respectively. The normalized rms errors between the final design and the ideal design are 4 percent and 5 percent, respectively. Furthermore, as seen in the figure, the IIR filter approximant preserves linearity (constant

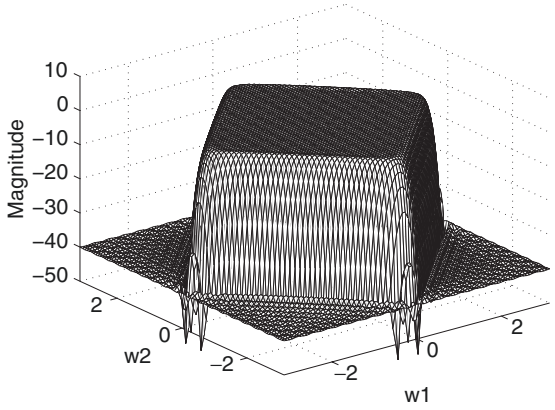


Fig. 28.2 Magnitude response of the 2D prototype lowpass FIR filter by SDP

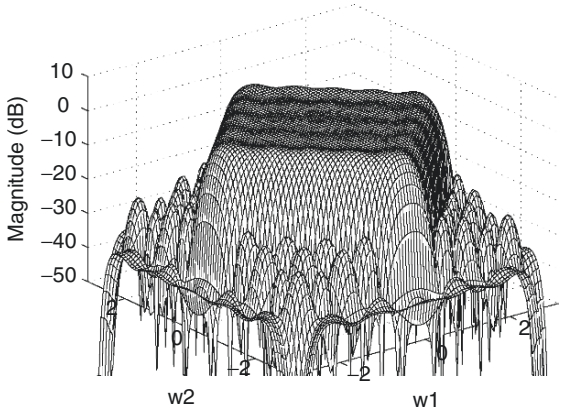


Fig. 28.3 Magnitude response of the approximated 2D lowpass FIR filter after SDM

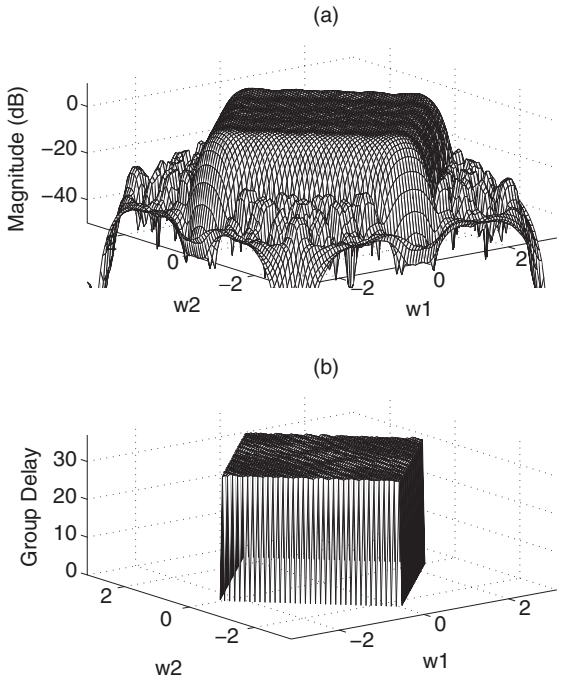


Fig. 28.4 Frequency response of the approximated 2D lowpass IIR filter via VFz: (a) magnitude response, and (b) group delay in the passband

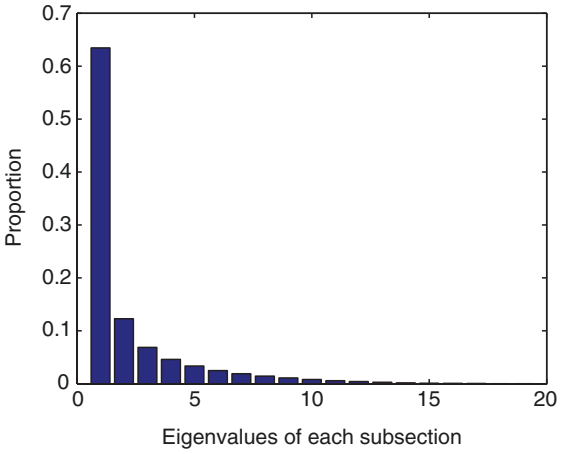


Fig. 28.5 Eigenvalues of the lowpass filter example H_1 of (28.7) in ratio, which shows the importance of each subsection

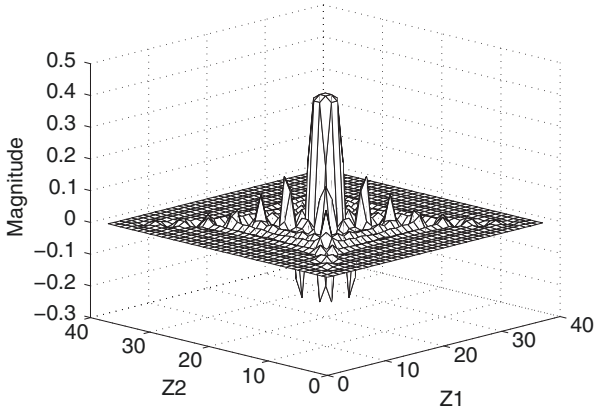


Fig. 28.6 Impulse response of the approximated 2D IIR lowpass filter via VFz

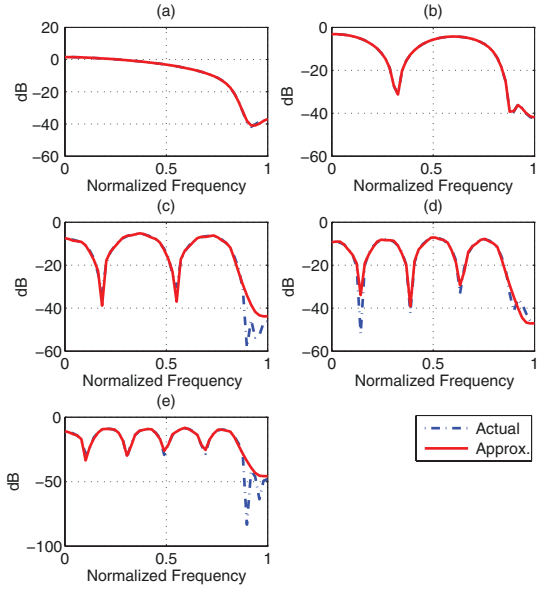


Fig. 28.7 Magnitude response of subfilter approximation using VFz. (a)–(e) are subfilters of Sections 1–5, respectively

group delay) in the passband and the approximation error is mainly introduced in SDM. The computation time is 507 CPU seconds for FIR prototype filter design (using SDP) and only 1.51 CPU seconds for IIR approximation (using Schur decomposition and VFz approximation). Its advantage in fast computation is therefore demonstrated. Figure 28.6 shows the impulse response of the approximated 2D IIR lowpass filter via VFz. Figure 28.7 shows that VFz can achieve good approximations for subfilter design. The normalized errors of subfilter approximations are 0.1 percent, 0.1 percent, 4 percent, 4 percent, and 5 percent, respectively. Compared to a

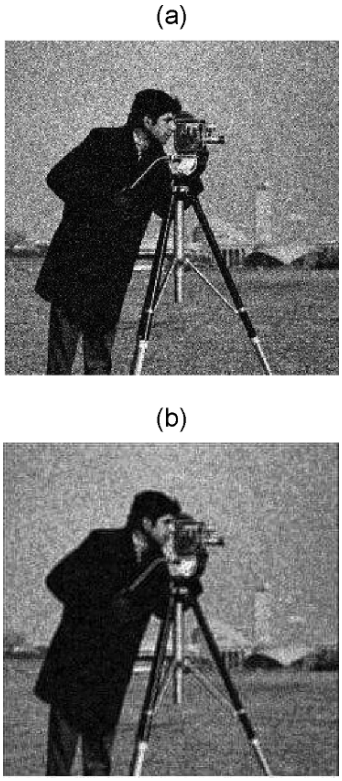


Fig. 28.8 Images in the numerical example: (a) noise-corrupted image, and (b) filtered result using 2D IIR filter

direct implementation of the original 2D FIR filter, the proposed SDP/SDM/VFz design flow has resulted in a hardware savings (essentially multipliers) of more than 50 percent. Figure 28.8 shows an image noise filtering example.

A bandreject filter example is used in the second example. Bandreject filters are popularly used to remedy images corrupted by additive periodic noise with a known frequency. A circular-shape linear-phase FIR filter with order = (37, 37) is used, whose specification is

$$W(\omega_1,\omega_2)=\begin{cases} 0\text{dB}, & \text{for } 0<\sqrt{\omega_1^2+\omega_2^2}\le 0.5\pi \\ 0\text{dB}, & \text{for } 0.8\pi<\sqrt{\omega_1^2+\omega_2^2}\le \pi \\ -40\text{dB}, & \text{for } 0.6\pi<\sqrt{\omega_1^2+\omega_2^2}<0.7\pi \end{cases} \tag{28.18}$$

Uniformly distributed 47 $\times$ 47 grid points are used. The prototype filter is decomposed into four sections (k = 4) by SDM. Figure 28.11 shows the proportion of the eigenvalues. The figure shows that the first few sections have the large rest

eigenvalues and contribute most proportion of filter characteristic. Each section is then approximated by a 1D IIR subfilter using VFz with orders 18, 19, 20, and 21, respectively. The numerator and denominator in each 1D filter are of the same order. One hundred sampling points and seven iterations are used in VFz. Figures 28.9 and 28.10 show the frequency response and impulse response of the final 2D IIR

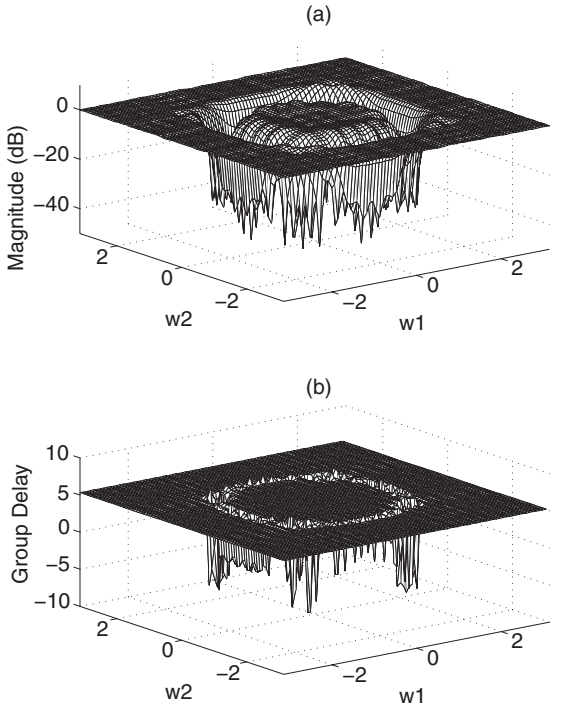


Fig. 28.9 Frequency response of the 2D bandstop IIR filter via the proposed algorithm: (a) magnitude response and (b) group delay in the passband

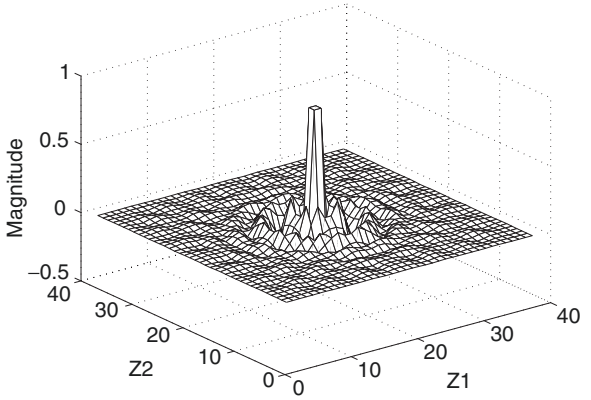


Fig. 28.10 Impulse response of the approximated 2D FIR bandstop filter via VFz

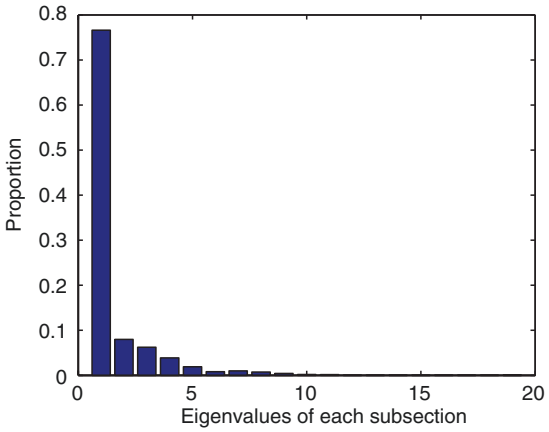


Fig. 28.11 Eigenvalues of the bandstop filter example H_1 of (28.7) in ratio, which shows the importance of each subsection

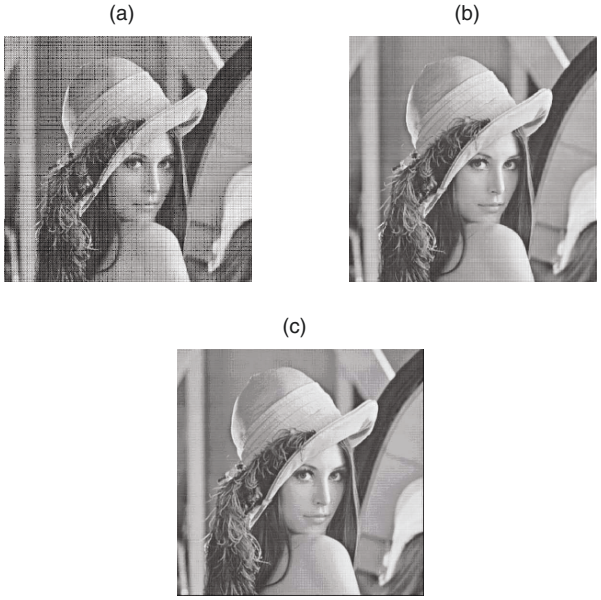


Fig. 28.12 Images in the numerical example: (a) noise-corrupted image, (b) filtered result using 2D FIR prototype filter, and (c) filtered result using 2D IIR filter

filter, respectively. As seen in the figure, the IIR filter approximant preserves linearity (therefore, constant group delay) in the passband. Two figures of noise removal are shown in Fig. 28.12(b) and Fig. 28.12(c) using a 2D FIR prototype filter and a 2D IIR filter, respectively. They both show that noise is greatly reduced without much blurring of the original image. The degradation in image quality is small when the 2D FIR prototype filter is replaced with the 2D IIR filter.

The computation time is 528 CPU seconds for FIR prototype filter design (using SDP) and 1.2987 CPU seconds for IIR approximation (using Schur decomposition and VFz approximation). Its advantage in fast 2D IIR approximation is therefore demonstrated. The number of multipliers in the 2D IIR filter is 156, which saves 45.5 percent of the multipliers of the original symmetric filter $(37 \times 37 \div 4 = 343)$. Consequently, it is demonstrated that the proposed IIR filter design flow reduces hardware costs while preserving similar filtering quality when compared to a direct implementation of the original 2D FIR filter.

28.5 Remarks

1. Besides using SDP, other common and faster design techniques, such as the windowing method, can be used to design the 2D FIR filter prototype. The SDM/VFz post-processing can then be used to achieve even faster IIR approximation.
2. The most direct approach is to direct-decompose the desired frequency response into sections using frequency SVD and then realize each section by VFz. This can avoid introduced errors and relative time consumed in the 2D FIR prototype filter design. This modification can generate a 2D IIR filter within a few seconds.
3. In addition to reducing multipliers, the proposed filter structure is also favorable for VLSI implementation. The identical subfilters exhibit regular and modular structures, which lead to a reduction in interconnect area and simple floor planning and layout. Multiplierless filter design techniques are also available for further reducing the hardware cost within the IIR filter structures [12].
4. This paper has extended the vector fitting concept to the 2D discrete-time domain. The idea can be further generalized to n-D IIR filter design, which is useful in video processing and medical imaging.
5. Besides the bandreject filter in our example, the proposed algorithm is also applicable to the efficient construction of 2D lowpass, highpass, and bandpass filters, with possible applications in image noise removal and edge detection, etc.
6. The relationship of error among a number of sections and filter orders is still being investigated. The investigation objective is to fully integrate Schur decomposition and VFz such that filters can be designed within a controlled error and with the lowest hardware cost. As the frequency response in later orders is irregular, it is not optimal to use lower order subfilters for later subsections.
7. VFz can be generalized as a frequency masking filter design technique. It is similar to the WLS method [13], with novel weighting construction to simplify calculation. Furthermore, VFz can limit the maximum pole radius, which can control the sensitivity of the quantization error.

28.6 Conclusion

A new 2D IIR filter design flow has been proposed that utilizes the SDP/SDM/ VFz integration to efficiently obtain accurate IIR approximants from 2D linear-phase FIR filter prototypes. Hardware cost is significantly reduced due to parallel and modular 1D IIR subfilters. Image processing examples have demonstrated that the proposed approach renders high approximation accuracy, low hardware costs, and low computational complexity, and effectively preserves passband phase linearity.

Acknowledgements This work was supported in part by the Hong Kong Research Grants Council and the University Research Committee of The University of Hong Kong.

References

1. Gonzalex R and Woods R (2001) Digital Image Processing, 2nd ed. Prentice-Hall, Upper Saddle River, N.J.
2. Kreuger R, Virtex-EM FIR filter for video applications. [Online]. Available: http://direct.xilinx.com/bvdocs/appnotes/xapp241.pdf
3. Gong SMS and Psarrou A (2000) Dynamic Vision: From Images to Face Recognition. 1st ed. Imperial College Press
4. Bouganis PCC, Constantinides G (2005) A novel 2D filter design methodology for heterogeneous devices. In IEEE FCCM05 pp. 13–22
5. Lu W (2002) A unified approach for the design of 2D digital filters via semidefinite programming. IEEE Trans. Circuits Syst. I, 49(6):814–826, June
6. Zhu MAWP and Swamy M (1999) Realization of 2-D linear-phase FIR filters by using the singular-value decomposition. IEEE Trans. Signal Processing 47(5) Mar. pp. 1349–1358
7. Aldhaheri R, Design of 2-D linear phase digital filters using schur decomposition and symmetries. 15(1) pp. 65–81
8. Gustavsen B and Semlyen A (1999) Rational approximation of frequency domain responses by vector fitting. IEEE Trans. Power Delivery 14(3) July pp. 1052–1061
9. Wong N and Lei C (2007) FIR filter approximation by IIR filters based on discrete-time vector fitting. In Proc. IEEE Symp. Circuits and Systems May 2007, to appear
10. Beliczynski B, Kale I, and Cain GD (1992) Approximation of FIR by IIR digital filters: an algorithm based on balanced model reduction. IEEE Trans. Signal Processing 40(3) Mar. pp. 532–542
11. Kwan HK and Jiang A (2006) Recent advances in FIR approximation by IIR digital filters. In Proc. Int. Conf. Communications, Circuits and Systems June pp. 185–190
12. Lei C and Wong N (2006) Multiplierless polynomial-operator-based IIR filters for efficient VLSI implementation. In IEEE TENCON
13. Lu W, Pei S, and Tseng C (1998) A weighted least-squares method for the design of stable 1-D and 2-D IIR digital filters. IEEE Trans. Signal Processing 46(1) Jan. pp. 1–10

Chapter 29
Image Enlargement Based on Low- to High-frequency Components Mapping Derived from Self-decomposed Images

Hideaki Kawano, Noriaki Suetake, Byungki Cha, and Takashi Aso

29.1 Introduction

Due to the wide diffusion of multimedia applications and wide area screens—for example, video walls—the problem of image size conversion (especially image enlargement) has significantly increased in importance. Various image enlargement methods have been proposed so far [1], and they are split into multi-frame image enlargement methods and single-frame methods.

The multi-frame image enlargement is a process of combining multiple frames of low-resolution images in the same scene for the reconstruction of a single high-resolution image. Many multi-frame image enlargement methods have been proposed up to now. For example, there are the frequency domain methods [2–4], the projection onto convex sets (POCS) methods [5, 6], and the maximum a posteriori (MAP) methods [7–9]. They can obtain good enlargement performance with respect to image quality. However, their computational complexity and storage consumption are tremendous. It is, therefore, very difficult to realize real-time processing by them.

On the other hand, the single-frame image enlargement is a process of reconstructing a high-resolution image only by using a low resolution single source image. The single-frame image enlargement is known as image scaling, image interpolation, image zooming, image super-resolution, and so on. In general, the

Hideaki Kawano
Kyushu Institute of Technology, Faculty of Engineering, Kitakyushu, Fukuoka, 804-8550, Japan
e-mail: kawano@ecs.kyutech.ac.jp

Noriaki Suetake
Yamaguchi University, Graduate School of Science and Engineering, Yamaguchi-shi, Yamaguchi, 753-8512, Japan
e-mail: suetake@sci.yamaguchi-u.ac.jp

Byungki Cha and Takashi Aso
Kyushu Institute of Information Sciences, Faculty of Management and Information Sciences, Dazaifu-shi, Fukuoka, 818-0117, Japan
e-mail: cha@kiis.ac.jp,taso@kiis.ac.jp

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

single-frame image enlargement method is inferior to multi-frame with respect to image quality. However, the practicality of the single-frame image enlargement method is very high, thus the single-frame image enlargement methods are far better than that of the multi-frame version. Typical single-frame image enlargement methods are based on interpolation by using kernels, and include the nearest neighbor interpolation (NNI), the bilinear interpolation (BLI), the cubic convolution interpolation (CCI), and so on [10,11]. These methods are very useful and powerful. In fact, some of them are used in major photo retouching software packages. However, they cannot restore the high-frequency image components lost in the sampling process. The resolution of the resultant image can hardly be improved, and the step edges and peaks of the image are not restored clearly. This is caused by the fact that high-frequency image components beyond the Nyquist frequency cannot be restored by these simple kernel-based methods.

To cope with these problems, various single-frame image enlargement methods have been proposed. For example, there are directional methods [12, 13], multiple kernel-based methods [14], orthogonal transform-based methods [15,16], nonlinear extrapolation methods (NE) [17,18], neural network-based methods [19–21], and so on. Although they have better performance than simple kernel-based methods, they are too complicated or highly mathematical for implementation. Furthermore, some of them are based on assumptions that are not always correct in practical situations, and thus they may become occasionally useless.

In this chapter, in order to realize the image enlargement method with the highest performance, we propose a new image enlargement method that restores high-frequency components lost in the sampling process by using a nonlinear mapping system constructed by pairs of low- and high-frequency components of the original image. The pairs of low- and high-frequency components are derived from the self-decomposed images that can be obtained through one-time, low-pass filtering and subtraction between the filtered image and the original image. As the mapping system, a least squares support vector machine (LS-SVM)[22] is employed here. Support vector machines (SVM) have been widely used in classification and nonlinear function estimation [23]. However, the major drawback of SVM is its higher computational burden for constrained optimization programming. This disadvantage has been overcome by the LS-SVM, which solves linear equations instead of a quadratic programming problem.

The rest of the paper is organized as follows. The algorithm of the proposed method is discussed in Sect. 29.2. Section 29.3 shows the experimental results, and the conclusions are described at the end.

29.2 Proposed Image Enlargement Method

The proposed image enlargement method goes through the following steps:

1. Generate pairs of low- and high-frequency components derived from self-decomposed images to train a mapping system—i.e., an LS-SVM model.

2. Train the LS-SVM model.
3. Enlarge the original image by CCI.
4. Estimate the high-frequency components using the trained LS-SVM model.
5. Add the estimated high-frequency components to the enlarged image by CCI.

In the following sections, the details of each step are described.

29.2.1 Generation of Training Data Set for an LS-SVM Model

The construction of a mapping system is a crucial part of the proposed image enlargement method. The mapping system is trained by using the training data set derived from self-decomposed images. In the generation of the training data set, a straightforward decomposition of an original image—i.e., to be an enlarged image—f ($N \times N$ pixels) into low- and high-frequency components is first performed. Suppose matrix L represents a low-pass operator, the original image f can be written as:

$$f = Lf + (I - L)f = f_l + f_h, \tag{29.1}$$

where I is the identity matrix, and f_l and f_h are the low- and high-frequency components of f, respectively. In this paper, f_l and f_h are referred to as self-decomposed images. If the original image f should be enlarged to M times size—i.e., the resultant image size is $NM \times NM$ pixels—the cutoff-frequency of the low-pass filter should be assigned to $1/M$ of maximum spatial frequency derived from the image size.

As shown in Fig. 29.1, in order to obtain training data set, f_l and f_h are decomposed to small patches that contain $w \times h$ pixels, respectively. w and h represent the width and height of each image block, respectively. The small patches obtained from f_l are denoted by $\mathbf{l}_i \in \mathbf{R}^d$, which composes a training input set. Correspondingly, the small patches obtained from f_h are denoted by $\mathbf{h}_i \in \mathbf{R}^d$, which composes a training target set. The dimensions of both $\mathbf{l}_i$ and $\mathbf{h}_i$ are equal to the number of pixels contained in a small patch—i.e., $d = w \times h$.

29.2.2 Training of an LS-SVM Model

To construct a map from low- to high-frequency components, an LS-SVM model for nonlinear function estimation is employed here. In the case of multiple target outputs, as this problem is, it is possible to treat the different output separately—that is, we can consider the model as d piece of LS-SVM models with d-input-one-output. Thus, we give a brief review on LS-SVM regression for a single output case.

Given a training set of n data points $\{\mathbf{x}_k, y_k\}_{k=1}^n$ on the Cartesian product space $\mathbf{X} \times \mathbf{Y}$, where $\mathbf{x}_k \in \mathbf{R}^d$, $y_k \in \mathbf{R}$. Speaking in the context of the proposed image enlargement process, $\mathbf{x}_k$ is a small patch $\mathbf{l}_k$, and y_k is an intensity of i-th pixel in

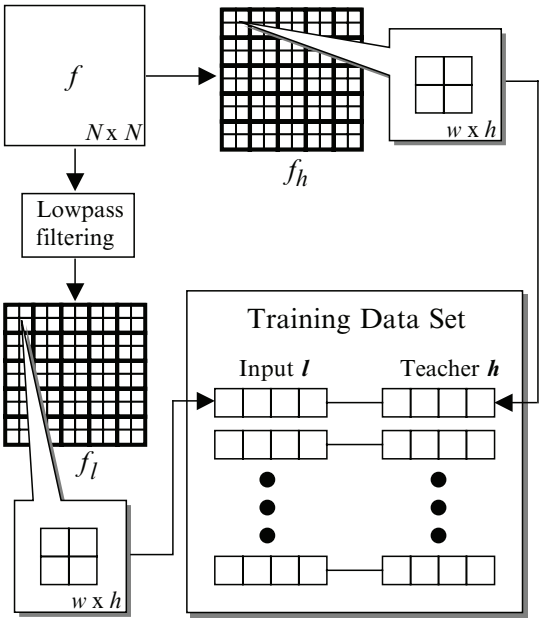


Fig. 29.1 Schematic diagram of a training data set generation from self-decomposed images

a small patch $\mathbf{h}_k$ if i-th pixel in a small patch $\mathbf{h}_k$ is estimated by the single output LS-SVM. Learning from the training data can be viewed as a multivariate function g approximation that represents the relationship between the input $\mathbf{X}$ and output $\mathbf{Y}$. Nonlinear mapping $\varphi(\cdot)$ is used to map input $\mathbf{X}$ into a hypothesis space $\mathbf{R}^{d_f}$ (feature space) in which the learning machine selects a certain function g. The feature space is the high dimensional and potentially infinite dimensional space.

Due to a generalized representer theorem [24], feature space endowed with an inner product is a function space with the linear structure of a vector space, in which linear function estimation is defined:

$$g(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b \tag{29.2}$$

In the case of LS-SVMs, the optimization problem is defined in the primal weight space

$$\min_{\mathbf{w},b,\mathbf{e}} \quad J(\mathbf{w},\mathbf{e}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma\frac{1}{2}\sum_{k=1}^{n} e_k^2, \tag{29.3}$$

$$s.t. \quad y_k = \mathbf{w}^T \varphi(\mathbf{x}_k) + b + e_k, \quad k = 1, \cdots, n, \tag{29.4}$$

where $e_k \in \mathbf{R}$ and γ denote the error and the regularization parameter, respectively. The regularization parameter γ is an arbitrary positive real constant. Note that this is, in fact, nothing else but a ridge regression [25] cost function formulated in the

feature space. However, one should be aware that when $\mathbf{w}$ becomes infinitely dimensional, one cannot solve this primal problem. Therefore, let us proceed by constructing the Lagrangian and derive the dual problem.

One constructs the Lagrangian

$$L(\mathbf{w},b,\mathbf{e};\alpha) = J(\mathbf{w},\mathbf{e}) - \sum_{k=1}^{n} \alpha_k \{\mathbf{w}^T \varphi(\mathbf{x}_k) + b + e_k - y_k\}, \tag{29.5}$$

where α_k are Lagrange multipliers. The conditions for optimality are given by

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \displaystyle\sum_{k=1}^{n} \alpha_k \varphi(\mathbf{x}_k), \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \displaystyle\sum_{k=1}^{n} \alpha_k, \\ \dfrac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k, \qquad\qquad k = 1,\cdots,n, \\ \dfrac{\partial L}{\partial \alpha_k} = 0 \rightarrow \mathbf{w}^T \varphi(\mathbf{x}_k) + b + e_k - y_k = 0, \quad k = 1,\cdots,n. \end{cases} \tag{29.6}$$

After elimination of the variables $\mathbf{w}$ and $\mathbf{e}$, one gets the following solution

$$\begin{bmatrix} \mathbf{0} & \mathbf{1}_n^T \\ \mathbf{1}_n & \mathbf{K} + \gamma^{-1}\mathbf{I}_n \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix}, \tag{29.7}$$

where $\mathbf{y} = [y_1,\cdots,y_n]^T$, $\mathbf{1}_n = [1,\cdots,1]^T$, $\alpha = [\alpha_1,\cdots,\alpha_n]^T$ and $K_{ij} = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$ for $i,j = 1,\cdots,n$. According to Mercer's theorem, the resulting LS-SVM model for function estimation becomes

$$\hat{g}(\mathbf{x}) = \sum_{k=1}^{n} \hat{\alpha}_k K(\mathbf{x}_k,\mathbf{x}) + \hat{b}, \tag{29.8}$$

where $\hat{\alpha}$, $\hat{b}$ are the solutions to the linear system Eqn. 30.17.

$$\hat{b} = \frac{\mathbf{1}_l^T \left(\mathbf{K} + \gamma^{-1}\mathbf{I}_l\right)^{-1} \mathbf{y}}{\mathbf{1}_l^T \left(\mathbf{K} + \gamma^{-1}\mathbf{I}_l\right)^{-1} \mathbf{1}_l} \tag{29.9}$$

$$\alpha = \left(\mathbf{K} + \gamma^{-1}\mathbf{I}_l\right)^{-1} \left(\mathbf{y} - \mathbf{1}_l \hat{b}\right). \tag{29.10}$$

As one of the most popular kernel functions in machine learning, Gaussian kernel function is selected for low- and high frequency mapping in this paper. It takes the following form

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{29.11}$$

where σ denotes the kernel parameter.

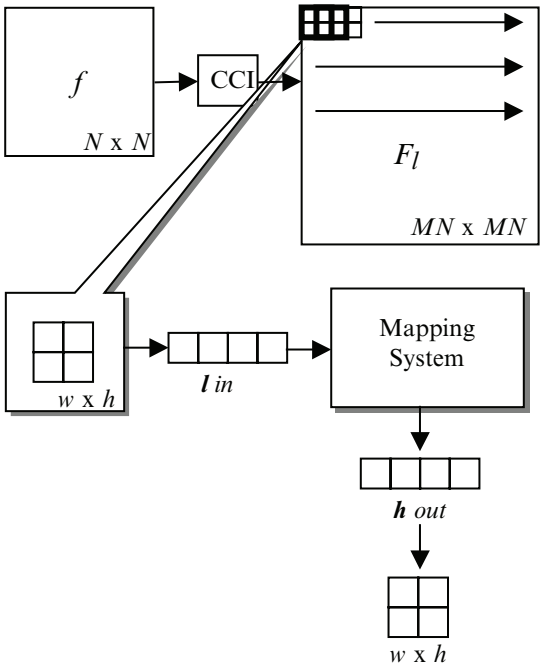


Fig. 29.2 Estimation of high-frequency components by the trained LS-SVM

29.2.3 Estimation of High-frequency Components by the Trained LS-SVM

After finishing the training of the LS-SVM model, the original image is enlarged by CCI to obtain the low-frequency component F_l. F_l is utilized as the source to derive the high-frequency components. To feed the F_l to the trained LSSVM, F_l is decomposed to $w \times h$ pixels per sliding image block, and then d-dimensional input vectors are obtained. Each obtained vector is fed to the trained LSSVM, and the high-frequency components for each small patch are obtained. This series of tasks is schematically explained in Fig. 29.2. In this manner, the estimation of high-frequency components is achieved thorough the mapping system.

29.2.4 Acquisition of Final Enlarged Image

The final enlarged image is obtained by adding the enlarged image by CCI and the estimated high-frequency components of the ideal enlarged image. Assuming the enlarged image by CCI could represent the low-frequency components of the ideal enlarged image, we can expect to get an enlarged image with edge-preserving and high performance.

29.3 Experimental Results

An attempt is made to verify the effectiveness and validity of the proposed algorithm by applying it to synthetic and natural gray-scale level images.

29.3.1 Synthetic Image

A synthetic image produced by the authors is used to show the superior characteristics of the proposed image enlargement method clearly.

This image consists of 64×64 pixels, and has two-tone, gray-level as shown in Fig. 29.3(a). Intensities of darker area and brighter areas are 50 and 150, respectively. Clear step-edge exists between the different tone areas. In the experiment, the original synthetic image is reduced to the scale of $\frac{1}{2}$ after anti-aliasing low-pass filtering. The resultant image shown in Fig. 29.3(b) is enlarged to the original size by the proposed image enlargement method and other comparable conventional methods—i.e., nearest neighbor interpolation (NNI), bilinear interpolation (BLI), cubic convolution interpolation (CCI), and nonlinear extrapolation (NE) [17]. NE has been proposed by Greenspan et al. and is a powerful image enlargement method based on the estimation of high-frequency components of the image.

Parameters of small patch size w and h are set as 4 and 4, respectively. Thus, the dimension of input vector to LS-SVM is 16. To quantitatively evaluate the enlargement performance in the methods mentioned above, mean square error (MSE) between the original image and the enlarged image by each method is calculated. The MSEs in these methods are shown in Table 29.1. Figure 29.4 shows the ideal image and the resulting image by each enlargement method. To give a detailed description of characteristics for each enlargement method, cross sections extracted

(a) (b)

Fig. 29.3 Synthetic image: (a) original size, and (b) $\frac{1}{2}$-reduced size after anti-aliasing filtering

Table 29.1 MSEs between the ideal image and enlarged image (synthetic image)

	NNI	BLI	CCI	NE	Proposed
MSE	140.4	81.5	65.9	53.3	51.2

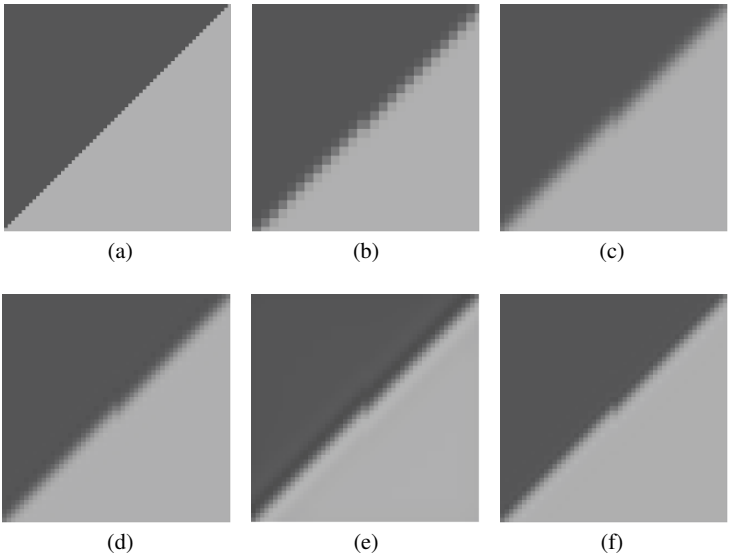


Fig. 29.4 Ideal image and enlarged images of the synthetic image: (a) ideal image, (b) NNI, (c) BLI, (d) CCI, (e) NE, and (f) proposed method

from the ideal image and the enlarged image by each method are shown in Fig. 29.5. From the cross-sectional diagrams, the following characteristics can be observed.

NNI: Stepped artifacts have appeared at the step-edge point in the original image.

BLI: The step-edge in the original image is blurred for the sake of loss of high-frequency components.

CCI: Indeed the blur of the step-edge has improved as compared with BLI, but the restoration of high-frequency components is still not enough.

NE: Under the favor of high-frequency components estimation, the step-edge is well-preserved in comparison to BLI and CCI. However, some degree of over-shooting in the estimation can be observed. Thus, the deterioration of such a ringing appeared in the image.

Proposed method: In both the step-edge preservation and the overshooting problem, the proposed method exhibits superior characteristics. The deterioration of such a ringing does not occur in the image enlarged by the proposed method. We can observe that the proposed method inhibits a ringing in comparison to NE.

29.3.2 Natural Gray-scale Level Images

The famous standard images Airplane, Boat, Building, Lenna, Text, and Woman included in the Standard Image Database (SIDBA) are used as original images. Each image consists of 256×256 pixels, and its resolution is 8 bit/pixel gray-level.

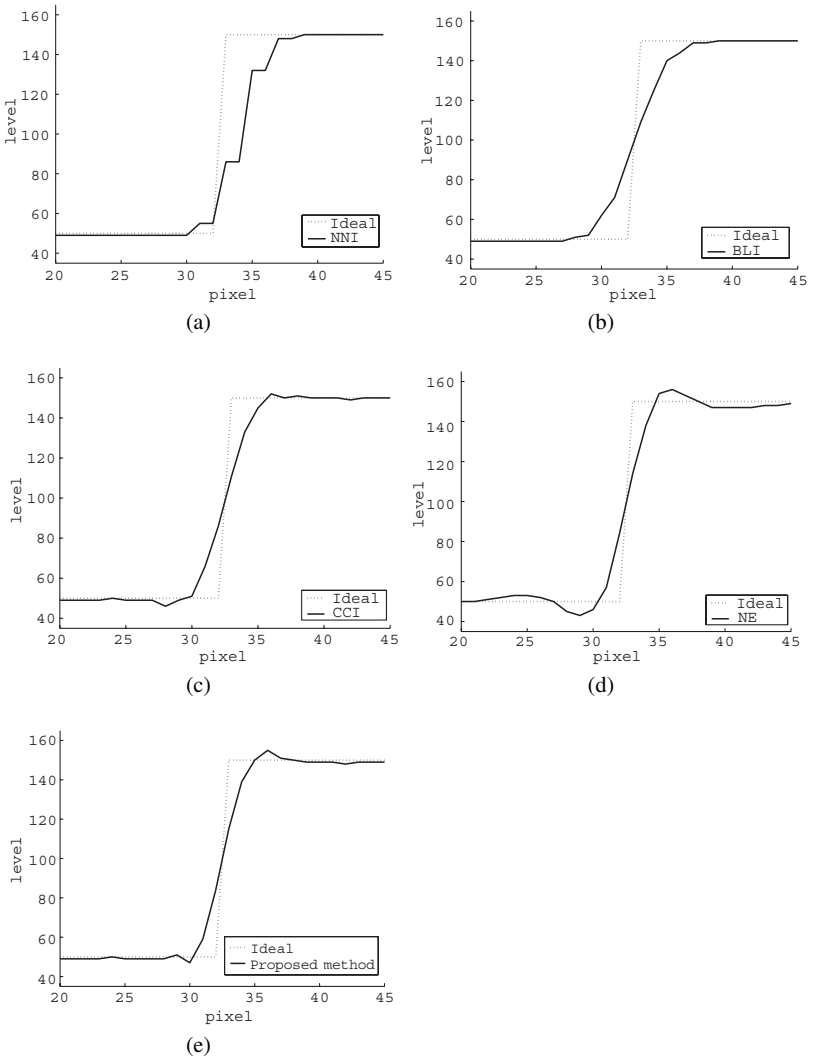


Fig. 29.5 Cross sections with respect to ideal image and the result of each method

The settings of experimental parameters and comparison with other conventional methods are the same as the case of synthetic image experiment. From Table 29.2, the superiority of the proposed method can be quantitatively confirmed with respect to MSE evaluations.

Table 29.2 MSEs between the ideal image and enlarged images

	NNI	BLI	CCI	NE	Proposed
Airplane	262.2	206.3	170.0	121.2	103.4
Boat	159.3	119.8	96.1	66.1	60.5
Building	224.9	179.4	146.3	100.3	87.2
Lenna	187.5	134.2	107.4	75.6	59.3
Text	443.2	321.1	254.2	173.1	146.9
Woman	163.0	121.9	100.8	77.1	67.9

29.4 Conclusion

In this paper, a new image enlargement method based on a mapping from the low- and high-frequency components—constructed by an original image itself that should be enlarged—proposed. Through experiments, it has been confirmed that the proposed method achieves superior enlargement in comparison to the conventional ones. The algorithm of the proposed method is also not as complicated. Thus, the proposed method has highly potential applications to a wide range of practical image enlargement.

Future works will accelerate the enlargement process by employing Fixed-Size LS-SVM [26], which is a modified version for large-scale problems, to efficiently extend the algorithm for color image enlargement.

Acknowledgements This work was supported by the Japan Society for the Promotion of Science under the Grant-in-Aid for Scientific Research (C) (No. 18500182): Development of Image Enlargement Method Based on Self-reduced Image and Vector Fuzzy Inference.

References

1. van Ouwerkerk JD (2006) Image super-resolution survey. Image and Vision Computing (24):1039–1052
2. Tsai RY and Huang TS (1984) Multiframe image restoration and registration. Advances in Computer Vision and Image Processing (1):317–339
3. Kaltenbacher E and Hardie RC (1996) High-resolution infrared image reconstruction using multiple low resolution aliased frames In *Proceedings of IEEE National Aerospace Electronics Conference*, New York, pp. 702–709
4. Kim SP, Bose NK, and Valenzuela HM (1990) Recursive reconstruction of high resolution image from noisy undersampled multiframes. IEEE Trans. Acoust. Speech Signal Process. (38):1013–1027
5. Oskoui-Fard P and Strak H (1998) Tomographic image reconstruction using the theory of convex projections. IEEE Trans. Med. Imaging (7):45–58
6. Patti AJ and Altunbasak Y (2001) Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolation. IEEE Trans. Image Process. (10):179–180

7. Schultz RR and Stevenson RL (1996) Extraction of high-resolution frames from video sequences. IEEE Trans. Image Process. (5):996–1011
8. Hardie RC, Barnard KJ, and Armstrong EE (1997) Joint MAP registration and high-resolution image estimation using a sequence of under-sampled images. IEEE Trans. Image Process. (6):1621–1633
9. Nguyen N, Milanfar P, and Golub G (2001) A computationally efficient super resolution image reconstruction algorithm. IEEE Trans. Image Process. (10):573–583
10. Lin JS (1990) Two-dimensional signal processing and image processing. (Prentice-Hall, London, UK)
11. Keys RG (1981) Cubic convolution interpolation for digital image processing. IEEE Trans. Acoust. Speech Signal Process. (29):1153–1160
12. Jensen K and Anastassiou D (1995) Subpixel edge localization and the interpolation still images. IEEE Trans. Image Process. (4):285–295
13. Li X and Orchard MT (2001) New edge-directed interpolation. IEEE Trans. Image Process. (10):1521–1527
14. Darwish AM and Bedair MS (1996) An adaptive resampling algorithm for image zooming. Proc. SPIE (2466):131–144
15. Martucci SA (1995) Image resizing in the discrete cosine transform domain. In *Proceeding of IEEE International Conference on Image Processing*. (Institute of Electrical and Electronics Engineers, New York) pp. 244–247.
16. Chang SG, Cvetkovic Z, and Vetterli M (1995) Resolution enhancement of images using wavelet transform extrema extrapolation. In *Proceeding of IEEE International Conference on Acoustic, Speech and Signal Processing*. (Institute of Electrical and Electronics Engineers, New York) pp. 2379–2382.
17. Greenspan H, Anderson CH, and Akber S (2000) Image enhancement by nonlinear extrapolation in frequency space. IEEE Trans. Image Process. (9):1035–1048
18. Aso T, Suetake N, and Yamakawa T (2006) A weighted linear extrapolation-based simple image enlargement algorithm. Intelligent Automation and Soft Computing V12(N3):345–353
19. Candonica FM and Principe JC (1999) Superresolution of images based on local correlations. IEEE Trans. Neural Netw. (10):372–380
20. Plaziac N (1999) Image interpolation using neural networks. IEEE Trans. Image Process. (8):1647–1651
21. Pan F and Zhang L (2003) New image super-resolution scheme based on residual error restoration by neural networks. Opt. Eng. (42):3038–3046
22. Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, and Vandewalle J (2002) Least Squares Support Vector Machines. World Scientific, Singapore
23. Vapnik VN (1995) The Nature of Statistical Learning Theory. Springer-Verlag, New York
24. Schölkopf B, Herbrich R, and Smola AJ (2001) A generalized representer theorem. Proceedings of 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory pp. 416–426
25. Golub GH and Van Loan CF (1989) Matrix Computations. Johns Hopkins University Press, Baltimore MD
26. Espinoza M, Suykens JAK, and Moor BD (2006) Fixed-size least squares support vector machines: a large scale application in electrical load forecasting. Journal of Computational Management Science 3(2):113–129

Chapter 30
Contour Descriptor Based on Affine Invariance Normalization

Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph

30.1 Introduction

The advent of multimedia and large image collections in different domains and applications brings with it a necessity for image retrieval systems. *Image retrieval* consists of the techniques used for query specification and retrieval of images from a digital image collection. It is considered part of the field of *Information Retrieval (IR)*, a large and mature research area. *IR* is the field that deals with "the representation, storage, organization and access to information items." The purpose of information retrieval is to provide the user with easy access to the information items of interest. Image retrieval has become an active research and development domain since the early 1970s [11]. During the last decade, research on image retrieval gained high importance. The most frequent and common means for image retrieval is to index images with text keywords. Although this technique seems to be simple, it rapidly becomes laborious when faced with large volumes of images. On the other hand, images are rich in content, and this can be exploited. So to overcome difficulties due to the huge data volume, *content-based image retrieval (CBIR)* emerged as a promising means for retrieving images and browsing large image databases.

CBIR is the process of retrieving images from a collection based on automatically extracted features. It is the part of pattern recognition that can help to resolve many problems, including optical character recognition (OCR), zip-code recognition, traffic sign recognition, bank check recognition, industrial parts inspection, medical image computer-aided diagnoses, retina recognition, iris recognition, face recognition, fingerprint recognition, palm recognition, document recognition, gait or gesture recognition, image indexing, etc.

Yang Mingqiang
IETR-INSA, UMR-CNRS 6164, Rennes, 35043, France
e-mail: myang@ens.insa-rennes.fr

Kpalma Kidiyo, Ronsin Joseph
IETR-INSA, Rennes, 35043, France

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

The main stages of image retrieval are *image description* and *similarity measurements*. According to MPEG-7, color, texture and shape can be used to represent images [8]. The use of shape as a visual cue in image retrieval is less developed than the use of color and texture, mainly because of the inherent complexity of representing it.

There are two main steps involved in shape feature extraction: *shape segmentation* and *shape representation*. In order to use shape as a tool for image indexing and retrieval, shapes must be identified in, and extracted from an image. *Shape segmentation* is considered to be one of the difficult aspects of image retrieval based on syntactical patterns—owing to the complexity of the individual object shape and background and the existence of noise and occlusion. Fortunately, there exists a large body of literature describing different techniques and approaches for shape segmentation, such as region growing, edge detection and texture-based techniques. One possible approach to automatic segmentation is to use a model-based approach. This makes use of pre-defined models as aids in identifying and segmenting shapes from the background. The shape segmentation is out of this chapter. *Shape description* is also a difficult problem. The main reason for this is the variety of ways in which a three-dimensional object can be projected into two-dimensional shapes. So the transformation of projection and affine is inevitable. Yet, shape and silhouette provide significant cues for human visual perception. This has been documented in a vast body of literature in cognitive psychology or biological neuroscience [7]. It is thus no surprise that a shape-based approach is popular in computer vision as well [10]. Thus retrieval by shape has the potential of being the most effective search technique in many application fields.

A shape is originally defined by the x and y coordinates of its boundary points and is subject to change: If the camera is allowed to change its viewpoint with respect to the object, the resulting boundary of the object will be deformed. This deformation can be approximated by general *affine transforms*. As a result, a shape description must be robust under similarity transformation, which includes scaling, changes in orientation, shearing and translation. Numerous shape descriptions have been suggested to recognize shapes even under affine transforms. Some of them are the extensions of well-known methods such as dyadic wavelet transform [4], Fourier descriptors [5], affine curvature scale space (CSS) [6], affine arc length [3] and moment invariants [2, 12], etc. In these methods, the basic idea is to use a parameterization that is robust with respect to affine transforms.

In this chapter, to extract an affine invariant attribute, a new algorithm based on equal area normalization is proposed. In theory this attribute is absolutely robust under affine transforms, even with serious transformations. Experimental results show that the proposed algorithm is also rather insensitive to noise. In addition, a high computationally efficient object recognition scheme is briefly presented.

30.2 Fundamental Concepts

The *shape* of a physical object is its external form or contour, the geometry of its external surfaces or contours, the boundary between the objects interior and the exterior. Shape is the outline or characteristic surface configuration of the object. We can use the *region shape* or *contour shape* to extract the attributes of images.

The *region shape* of an object may consist of either a single region or a set of regions as well as some holes in the object. Since the region-based shape descriptor makes use of all pixels constituting the shape within a frame, it can describe diverse shapes efficiently in a single descriptor. It is also robust to minor deformation along the boundary of the object. The *contour shape* of an object is the exterior boundary of the object; it is a continuous curve in the plane. The contour shape descriptor captures characteristic shape features of an object based on its contour. In general, a contour shape descriptor is simpler with fewer computation complexities than a region shape descriptor; it's not reliable and not very resistant to noise.

Efficient shape features must present some essential properties such as:

- *Translation, rotation* and *scale invariance*: The location, rotation and scale of the shape changing must not affect the extracted features.
- *Affine invariance*: The affine transform is achieved by looking at a planar object from different view angles. The affine transform performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the "straightness" and "parallelism" of lines. The affine transform can be constructed using sequences of translations, scales, flips, rotations, and shears. The extracted features must change as little as possible with affine transforms.
- *Noise resistance*: Features must be as robust as possible against noise, i.e., they must be the same regardless of the strength of the noise that affects the pattern.
- *Occlusion invariance*: When some parts of a shape are occluded by other objects, the feature of the remaining part doesn't change compared to the original shape.
- *Statistically independent*: Two features must be statistically independent.
- *Reliable*: as long as one deals with the same pattern, the extracted features must remain the same.

Good *retrieval accuracy* requires a shape descriptor able to effectively find perceptually similar shapes from a database. *Shape descriptors* are mathematical and/or statistical descriptions of syntactical shape features. Usually, the descriptors are in the form of a vector. Shape descriptors should meet the following requirements:

- The descriptors should be as complete as possible to represent the content of the information items.
- The descriptors should be represented and stored compactly. The number of a descriptor vector must not be large.
- The computation of distance between descriptors should be efficient; otherwise the execution time would be too long.

So an efficient shape descriptor should have the following properties: accessibility, large scope, uniqueness, stability and sensitivity. *Accessibility* describes how

easy (or difficult) it is to compute a shape descriptor in terms of memory requirements and computational time. *Scope* demonstrates the class of shapes that can be described by the method. *Uniqueness* describes whether a one-to-one mapping exists between shapes and shape descriptors. *Stability and sensitivity* describe how sensitive a shape descriptor is to "small" changes in shape.

Many shape-description and similarity measurement techniques have been developed in the past, and number of new techniques have been proposed in recent years. According to processing contents in the shape, shape representation and description techniques can be generally classified into three classes: contour-based methods, region-based methods and space-based methods. The classification is based on whether shape features are extracted from the contour only, are extracted from the whole shape region or are extracted from the main part of the shape. Under each class, the different methods are further divided into structural approaches and global approaches. This sub-class is based on whether the shape is represented as a whole or represented by segments/sections. These approaches can be further distinguished into space domain and transform domain, based on whether the shape features are derived from the spatial domain or from the transformed domain.

The method of the affine invariance contour descriptor is based on contours. Before introducing the approach, we first presented some fundamental concepts about the contours and affine transforms.

– A. Closed Curve
  Let us consider the discrete parametric equation of a closed curve:

$$\Gamma(\mu) = (x(\mu),\ y(\mu)) \tag{30.1}$$

  where $\mu \in \{0, \ldots, \mathrm{N}-1\}$, an application may have been parameterized with any number of vertices N.
– B. Affine Transforms
  The affine transformed version of a shape can be represented by the following equations:

$$\begin{bmatrix} x_a(\mu) \\ y_a(\mu) \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x(\mu) \\ y(\mu) \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} + B \tag{30.2}$$

  where $x_a(\mu)$ and $y_a(\mu)$ represent the coordinates of the transformed shape. Translation is represented by matrix B, while scaling, rotation and shear are reflected in the matrix A. They are represented by the following matrices:

$$A_{Scaling} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix},\ A_{Rotation} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},\ A_{Shear} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

  If S_x is equal to S_y, $A_{Scaling}$ represents a uniform scaling. A shape is not deformed under rotation, uniform scaling and translation. However, non-uniform scaling and shear contribute to the shape deformation under general affine transforms.
– C. Affine Invariant Parameters
  There are two parameters that are linear under affine transforms. They are the *affine arc length*, and the *enclosed area* [9].
  The first parameter can be derived based on the properties of determinants. It is defined as follows:

$$\tau = \int_a^b [x'(\mu)y''(\mu) - x''(\mu)y'(\mu)]^{1/3} d\mu \tag{30.3}$$

The second affine invariant parameter is enclosed area, which is derived based on the property of affine transforms: Under affine mapping, all areas are changed in the same ratio. Based on this property, Arbter et al. [5] defined a parameter σ that is linear under a general affine transform, as follows:

$$\sigma = \frac{1}{2} \int_a^b \left| x(\mu)y'(\mu) - y(\mu)x'(\mu) \right| d\mu \tag{30.4}$$

where $x(\mu)$ and $y(\mu)$ are the coordinates of points on the contour with the origin of system located at the centroid of the contour. The parameter σ is essentially the cumulative sum of triangular areas produced from connecting the centroid to a pair of successive vertices on the contour.

30.3 Equal Area Normalization

All points on a shape contour could be expressed in terms of the parameter of index points along the contour curve from a specified starting point. With the affine transforms, the position of each point changes, and it is possible that the number of points between two specified points changes too. So if we parameterize the contour using the equidistant vertices, the index point along the contour curve will change under affine transforms.

For example, Fig. 30.1a is the top viewpoint of a plane, and Fig. 30.1e is the back top viewpoint of the plane, so Fig. 30.1e is one of the affine transforms of the image Fig. 30.1b and Fig. 30.1f. Fig. 30.1c and Fig. 30.1g are the parts of the contour Fig. 30.1b and Fig. 30.1f normalized by equal distance, respectively. In Fig. 30.1g. So, the contour normalized by equidistant vertices is variance to possible affine transforms.

To have invariance under affine transforms, a novel curve normalization approach is proposed that provides an affine invariant description of object curves at a low computational cost, while at the same time preserving all information on curve shapes. This approach is equal area normalization (EAN), and we present it as follows:

All points on a shape contour could be expressed in terms of two periodic functions $C(\mu') = (x(\mu'), y(\mu'))$, where variable μ' is measured along the contour curve from a specified starting point.

1) Normalize the contour to N points with equidistance vertices. The new periodic functions are $C(\mu) = (x(\mu), y(\mu))$, and all the points on the contour are p_μ. where $\mu \in [0, N-1]$.
2) Calculate the second-order moments of the contour as its centroid G.
3) Transfer the contour to make centroid G be at the origin of coordinates.
4) The last point $(x(N), y(N))$ is assumed to be the same as the first. Compute the area of the contour using the formula:

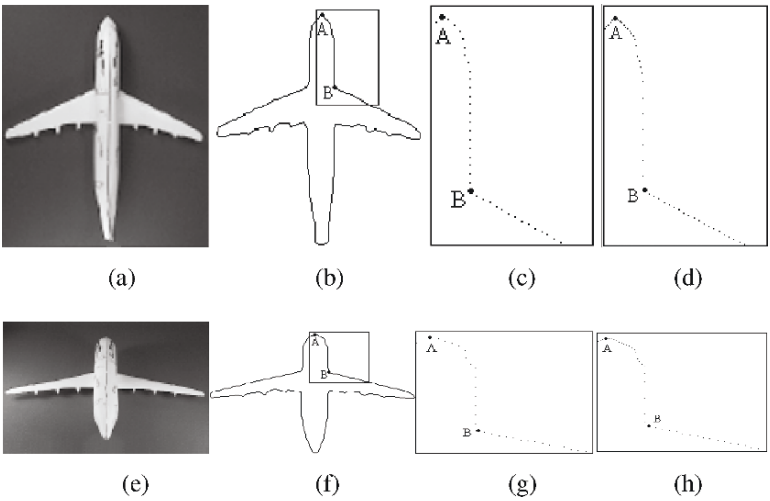


Fig. 30.1 The comparison of equidistant vertices normalization and equal area normalization. **a** is the image of top viewpoint of a plane. **b** is the contour of image **a**. **c** is a part of contour **b** normalized by equidistant vertices. **d** is a part of contour **b** normalized by equal area. **e** is the image of back top viewpoint of the plane. **f** is the contour of image **e**. **g** is a part of contour **f** normalized by equidistant vertices. **h** is a part of contour **f** normalized by equal area

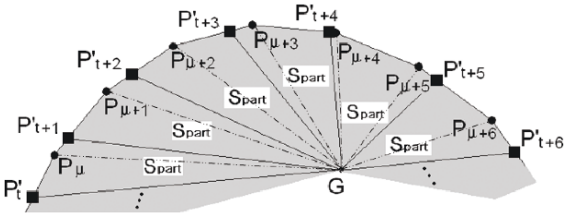


Fig. 30.2 The method of normalization equal area. "•" is the vertex p of equidistant vertices normalization, and "■" is the point p' of equal area normalization. G is the centroid of the contour

$$S = \frac{1}{2}\sum_{\mu=0}^{N-1} |\, x(\mu)y(\mu+1) - x(\mu+1)y(\mu)\,| \tag{30.5}$$

where $\frac{1}{2}|x(\mu)y(\mu+1) - x(\mu+1)y(\mu)|$ is the area of the triangle whose vertices are $p_\mu\ (x(\mu),\ y(\mu))$, $p_{\mu+1}\ (x(\mu+1),\ y(\mu+1))$ and centroid G (cf. Fig 30.2).

5) Let the number of points on the contour after EAN be N too. Of course, any other number of points could be chosen. Therefore, after EAN, each enclosed area S_{part} defined by the two successive points on the contour and centroid equals $S_{part} = S/N$.
6) Suppose all the points on the contour after EAN are p'_t. Let $C(t) = (x'(t), y'(t))$ represent the contour, where $t \in [0, N-1]$. Choose the starting point p_0 $(x(0),\ y(0))$ on the equidistant vertices normalization as the starting point $p_0{}'$

$(x'(0), y'(0))$ of the EAN, on the segment p_0p_1, we seek for a point $p_1'(x'(1), y'(1))$, let the area $s(0)$ of the triangle whose vertices are $p_0'\ (x'(0), y'(0)), G(0)$ and $p_1'(x'(1), y'(1))$ equal to S_{part}. If there is not a point to satisfy the condition, then we look for the point p_1' on the segment p_1p_2. So the area $s(0)$, which is the sum of the areas of triangle p_0Gp_1 and triangle p_1Gp_1', is equal to S_{part}. If there is not a point to satisfy the condition yet, we continue to seek for this point at the next segment until the condition is satisfied. This point p_1' is the second point on the normalized contour.

7) From point $p_1'(x'(1), y'(1))$, we use the same method to calculate all the points $p_t'(x'(t), y'(t))$, $t \in [0, N-1]$ along the contour. Because the area of each closed zone, for example, the polygon $p_t'[p_\mu p_{\mu+1} \cdots]p_{t+1}'G\ t \in [0, N-2]$ equal to S_{part}, the total area of $N-1$ polygon is $(N-1) \cdot S_{part}$. According to 5), we know it equal to $S - S_{part}$. So the area of the last zone $p_{N-1}'Gp_0'$ equals S_{part} exactly.

From Fig. 30.2 we know, the area of the triangle $p_t'Gp_{t+1}'$ is equal to the area S_t of polygon $p_t'[p_\mu p_{\mu+1} \cdots]p_{t+1}'G$ approximately if the distance between the two points $p_\mu p_{\mu+1}$ is close enough or the number N of the points on the contour is big enough. Therefore, we can use the points $p_t'\ t \in [0, N-1]$ to replace the points $p_\mu\ \mu \in [0, N-1]$. Then the process of EAN is accomplished.

According to subsection 2.C, after this normalization, the number of vertices between the two appointed points on a contour is invariant under affine transforms. Fig. 30.1d and Fig. 30.1h are the same parts of Fig. 30.1c and Fig. 30.1g, respectively. We notice that the distance between the consecutive points is not uniform. In Fig. 30.1d, the number of points between the appointed points A and B is 23; the number is also 23 in Fig. 30.1g. Therefore, after applying EAN, the index of the points on a contour can maintain stability with their positions under affine transforms. This property will be very advantageous to extract the robust attributes of a contour and decreasing complexity in the similarity measure. We can also apply EAN to the other algorithms to improve their robustness with affine transforms. For example, before doing the algorithm of curvature scale space (CSS) [1], the contour is normalized by EAN. All of the maximum points in the image CSS will not change under affine transforms. This is beneficial to calculate the similarity between two CSS attributes.

In the following, we will study the part area S_{part} change with the effect of affine transforms and filtering.

30.4 Normalized Part Area Vector

In this section, we look for the existing relations between the part area S_{part}, affine transforms and low-pass filtering.

Theorem 30.1: *If $\Gamma_a(x_a(\mu), y_a(\mu))$ is the transformed version of a planar curve $\Gamma(x(\mu), y(\mu))$ under an affine transform A, where μ is an arbitrary parameter, $\Gamma_{af}(x_{af}(\mu), y_{af}(\mu))$ notes that Γ_a is filtered by a linear low-pass filter F;*

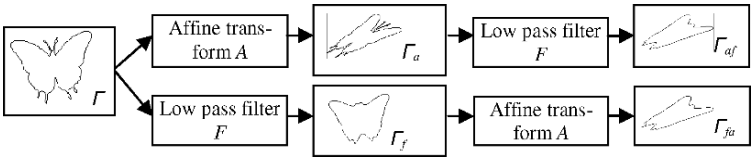


Fig. 30.3 Demonstration of Theorem 1

and if $\Gamma_f(x_f(\mu),\ y_f(\mu))$ *notes that* Γ *is filtered by the same low-pass filter* F, $\Gamma_{fa}(x_{fa}(\mu),\ y_{fa}(\mu))$ *refers to the transformed version of* Γ_f *under the same affine transform* A. *Planar curve* Γ_{af} *is then the same as planar curve* Γ_{fa}. *In other words:* $F[A(\Gamma)] = A[F(\Gamma)]$. (*cf. Fig. 30.3*).

Proof: From Eq. (30.2) we have

$$x_a(\mu) = ax(\mu) + by(\mu) + e \tag{30.6}$$

$$y_a(\mu) = cx(\mu) + dy(\mu) + f \tag{30.7}$$

Translation is represented by e and f. For all affine transform contours, we transfer the center gravity to origin. So the represents of translation e and f can be removed. Therefore, the affine transform can be represented by two simple formulae:

$$x_a(\mu) = ax(\mu) + by(\mu) \tag{30.8}$$

$$y_a(\mu) = cx(\mu) + dy(\mu) \tag{30.9}$$

The computation starts by convolving each coordinate of the curve $\Gamma_a(\mu)$ with a linear low-pass filter F whose impulse response is $g(\mu)$. In continuous form we have

$$\begin{aligned} x_{af}(\mu) &= x_a(\mu) * g(\mu) = [ax(\mu) + by(\mu)] * g(\mu) \\ &= ax(\mu) * g(\mu) + by(\mu) * g(\mu) = ax_f(\mu) + by_f(\mu) \end{aligned} \tag{30.10}$$

where $*$ denotes convolution. Likewise,

$$y_{af}(\mu) = cx_f(\mu) + dy_f(\mu) \tag{30.11}$$

Compare Eqs. (30.10), (30.11) and Eqs. (30.8), (30.9); it is clear that point $(x_{af}(\mu),\ y_{af}(\mu))$ is also the point $(x_f(\mu),\ y_f(\mu))$ transformed by the affine transform A. So the planar curve Γ_{af} *i*s same as the planar curve Γ_{fa}.

Theorem 1 indicates that after exchanging computation order between affine transform and filtering, there is no change in the result.

Theorem 30.2: *For any affine transform of a closed contour, using EAN to set the parameters produces planar curve* $\Gamma_a(x_a(t),\ y_a(t))$. *If area* $s_p(t)$ *is the area of an*

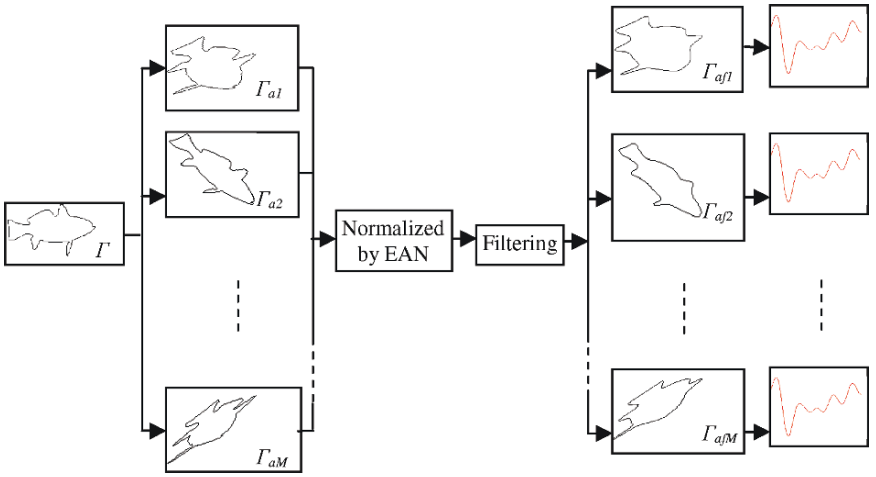


Fig. 30.4 Demonstration of Theorem 2

enclosed sector whose vertices are a pair of successive points and the centroid of the contour, and if $\Gamma_{af}(x_{af}(t),\ y_{af}(t))$ *indicates that* Γ_a *is filtered by a low-pass filter* F*, then the changes in enclosed areas* $s_p(t)$ *on the* Γ_{af} *are linear with affine mapping. See Fig. 30.4.*

Proof: In Sect. 30.3, we saw that the enclosed area $s_p(t)$ of the triangle on the contour filtered whose vertices are $(x_{af}(t),\ y_{af}(t))$, $(x_{af}(t+1),\ y_{af}(t+1))$ and the centroid G is

$$s_p(t) = \frac{1}{2} abs[x_{af}(t)y_{af}(t+1) - x_{af}(t+1)y_{af}(t)] \tag{30.12}$$

In the following, we will show that $s_p(t)$ is absolutely affine invariant to affine transforms. Due to Theorem 1,

$$x_{af}(t) = ax_f(t) + by_f(t) \tag{30.13}$$

$$y_{af}(t) = cx_f(t) + dy_f(t) \tag{30.14}$$

and

$$x_{af}(t+1) = ax_f(t+1) + by_f(t+1) \tag{30.15}$$

$$y_{af}(t+1) = cx_f(t+1) + dy_f(t+1) \tag{30.16}$$

Therefore from Eq. (30.12)

$$\begin{aligned} s_p(t) =& \frac{1}{2} abs\{[ax_f(t)+by_f(t)][cx_f(t+1)+dy_f(t+1)] \\ & -[ax_f(t+1)+by_f(t+1)][cx_f(t)+dy_f(t)]\} \\ =& \frac{1}{2} abs[adx_f(t)y_f(t+1)+bcx_f(t+1)y_f(t)-adx_f(t+1)y_f(t) \\ & -bcy_f(t+1)x_f(t)] \\ =& \frac{1}{2} abs(ad-bc)\cdot abs[x_f(t)y_f(t+1)-x_f(t+1)y_f(t)] \end{aligned} \tag{30.17}$$

Observe that Eq. (30.17), $s_p(t)$ is just linearly proportional by a scale factor *abs(ad-bc)*. Accordingly, we have proved that enclosed areas $s_p(t)$ are linear with affine mapping.

DEDUCTION The proportion $v'(t)$ of the closed areas $s_p(t)$ with the total area S of the filtered contour is preserved under general affine transforms.

Proof: According to Eq. (30.17), the total area S of the filtered contour is:

$$S = \frac{1}{2} abs(ad-bc)\cdot \sum_{t=0}^{N-1} abs[x_f(t)y_f(t+1)-x_f(t+1)y_f(t)] \tag{30.18}$$

So

$$\begin{aligned} v'(t) &= s_p(t)/S \\ &= abs[x_f(t)y_f(t+1)-x_f(t+1)y_f(t)]/ \\ &\quad \sum_{t=0}^{N-1} abs[x_f(t)y_f(t+1)-x_f(t+1)y_f(t)] \end{aligned} \tag{30.19}$$

Observe that Eq. (30.19), $v'(t)$ doesn't relate with the affine parameters *a,b,c* and *d*. Therefore $v'(t)$ is preserved under general affine transforms.

In addition, we can deduct major property of $v'(t)$: the integration of $v(t) = [v'(t) - 1/N]$ equal to zero.

Proof:

$$\sum_{t=0}^{N-1} v(t) = \sum_{t=0}^{N-1} [v'(t) - \frac{1}{N}] = \sum_{t=0}^{N-1} \frac{s_p(t)}{S} - \sum_{t=0}^{N-1} \frac{1}{N} = \frac{S}{S} - \frac{N}{N} = 0$$

We refer to vector $v(t)$ as the normalized part area vector(NPAV). Fig. 30.5 is an example of NPAV. The contour is normalized to 512 points by EAN.

As Theorem 30.2 and its deduction show, in all cases, even those with severe deformations, the function $s_p(t)$ is also preserved. Only the amplitude changes under general affine transforms; the NPAV $v(t)$ has an affine-invariant feature. In the following section, we present the results of our experiments, which evaluate the property of the proposed algorithm.

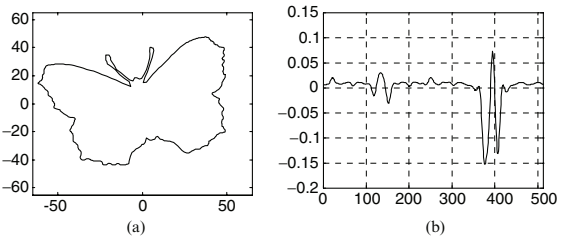


Fig. 30.5 An example of NPAV. **(a)** The contour of a butterfly. **(b)** The NPAV of the contour of butterfly

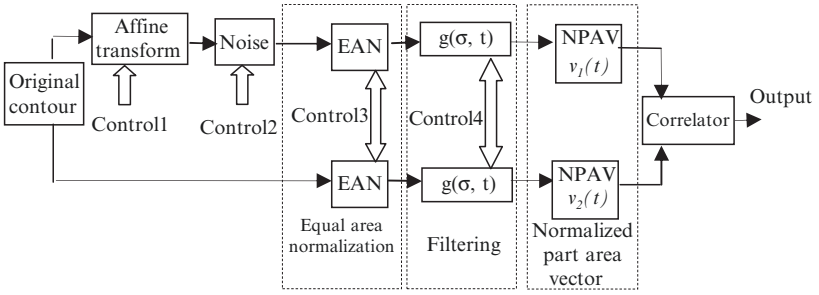


Fig. 30.6 The framework of experiments

30.5 Experimental Results and Multiscale NPAV

In experimental results, we will observe NPAV behavior in relation to affine transforms, EAN parameterization, filtering and noise. We experiment on the MPEG-7 CE-shape-1 database containing 1400 shape images. The framework of these experiments can be seen in Fig. 30.6.

Here, the low-pass filter is a Gaussian filter with a scale parameter σ. The evaluation output is the maximum linear correlation coefficient between the NPAV of the upper pathway after affine transforms, noise, EAN parameterization and filtering and the NPAV of the lower pathway with the original contour.

First, we evaluate the effect of affine transforms on NPAV. Let the number of normalized points be 512, the position of the starting point be the same as the original contour and $\sigma = 10$. Accordingly, if the range of uniform scaling transforms and shearing transforms changes from 0.1 to 10, and the respective rotation angles are 60°, 120°, 180°, 240° and 300°, then all the linear correlation coefficients on NPAV have values above 97 percent.

We then assess the effect of equal area normalization (EAN) on NPAV. This experiment includes two aspects: on one hand the relation between NPAV and the number of points normalized by EAN and on the other hand the relation between NPAV and the position of the starting points on a contour. The results show that for a number of points on a contour normalized to 64, 128 and 256, and simultaneously

with a starting point position displaced to 12.5, 25, 37.5 and 50 percent of the index points along the contour, all the linear correlation coefficient values stay above 97 percent.

Finally, we study the effect of noise on NPAV. When we resample the edges of an object, the original shapes are affected by noise and impairments, generating fluctuations on the boundary. The general method for noise simulation on a contour is to shift the x, y coordinates of the points on a contour independently. In this way, the order of certain points on the contour becomes confused (cf. Fig. 30.7a). However, in practice, such distortion cannot be present. In order to simulate the real effect of noise on an object, the displacement direction of each point affected by noise is controlled so that it is only vertical to the tangent of the point. Fig. 30.7b shows the effect of noise simulated with the proposed method, i.e., there is a significant decrease in self-intersections.

If its amplitude is controlled by a uniform random value, suppose the amplitude range is $[-r, r]$, and the average distance between all the points on the contour and its centroid is D, we define the signal-to-noise ratio (*SNR)* as follows:

$$\mathrm{SNR} = 20\lg\frac{2\mathrm{D}}{\mathrm{r}}(\mathrm{dB})$$

We performed the experiments in such a way that the test contours were contaminated by random uniform noise ranging from high to low SNRs (cf. Fig. 30.8a–d).

Table 30.1 shows the average correlation coefficient of all the NPAVs of shapes in the database with different *SNRs*. Correlation between NPAVs remains at an adequate level to allow efficient pattern recognition.

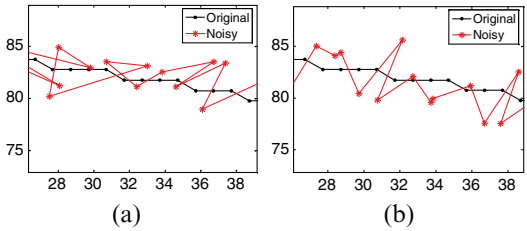


Fig. 30.7 Two effects of adding simulated noise. **(a)** is the effect of the general method of simulation. **(b)** is the effect of the proposed method

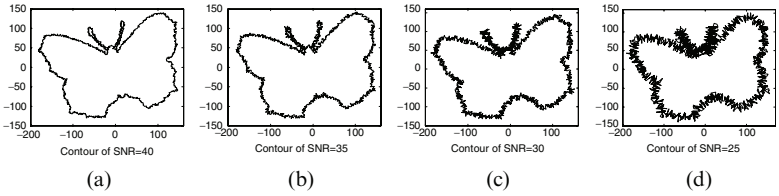


Fig. 30.8 Demonstration of a contour contaminated by various SNRs

Table 30.1 The average correlation coefficient of different SNR

SNR	Correlation Coefficient
40dB	0.96366
35dB	0.96298
30dB	0.94907
25dB	0.89765

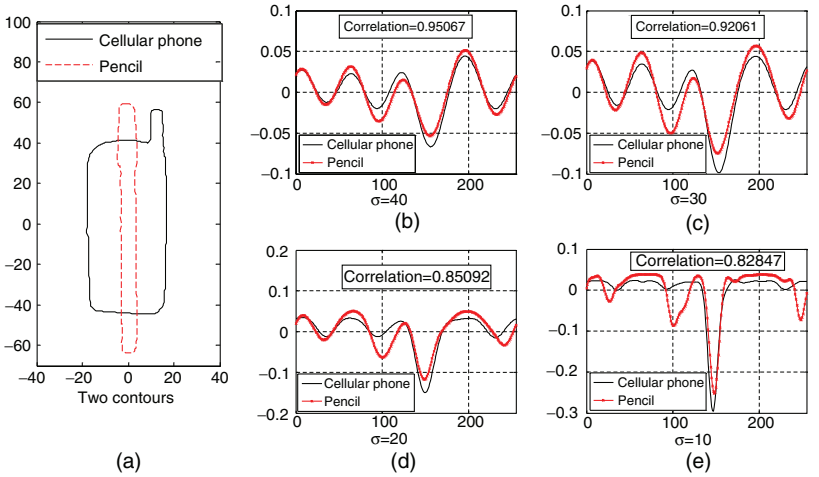


Fig. 30.9 NPAV and the scale. **a** is two original contours. **b–e** are their NPAVs with different scales

By analyzing the results of experiments, we notice that NPAV is quite robust with respect to scale, orientation changes of objects, shearing, the position of the starting point and noise. Therefore, NPAV can be used to characterize a pattern for recognition purpose. In the following, we propose an object recognition scheme based on multiscale technology.

Under the high scale, it is possible that some of different objects have a similar NPAV. However the NPAVs of these objects probably have different features under low scale. Fig. 30.9 shows an example; the two objects in **a** are normalized by EAN to 256 points. Fig. 30.9b–e are their NPAVs under the scales $\sigma = 40$, $\sigma = 30$, $\sigma = 20$ and $\sigma = 10$, respectively. As can be seen in the figures, the higher the scale is, the more similar they are. By contrast, the lower scale is, the more different they are. Thus we can present a shape with different scales: high-scale NPAV can be used to eliminate dissimilar shapes, and low-scale images can discriminate between similar shapes.

In addition, as the NPAV expresses a contour with all the points normalized by EAN, it is rather redundant. To decrease the redundancy, we apply the wavelet algorithm to compress the NPAV. We have noticed that the higher the value of σ, the less high frequency remains in the NPAV; By contrast, the lower the value of σ,

the more high frequency remains in it. Thus we can use the high compression ratio to NPAV with higher σ, and low compression ratio to that with lower σ. In this way, a high computationally efficient object recognition scheme can be designed.

30.6 Conclusion

In this chapter, a new method of extracting invariants of an image under affine transform is described. Our representation is based on the association of two parameters: the affine arc length and enclosed area, viz., we normalize a contour to invariant length by the affine enclosed area. We then prove two theorems and a deduction. They reveal that for a filtered contour, the part area is linear under affine transforms. We further define an affine-invariance vector: normalized part area vector (NPAV). A number of experiments on the MPEG-7 database demonstrate that NPAV is quite robust with respect to affine transforms and noise, even in the presence of severe noise. In addition, we propose a method to simulate the noise contaminating the test shapes and define the signal-to-noise ratio for a shape. Finally, a shape recognition scheme is briefly presented.

Future directions of the research include doing more extensive tests to assess retrieval results by multiscale NPAV, ameliorating the retrieval performance of the NPAN under part of the shape occluded.

References

1. Jalba A, Wilkinson M, Roerdink J (2006) Shape representation and recognition through morphological curvature scale spaces. IEEE Trans. Image Processing 15(2):331–341
2. Zhao A, Chen J (1997) Affine curve moment invariants for shape recognition. Pattern Recognition 30(6): 895–901
3. Cyganski D, Vaz RF (1991) A linear signal decomposition approach to affine invariant contour identification. In: Proc. of SPIE-Intelligent Robots and Computer Vision X: Algorithms and Techniques 1607:98–109
4. Bala E, Enis Cetin A (2004) Computationally efficient wavelet affine invariant functions for shape recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 26(8):1095–1099
5. Arbter K, Snyder WE, Burkhardt, H, Hirzinger G (1990) Applications of affine-invariant Fourier descriptors to recognition of 3-D objects. IEEE Trans. Pattern Analysis and Machine Intelligence 12(7): 640–646
6. Mokhtarian F, Abbasi S (2002) Shape similarity retrieval under affine transform. Pattern Recognition 35(1): 31–41
7. Tarr MJ, Bulthoff HH (1998) Image-based object recognition in man, monkey, and machine. Cognition 67:1–20
8. MPEG-7 Overview (Version 10) (2004), ISO/IEC JTC1/SC29/WG11
9. Tieng QM, Boles WW (1997) Wavelet-based affine invariant representation: A tool for recognizing planar objects in 3D space. IEEE Trans. Pattern Analysis and Machine Intelligence 19(8):846–857

10. Loncaric S (1998) A survey of shape analysis techniques. Pattern Recogntion 31(8):983–1001
11. Sikora T (2001) The MPEG-7 visual standard for content description—an overview. IEEE Tran. On Circuits and Systems for Video Technology 11(6):696–702
12. Zhang DS, Lu G (2004) Review of shape representation and description techniques. Pattern Recognition 37:1–19

Chapter 31
Adaptive Difference of Gaussian for Speed Sign Detection in Night-time Conditions

Tam Phuong Cao and Darrell M. Elton

31.1 Introduction

Road safety is gaining more public attention nowadays. Safety technologies, such as air bags or antilock-braking systems (ABS), have been providing a much higher level of safety to road traffic participants. Intelligent vehicles that can actively help to prevent accidents are also being investigated and developed. One area of research is in speed sign detection systems that alert the driver to the current speed limit.

Vision-based systems that are able to detect and classify speed signs have been under investigation since the late 1980s [1]. Many techniques have been explored, including artificial neural network (ANN) [2, 3], Support Vector Machine (SVM) [4], gradient-based detection [5–7], geometric labeling [8] and signature (FFT) matching [9]. Most of those algorithms perform well in a limited range of lightning conditions, which are normally restricted to daytime lighting conditions. In order to be reliably integrated into an automobile, any vision system needs to be able to run accurately and in real-time regardless of weather and lighting conditions.

Detecting speed signs in nighttime driving conditions is a challenging task. The major reasons for this are the low contrast of the objects under low illumination and distortions introduced by the camera used in the vision system. Generally, a camera reduces its shutter speed and/or increases its sensitivity (ISO) index to record images in low light. As a result, the noise level in the images increases significantly. The term noise here refers to the graininess of the image due to high ISO, motion blur due to slow shutter speed, low contrast due to under-illuminated objects, saturation due to direct head lights and color distortions (in the case of color camera used). In modern video cameras, distortion may be reduced by the prudent use of the gain control (which amplifies both signal and noise) and high dynamic range mode (which reduces the over-saturation problem but compresses information at bright

Tam Phuong Cao and Darrell M Elton
Department of Electronic Engineering, Latrobe University, Australia
e-mail: {t.cao,D.Elton}@latrobe.edu.au

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

pixel levels). Cameras with those features are still expensive and not at the mature stage. Reducing motion blur is still a real challenge for these cameras.

The combined effects of these noise sources result in a small number of frames usable for detection in a video sequence. We do not attempt to correct any of the mentioned distortions in the image. Instead we try to increase the number of usable frames by compensating for frames that have relative low contrast and a certain level of motion blur without causing significant false detection and/or classification. Even though many of the detected signs are not classifiable; i.e., the text within the sign is not human readable, it is still beneficial to keep track of the location of the sign when it appears in the image sequence.

This chapter is organized in five sections. This introduction is followed by a brief introduction on background and related research in Sect. 31.2. Gaussian and difference of Gaussian filters will be introduced in Sect. 31.3. In Sect. 31.4, a reference speed sign detection algorithm called Radial Symmetry will be described together with experimental results when using adaptive DoG with this algorithm. Finally in Sect. 31.5 is a short conclusion.

31.2 Background and Related Research

Various approaches have been proposed for night-time operation of vision systems, including the use of an infrared camera [10], adaptive segmentation [11], and using an illumination robust color scheme [7].

In [10], input images for the algorithm were acquired from an infrared camera. This is a common method for using vision systems at night. However, this is not suitable for detecting speed signs as well as other road signs because they are cold objects, and hence have very small infrared signatures. Limited range and resolution are other potential problems for using an infrared camera in a vision-based system.

Bin [11] proposes an approach for detecting pedestrians at night using a normal camera. In this approach, objects of interest are assumed to be brighter than the background, and objects are segmented using an adaptive thresholding method. The output of segmentation is the blocks representing areas that have a higher probability of containing useful information. However, in speed-sign detection we are also interested in the content of the speed sign—the circle and text—which is not preserved by this approach.

In [7], a color vision system is used, where input images are recorded in L^*a^*b format, which is meant to be robust to lighting conditions. This algorithm is based on a popular speed sign detection algorithm named radial symmetry [5]. Experiment results have shown that this algorithm is fast and accurate in various challenging daytime conditions. However, due to distortions at nighttime, the authors have reported unsatisfactory performance due to the failure of the L^*a^*b color filter scheme in low light conditions, similar to other color schemes, such as RGB, HSI (HSV).

In this chapter, we introduce a new approach based on the well-known difference of Gaussian (DoG) filter. Like DoG, the input image is filtered with Gaussian kernels. The difference between the first and the second filter level is computed. The output is then scaled and truncated appropriately instead of performing non-maxima suppress as in a standard DoG filter. The output from this computation is the image with silhouette-like edges at areas that have high variation in the high frequency component. The image can then be used for detection as a normal image or can be used to enhance the original image. Analysis of this approach is shown in Sect. 31.4. It is worth mentioning that this approach is not a night vision system that can operate without light. In fact, this approach works under artificial lightning sources such as street lights and vehicle head-lights.

31.3 Algorithm Descriptions

31.3.1 Gaussian Filter and Classical DoG

The Gaussian filter is a well-known filter that has been used for applications in many areas. In image processing, the Gaussian filter is used extensively for anti-aliasing. It is, in essence, a low pass filter in which the high frequency components of the image are removed. The cut-off frequency is determined by its standard deviation sigma σ. Eq. (31.1) describes the 2D Gaussian filter in a continuous domain [12]. The visualization of the filter is shown in Fig. 31.1, where x-y represents the 2D image.

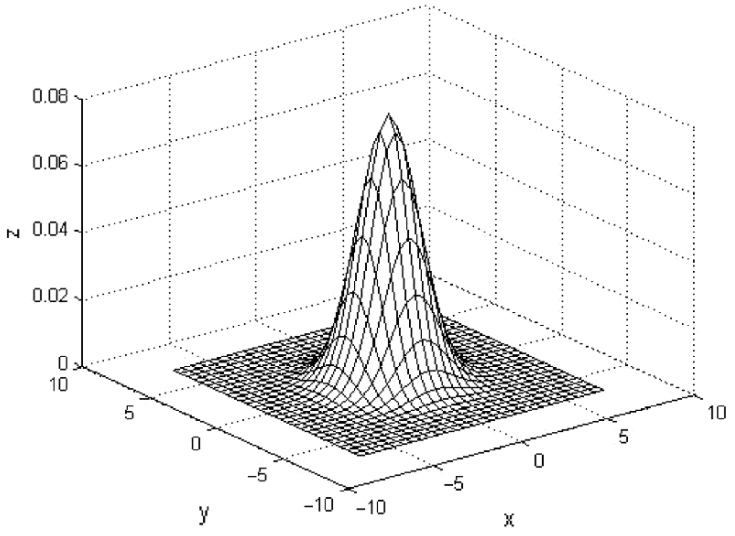


Fig. 31.1 Response of a Gaussian filter

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{31.1}$$

For a discrete signal like a digital image, we use the approximation of the Gaussian filter called a kernel of different size and standard deviation σ. Shown in Fig. 31.2 is a 5×5 Gaussian kernel approximation with $\sigma = 1$.

The DoG filter is calculated as the difference between the first and the second order of the convolved image with the Gaussian smoothing kernel. The original image is smoothed with the Gaussian kernel then smoothed again by the same kernel. The difference between the two smoothed images is the DoG filtered image K, shown in (31.2):

$$K = G_\sigma * (G_\sigma * I) - G_\sigma * I \tag{31.2}$$

where I is the original grayscale input image, and * indicates convolution operation. The characteristic graph of the classical DoG is shown in Fig. 31.3, with the current located at x = 0 and y = 0.

$$G_1 = \frac{1}{331}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 20 & 33 & 20 & 4 \\ 7 & 33 & 55 & 33 & 7 \\ 4 & 20 & 33 & 20 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Fig. 31.2 5×5, Gaussian approximation kernel, $\sigma = 1$

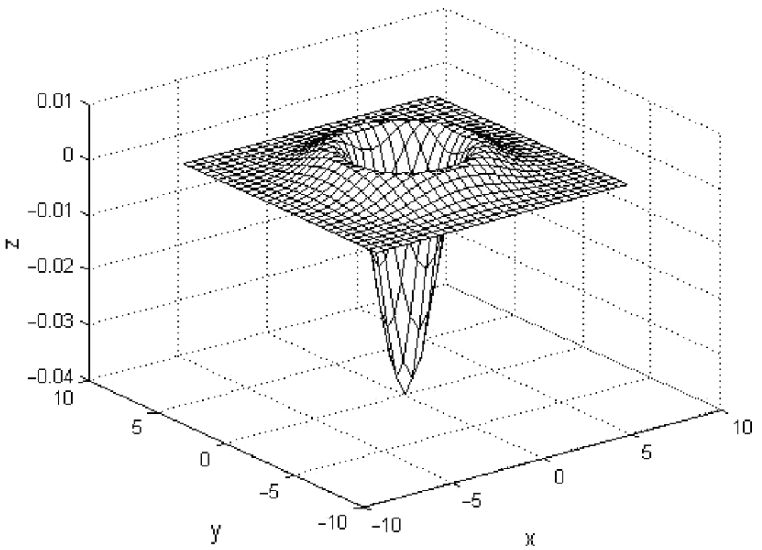


Fig. 31.3 Characteristic response of classical difference of Gaussian (DoG) filter

31.3.2 Adaptive Difference of Gaussian

In literature, DoG is used for edge detection. The input image is filtered with the Gaussian kernel twice, and the difference between them undergoes non-maxima suppress to detect the edge of objects. However, in our case the output of DoG filtering is kept without performing non-maxima suppress, and the DoG filtered image is scaled with a fix constant between 20 and 50. Any pixels exceeding 255 are set to 255. In our experiments 30 was chosen as the constant value. Any pixel with a negative value will be set to 0. The characteristic response of the adaptive filter is illustrated in Fig. 31.4. Note that there are some differences from the response of the classical DoG filter shown in Fig. 31.3, especially the inside region of the response. This characteristic response does not show the scaling factor or the relationship between response magnitude and frequency content within the local region, i.e., if the response is high the surrounding areas at a distance from the current pixel have high frequency content (edges). The immediate local region does not have any affect on the output of the pixel, which reduces the effect of motion blur on the speed sign.

It has been pointed out [12] that the DoG for edge detection is sensitive to noise, but for this application we are not truly performing edge detection on the input image. We are loosely estimating the position of areas with high pixel variation and representing them by thick trails (silhouette). The affect of random noise is also reduced by the use of a relatively large Gaussian kernel. The size of the Gaussian

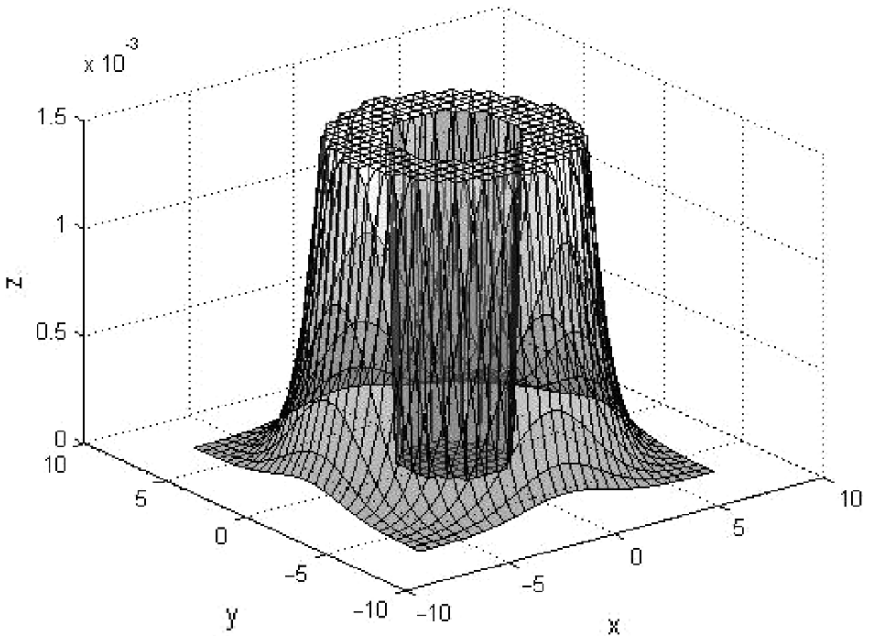


Fig. 31.4 Characteristic response of adaptive difference of Gaussian filter

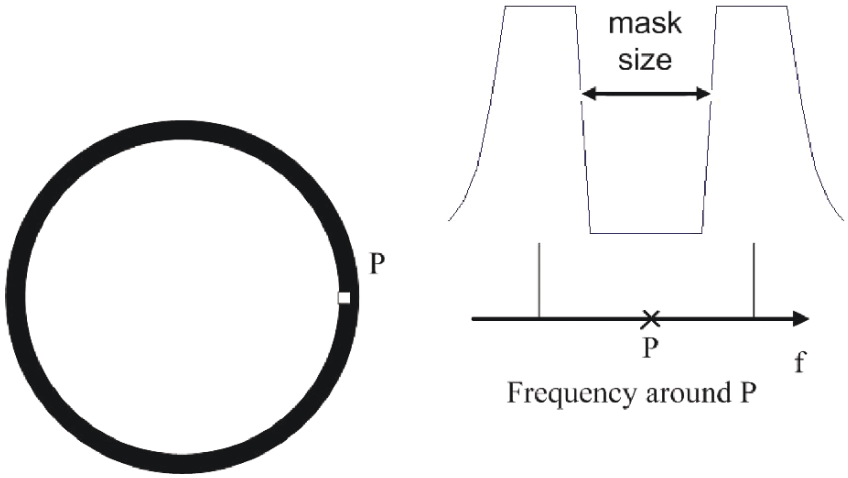


Fig. 31.5 Output of a point P on the circle being filtered by adaptive DoG

mask is chosen so that it is normally larger than the width of the circle's border. Typically the width of the circle's border does not exceed 10 pixels. In our experiments, we use a mask size of 7×7. The relationship between output of the adaptive DoG filter and the choice of Gaussian kernel size can be visualized in Fig. 31.5, where P is a point on the circle of the speed sign (or any alike structure). If the mask size is too large or too small, the high frequency of the regions surrounding P (to the left and the right of P) may not lie within the response region of the adaptive DoG filter (shown as a 2-D response).

The choice of standard deviation is also important. The larger the standard deviation is, the thicker the trail in the filtered output and the more distortion that is introduced to the object's structure. Typically, a σ value between 1 and 2 is chosen. We chose $\sigma = \sqrt{2}$ as used by Lowe [13].

31.4 Experiments and Results

31.4.1 Introduction of Radial Symmetry

This section briefly summarizes the main points of the radial symmetry algorithm; more details can be found in [5]. In this algorithm, grayscale images are used for detecting a speed sign by finding the circles within the image. Figure 31.6 shows the example images. The algorithm is targeted to find circle(s) in the image whose radii fall within a specified range, referred to as the searching range. At each radius within the searching range, an orientation projection image O_r and a magnitude projection image M_r are created. Every pixel in the input image votes for two potential circle

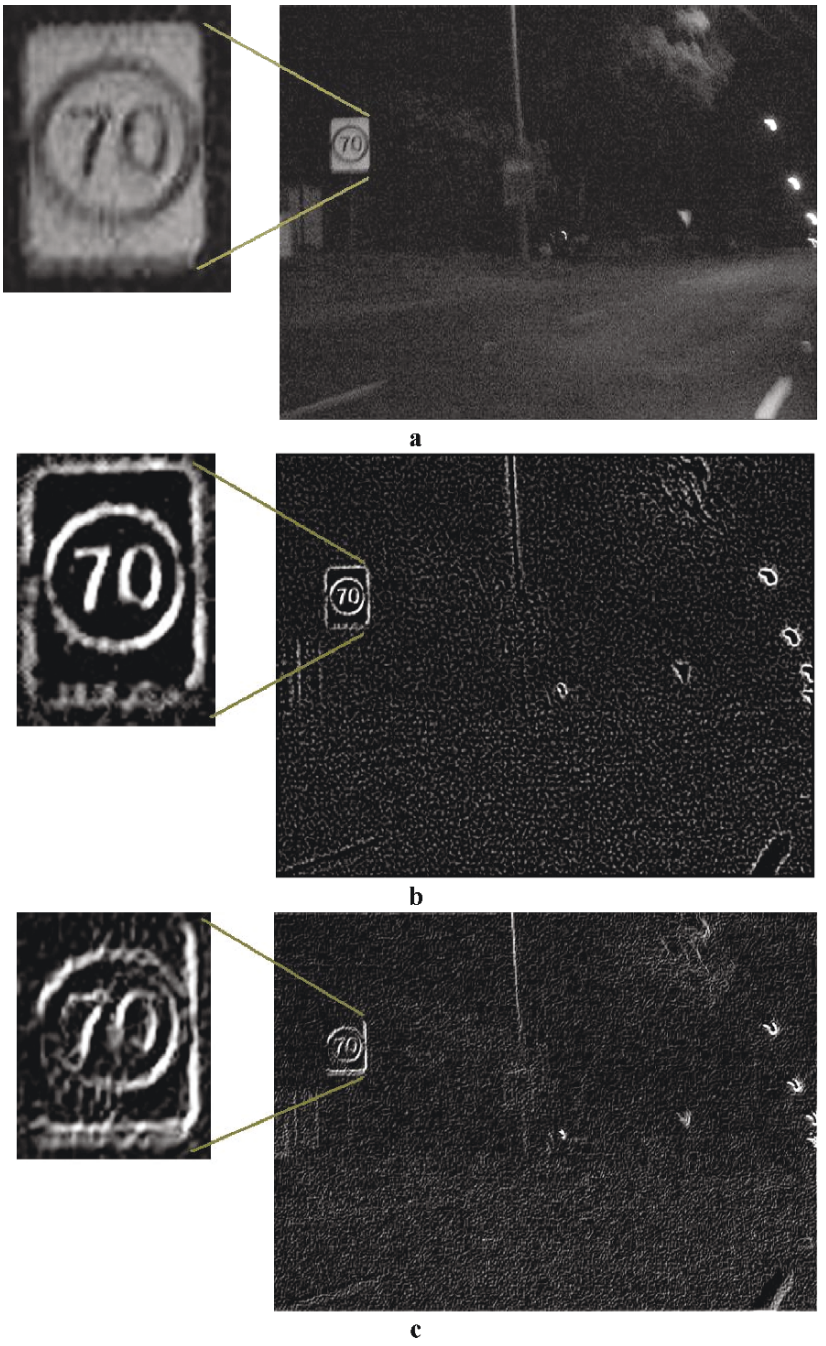


Fig. 31.6 Example images. **a** Recorded image; **b** Image processed with adaptive DoG, **c** Image processed with Sobel filtere (combined in both directions)

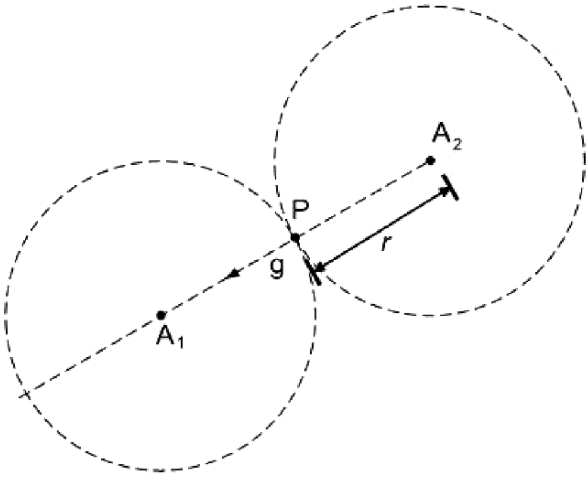


Fig. 31.7 Voting mechanism of radial symmetry algorithm

centers, referred to as affected pixels, at a distance r away from the current pixel's location, along the direction of its gradient. One of the two affected pixels is voted along the positive direction of the gradient; the other is voted along the negative direction. The pixel's gradient is calculated using Sobel edge filter masks in the horizontal and vertical directions. Positive affected and negative affected pixels are shown in Fig. 31.7, where P is the current pixel of the original image, and A1 and A2 are positive and negative affected pixels respectively, g is P's gradient vector, and r is the current radius under consideration. The two dashed circles are the two potential circles seen by the point P. If P is on the circle, then either A1 or A2 will be the true circle's center, and this center will be voted for by many other points on the circle. As a result, the true center will have the highest vote and will be selected.

The location of positive and negative affected pixels (in x,y coordinates) are determined as:

$$p_{\pm ve}(x) = x \pm round\left(\frac{g(x)}{||g||}r\right) \tag{31.3}$$

$$p_{\pm ve}(y) = y \pm round\left(\frac{g(y)}{||g||}r\right) \tag{31.4}$$

where $g(x)$ and $g(y)$ are the gradients of P in x and y direction respectively, and $\|g\|$ is magnitude of gradient.

The current values at the location of the positive affected pixel in the orientation image and in the magnitude image are increased by 1 and the magnitude of the gradient, respectively. In contrast, the current value at negative affected locations in the orientation and in the magnitude image is decreased by 1 and the gradient magnitude respectively. As shown by (31.5) and (31.6),

$$O_r(p_{\pm ve}) = O_r(p_{\pm ve}) \pm 1 \tag{31.5}$$

$$M_r(p_{\pm ve}) = M_r(p_{\pm ve}) \pm ||g|| \tag{31.6}$$

where $p_{\pm ve}$ is location of positive or negative affected pixels (in x,y coordinates).

After all pixels within the original image are processed, a symmetry image for the radius under consideration is created based on the processed orientation projection image and a magnitude projection image. After all of the radii within the searching range have been searched, a final symmetry image is created, and the best candidates are selected and verified.

31.4.2 Experiments

From practical data collections, a data pool is constructed from about 40 video sequences at a frame resolution of 640×480. Each video sequence contains between 130 and 1049 frames. These video sequences were recorded at night in different locations in Melbourne, including highway, city and residential areas. A test data set of about 1400 images was constructed by extracting representative video frames from those video sequences. Each video sequence contributes a different number of frames (images) to the test set depending on the content. In the test images, we consider a speed sign to be valid if it satisfies the following two conditions:

1. A human can comfortably tell what the content of the sign is without reasoning (the inferring process in the human brain)
2. Has object(s) enclosed in a circle and the circle's radius is between 12 and 40 pixels (inclusive)

Condition 1 is necessary because many frames in the video sequence exhibit an excessive level of motion blur that make them unreliable for any vision-based system. Other images that do not satisfy the two conditions above were treated as images without valid speed signs.

31.4.3 Results

In our experiments, a MATLAB model of the radial symmetry algorithm is used to compare its performance when using the DoG filtered images with its performance on unfiltered images. For the moment, we only consider the detection stage of the algorithm. Although classification of the speed sign text can reduce false detections, number classification is not considered at this point.

The PC used for testing was equipped with an AMD Athlon 64 X2 Dual Core 4800+, 2.41 GHz processor, 2 GB RAM, Windows XP SP2, and Matlab 7.0. A non-professional digital still camera (Nikon E8700) was used to record video sequences. Free software programs were used to extract frames from video sequences recorded.

Table 31.1 Test results

	Without DoG	With DoG
Number of Images Tested	1400	1400
Images with Valid Speed Sign	458	458
Detected Images	262	368
Detection Time (Seconds)	20.29	20.94

The overall test results are summarized in Table 31.1. We are comparing the detection performance of radial symmetry with and without DoG filter, so we consider the total number of images with speed sign(s) as a starting point. These images are selected from the total number of images (1400) by different local people. The criterion of selection is any image from which a person can correctly determine the content of the speed limit sign without reasoning (inferring) or guessing.

Table 31.1 has shown that without the DoG filter, the radial symmetry algorithm detects less than 60 percent of images containing speed sign(s). This figure is increased to 80.3 percent by applying the DoG filter. The difference between the two approaches is mostly due to the low contrast images recorded at night. A detected image is defined as when the speed sign(s) in the image is/are detected (highlighted) within 4 pixels from their true center.

The DoG filter cost about 650 ms of extra computation time on the PC. However, the computation requirement for filtering an image with adaptive DoG is insignificant compared to the voting process of the radial symmetry algorithm itself, which is run-time dependent. Additionally, due to the deterministic nature of filtering with DoG kernels and due to the separable property of DoG kernels, the DoG filtering process can be pipelined to increase speed.

The false detection rate of the radial symmetry algorithm is reduced if the adaptive DoG filtered image is used. This is because artificial light sources such as advertisement lights or vehicle headlights in the opposite direction can dominate the scene, which results in avery low contrast speed sign. Without using adaptive DoG, the algorithm may wrongly detect other bright objects as valid candidates. With adaptive DoG, speed signs and other objects within the scene are processed to the appropriate range of brightness and contrast, hence causing less confusion to the radial symmetry algorithm.

31.4.4 Initial Hardware Implementation of DoG

An initial implementation of the adaptive DoG filter has been developed using field programmable gate arrays (FPGA) technology. This technology has been chosen because of its parallel nature, i.e., the FPGA device can be configured to process multiple tasks in parallel, given that there are enough hardware resources available.

For this application, a model of the adaptive DoG filter has been developed to examine its suitability for implementation in hardware. The high-level block design

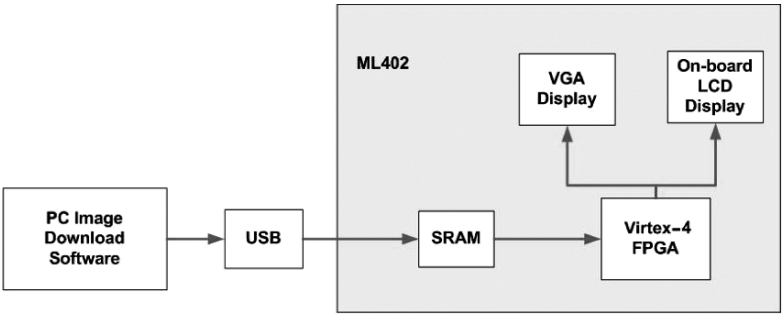


Fig. 31.8 High-level design of the initial FPGA implementation

of the model has been shown in Fig. 31.8. The development board chosen for this project is a Virtex-4 ML402. The grayscale image is read by a C++ program, after which it is sent to the SRAM of the development board. The image is then processed by the FPGA, and the resulting image is displayed to the video output (VGA) port available on the development board. The time taken to process is also displayed (in 1 ms resolution).

As a stand alone application, computing DoG takes about 16.9ms to complete on the development board with the default 100MHz system clock. However, in an integrated application, computing DoG can be pipelined with other processes, hence reduce the processing time. The time required can be greatly reduced by using block Ram (BRAM) on board the FPGA, as this can run at a much faster clock rate than the SRAM.

31.5 Conclusions and Future Work

In this chapter, we have presented a way of using the difference of Gaussian filter for speed sign detection during nighttime conditions. As opposed to normal use of DoG for edge detection method, we utilize the output of the DoG without performing non-maxima suppress. Instead, we perform scaling to amplify the high frequency component of the areas at a small distance away from the current pixel's location. The resultant image with edges highlighted as thick trails (rather than the normal edge detection output of 1 pixel line) is then processed with a radial symmetry algorithm. Experiments have been conducted with a large number of practical images and show that using a DoG filtered image increases by about 30 percent the detection capability of the radial symmetry algorithm.

Due to the nature of Matlab simulation, the whole system still requires a lot of computation time, but the majority of the computational effort is in the voting process of the radial symmetry algorithm, not the filtering. An initial hardware FPGA implementation of the adaptive DoG filter shows that the filtering process can be computed very quickly in hardware.

More investigation will be carried out to evaluate the performance of the adaptive DoG with other speed sign detection algorithms.

Acknowledgements The authors wish to acknowledge the support of the Co-operative Research Centre for Advanced Automotive Technology (AutoCRC) Australia, staff members at the Department of Electronics Engineering, Latrobe University, Australia; and Seeing Machines Ltd Pty, Australia; for the work described in this chapter.

References

1. Akatsuka H, and Imai S (1987) Road signposts recognition system. In: Proc. SAE Vehicle Highway Infrastructure: Safety Compatibility, pp 189–196
2. Bahlmann C, Zhu Y, Ramesh V, and Pellkofer MK (2005) A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In: Proc. 2005 IEEE Intelligent Vehicles Symposium, Las Vegas, USA
3. Shuang-dong Z, Yi Z, and Xiao-Feng L (2005) Detection for triangle traffic sign based on neural network. In: Proc. 2005 IEEE International Conference on Vehicular Electronics and Safety, Xi'an, China
4. Lafuente-Arroyo S, Gil-Jimenez P, Maldonado-Bascon R, Lopez-Ferreras F, and Maldonado-Bascon S (2005) Traffic sign shape classification evaluation I: SVM using distance to borders. In: Proc. 2005 IEEE Intelligent Vehicles Symposium, Las Vegas, USA
5. Barnes N and Alex Z (2004) Real-time radial symmetry for speed sign detection. In: Proc. 2004 IEEE Intelligent Vehicles Symposium, University of Parma, Parma, Italy
6. Barnes N (2005) Improved signal to noise ratio and computational speed for gradient-based detection algorithms. In: Proc. 2005 International Conference on Robotics and Automation, Barcelona, Spain
7. Siogkas GK and Dermatas ES (2006) Detection, tracking and classification of road signs in adverse conditions. In: Proc. 2006 IEEE Mediterranean Electrotechnical Conference, Spain
8. Soetedjo A and Yamada K (2005) Fast and robust traffic sign detection. In: Proc. 2005 IEEE International Conference Systems, Man and Cybernetics, Hawaii, USA
9. Lafuente-Arroyo S, Gil-Jimenez P, Maldonado-Bascon R, Lopez-Ferreras F, and Maldonado-Bascon S (2005) Traffic sign shape classification evaluation II: FFT applied to the signature of blobs. In: Proc. 2005 IEEE Intelligent Vehicles Symposium
10. Bertozzi M, Broggi A, Fascioli A, Graf T, Meinecke MM (2004) Pedestrian detection for driver assistance using multi-resolution infrared vision. IEEE Transactions on Vehicular Technology 53(6):1666–1678
11. Bin Z, Qiming T and Yupin L (2005) An improved pedestrian detection approach for cluttered background in nighttime. In: Proc. 2005 IEEE International Conference on Vehicular Electronics and Safety
12. Forsyth DA and Ponce J (2003), Computer Vision: A Modern Approach. Prentice Hall, Upper Saddle River, NJ, pp 169–175
13. Lowe D (2003) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60: 91–110

Chapter 32
Binarization of Engineering Drawings Using Adaptive Thresholding Techniques

Shih-Chang Liang and Wen-Jan Chen

32.1 Introduction

Image binarization is an important issue in image segmentation. In recent years, numerous studies have been focused on this area. Navon et al. [5] described the segmentation methods including histogram-based methods, boundary-based methods, region-based methods, hybrid-based methods, and graph-based technique. When the threshold method is applied to image binarization, either a global threshold or local threshold method will be used. The former method uses histogram analysis to search for a single threshold in an image and differentiate objects and background according to a comparison of all the pixels within the threshold. The latter method first segments an image into individual blocks and calculates the threshold of each block or even each pixel.

Fig. 32.1 shows three common engineering drawings including **a** Blueprint, **b** Hand drawing and **c** Printed document. Engineering drawings have four major characteristics:

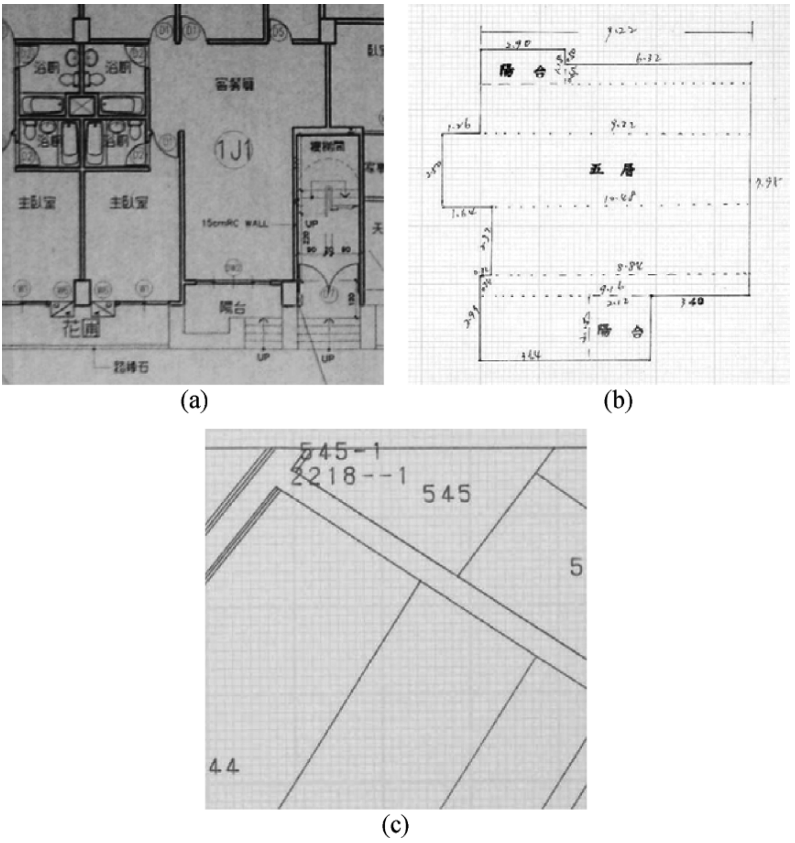
1. The content is mainly composed of straight lines and some text and arcs.
2. The variation in the length and width of lines is large, and no rule can apply.
3. Gray-scale values that compose straight lines appear erratically.
4. There is more background noise, and the intensity is not smooth.

Both global and local threshold methods have a certain weakness. While the global threshold method ignores regional information, the local threshold method will ignore global information. The accuracy of binarization will be affected.

O'Gorman [6] measured the intensity of vertical and horizontal gray-scale values in a histogram and derived regional information such as degree of flatness by calculating variances with a sliding profile. At last, the number of global threshold

Shih-Chang Liang and Wen-Jan Chen
Department of Computer Science and Information Engineering, DaYeh University, Taiwan, R.O.C.

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*



(a) (b) (c)

Fig. 32.1 Engineering drawing images: **a)** blueprint **b)** hand drawing, and **c)** printed document

values will be determined. This method improves the global threshold method with regional connectivity information and is an integrated application of threshold methods.

Sauvola [7] classified all the threshold methods into multi-threshold, single threshold, local threshold, and global threshold. The first letters of the 4 methods compose the MSLG method. In practical application with different image sources, M, S, L, and G can be integrated into ML, SL, MG, and SG hybrid methods for image binarization.

The traditional local threshold method usually segments an image into multiple small ones and processes them one by one. In 2000, Yang et al. [9] and Zhao et al. [10] proposed to adaptively calculate a threshold value for each pixel. The adaptive method refers to the intensity and position of adjacent gray-scale values and compares it with a certain range based on the pixel. As Yang et al.'s method mainly deals with printed texts and Zhao et al's method focuses more on blueprint images, the main difference between the two methods lies in the different shapes adopted in determining adjacent pixels.

When the adaptive local threshold method is applied to binarization of engineering drawings, lines are the objects to be processed. The most difficult task is to avoid deletion of line terminal points and intersections, prevent line breaks, and remove isolated pixels as noise. The processing of adjacency is similar to a peeling approach, commonly called thinning or skeletonization. The various line styles proposed by Deng et al. [1] can be further applied to adaptive calculation of pixel adjacency styles.

It can be discovered from Fig. 32.1 that on different engineering drawings the distribution of gray-scale values may vary tremendously. Despite the application of adaptive local calculation, it is still hard to correctly differentiate objects and background. In recent years, some studies have been dedicated to address different image sources and propose new solutions for threshold methods. For instance, images can be processed with a low-pass filter before binarization. The edge regions with blurry sharp variation are used to eliminate most of the noise. A Wiener filter is commonly used in this pre-processing stage. After binarization, post-processing will be conducted to remove isolated pixels, amend broken lines, and fill in spaces. Lee et al. [4] adopted automatic analysis of histograms for engineering drawings. According to experiential figures, images are automatically classified to derive knowledge parameters for different drawing areas as important information for calculating adaptive threshold values.

This chapter integrates a hybrid SLG method. A global threshold value is first applied to filter background pixels, and the adaptive local threshold method is adopted to add multiple line styles and detect characteristics of thin lines. Dynamic multilevel calculation is used to reinforce the rule of calculating the threshold value of each pixel. Gray-scale engineering drawings can be binarized effectively and accurately. The proposed adaptive threshold-based image binarization technique has been empirically proven effective in gaining a high quality result from different kinds of gray-scale drawings, especially the ones with thin lines where information can be easily lost. In Sect. 32.2, an overview of related studies on adaptive threshold-based image binarization will be provided. Section 32.3 details our method and procedure, and experimental results and conclusions are proposed in Sects. 32.4 and 32.5, respectively.

32.2 Related Work

32.2.1 Adaptive Logical Level Thresholding Technique

Yang and Yan's [9] method, which binarizes images with an adaptive threshold, uses histogram analysis on local regions and automatically calculates the local regions around the center of the processed pixel, as shown in Fig. 32.2. The threshold value of the local region will be adaptively derived.

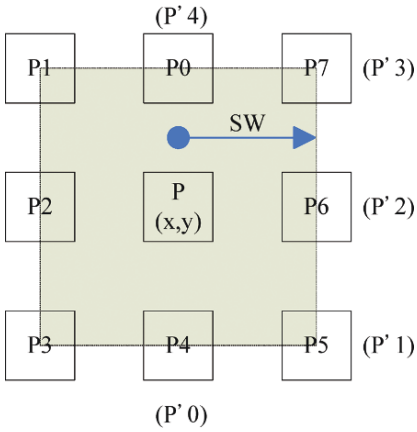


Fig. 32.2 The neighboring window of processing point

In image binarization, all the pixels are scanned one after another, and the binarization result is determined by the following logic equation:

$$b\{x,y\} = \begin{cases} 1 & \text{if } \vee_{i=0}^{3} [L(P_i) \wedge L(P_i) \wedge L(P_{i+1}) \wedge L(P_{i+1})] \text{ is true.} \\ 0 & otherwise. \end{cases}$$

$$P_i' = P_{(i+4) \bmod 8}, \text{ for } i = 0, \ldots, 7,$$

$$L(P) = ave(P) - g(x,y) > T$$

$\wedge$ is AND operate, $\vee$ is OR operate

$$\text{ave}(P_i) = \frac{1}{(2sw+1)^2} \times \sum_{-sw \le m \le sw} \sum_{-sw \le n \le sw} f(x_i + m, y_i + n),$$

The adaptive logic threshold technique is able to automatically calculate local regions and thresholds and reinforce the original logic threshold method. For text-based printed document images, especially when the contrast is low and noise is high, a better binarization result can be obtained. This technique can also be applied to process documents with interwoven text and lines but not portrait or landscape photos.

32.2.2 C Shape Method

Zhao et al.'s [10] method discovered the characteristic that blueprint images are mainly composed of straight lines and arcs. Two conclusions were thus proposed: Objects are relatively darker than adjacent pixels, and objects in small regions are in a shape close to "C." Under this assumption, isolated pixels have a lower gray-scale value, but they cannot create a C shape with adjacent pixels, so they should not be selected as objects.

Take Fig. 32.2 as an example. The stereotypical C shape is created by P0-P1-P2 or P2-P3-P4 or P4-P5-P6 or P6-P7-P0. When any of the three adjacent windows with a lower gray-scale value surrounds P(x, y), the processed pixel P(x, y) is set as an object; otherwise, it is set as background. When the three pixels with a lower gray-scale value constitute a straight line, it will be an exception of the C shape, and the processed pixel P(x,y) is set as an object.

Multiple SWs are adopted for the C shape. In other words, a large SW is chosen first to calculate the processed pixels. If a pixel is binarized as an object pixel, the next pixel will be processed; otherwise, a smaller SW will be adopted for recalculation. The average gray-scale values of pixels around the processed pixel and the average gray-scale values of eight neighboring areas will be taken as the basis of analysis. The distance from the processed pixel and the center of eight neighboring areas is SW. The algorithm is detailed as follows:

1. The processed pixel is taken as the center to analyze the histogram. 2 square sizes W1 and W2 are selected $(\mathrm{W1} < \mathrm{W2})$ and $\mathrm{SW} = \mathrm{W2}$.
2. The square size is $(2\mathrm{SW}+1) \times (2\mathrm{SW}+1)$. On the processed pixel's C shape, ave(Pi), ave$(\mathrm{Pi}+1)$, ave$(\mathrm{Pi}+2)$, are average gray-scale values of any 3 consecutive squares used to compare with borders of lines or arcs.

$$t_i = \begin{cases} 1 & \text{if } (L(P_i) \wedge L(P_{i+1}) \wedge L(P_{i+2})) \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

$$(i = 0, \ldots, 7; \text{ and } \mathrm{ave}(P_8) = \mathrm{ave}(P_0), \mathrm{ave}(P_9) = \mathrm{ave}(P_1))$$

then

$$b(x,y) = \begin{cases} 1 & \text{if } \bigvee\limits_{i=0,\ldots,7} t_i \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

$$L(P_i) = \mathrm{ave}(P_i) - f(x,y) > T.$$

3. If b(x, y) = 1, continue to process the next pixel; otherwise, set SW = W1 and repeat Step 2.

If the threshold value is calculated by the minimum gray-scale value and the average value of local regions, when the image has a high brightness background, object brightness will be higher too. The minimum gray-scale value and the average value of local regions will increase and result in an excessive threshold value and incorrect binarization. On the contrary, for images of a lower brightness, the threshold value may be too small. Thus, a parameter α can be added to rectify the threshold value T.

$$\alpha = \begin{cases} 1 & \text{if } a < 90. \\ 0.33 & \text{if } 90 \le a \le 170. \\ 0.1 & \text{if } a > 170, \end{cases}$$

a is the average gray-scale value of the square around the processed pixel.

$T \rightarrow T + \Delta T$, and ΔT is within a smaller range of integers (such as $[-20, 20]$). As $0 < \alpha \leqq 1$, and in most cases, $\alpha = 0.33$, the parameter α can be used to adjust the threshold value of the range $[-6, 6]$.

32.3 Proposed Method

It can be discovered from Fig. 32.1 that different types of engineering drawings have significant differences in image quality, brightness, and contrast. And Zhao et al.'s [10] method is applied to these line and text-based gray-scale images. The adjacent regions are used to calculate the dynamic threshold for local regions. For printed texts and thick lines, a proper square size (such as $SW = 3$) and the threshold adjustment range (such as $20 \sim 120$) can be set to acquire a good binarization result. But in the case of a thin line as wide as a pixel, errors may occur.

On the left image of Fig. 32.3, the central processed pixel (R2, C2) has a gray-scale value of 200 and should be judged as an object (black). As the 8 neighboring points do not constitute the shape defined by method [9] and [10], it can be mistakenly classified as background (white). On the right image of Fig. 32.3, the central processed pixel (R2, C2) has a gray-scale value of 202 and should be judged as an object (black). However, method [9] and [10] will also mistake it as background (white). This error is simply the misclassification of a pixel but will lead to broken lines and the damage of line intersections. The follow-up operation of thinning and feature recognition will be difficult.

32.3.1 Recognition of Line Patterns

To solve misrecognition of thin lines and line intersections with the method [10], the property of line adjacency should be used to correctly judge whether the processed pixel should be an object or background.

In the first-level calculation, the large C shape square $(SW = 3)$ derived with the method [10] is first used to calculate local dynamic threshold values. If the processed

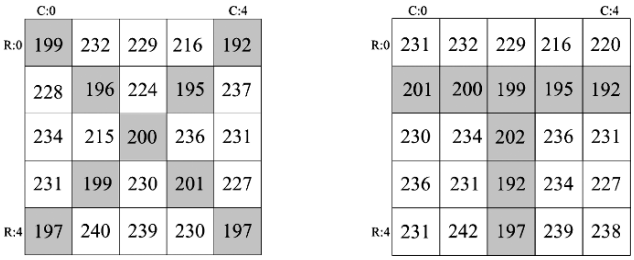


Fig. 32.3 Intersection line with one pixel

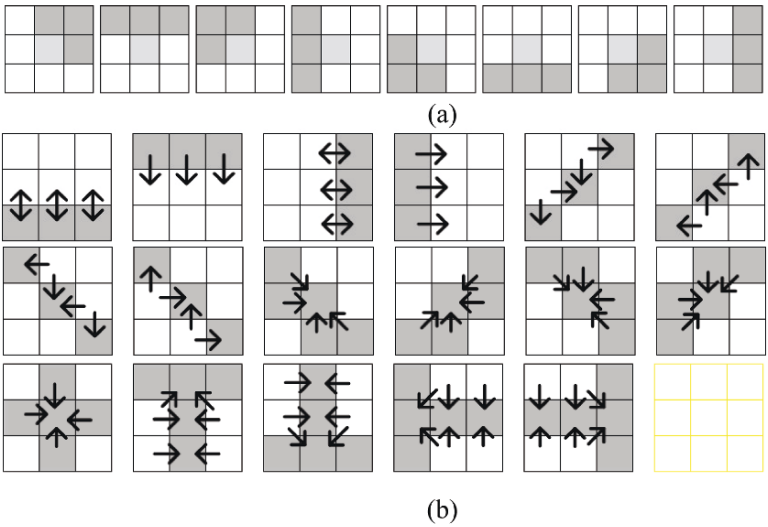


Fig. 32.4 3×3 templates for line recognize **a** "C" shape of Zhao et al. [10]. **b** Proposed line patterns

pixel is an object, process the next pixel; otherwise, enter the second-level calculation with a small-size square $(\mathrm{SW} = 1)$. In the second-level calculation, use the method [10]'s C shape operation as shown in Fig. 32.4a and expand the judgment of C-shape flexible model. If the processed pixel is still not an object, continue to use the 17 straight line styles shown in Fig. 32.4b to judge the pixel adjacency and determine the threshold value for binarization.

On Fig. 32.4, a total of 25 3×3 straight line styles are used for comparison, so thin lines and intersection points can be accurately processed. The arrows on Fig. 32.4b indicate the direction for calculating 2 adjacent pixels, which can be used to compare the edges of straight lines and arcs.

Through the above-mentioned comparison of straight line patterns, on the left image of Fig. 32.3, the central processed pixel (gray-scale value 200) can be correctly recognized with one of the four adjacency styles (The 5th and 6th on the 1st row and 1st and 2nd on the 2nd row, shown in Fig. 32.4b) as an object (the binarization result = 1). On the right image of Fig. 32.3, the processed pixel (gray-scale value 202) can be correctly recognized with T style (the 2nd style on the 3rd row, as shown on Fig. 32.4(b)) as an object (the binarization result = 1).

32.3.2 Calculation of Threshold Values

The method [10] calculates threshold values based on the average of all the gray-scale values in the square around the processed pixel and determines the coefficient

α by the average value. In addition to textures, the background of engineering drawings is usually white. Thus, the average is larger than 170, and the α value will not constantly be 0.33 as proposed by method [10], but almost 0.1. This requires an increase of the adjustment range T to 20–120, so as to correctly differentiate objects and background. Thus, according to image brightness and contrast, α should be properly adjusted.

The proposed method is to automatically analyze the min, max, average, and variance of gray-scale values in the beginning. Further, based upon the knowledge-base established by the above parameters, engineering drawings will be classified. The global threshold, dynamic local threshold α, and threshold adjustment range T of the classification can be acquired.

The application of global threshold T1 is a method to enhance the overall processing efficiency. Through the analysis of various engineering drawings, it can be discovered that the proportions of object pixels and background pixels are in great contrast. In other words, the gray-scale value of object pixels is concentrated on the end of lower gray-scale values. According to the classification of images, a global threshold value T1 can be configured (ranging between average gray-scale value and max gray-scale value). If a gray-scale value is larger than that of the T1 pixel, it can be directly judged as background; if it is smaller than or equal to T1, local threshold can be further used to process it. This can avoid the comparison of C-shape adjacency and straight lines styles for all the pixels and significantly improve operation efficiency.

32.3.3 Algorithm Procedure

The proposed algorithm is detailed as follows (with the flowchart of the proposed method shown in Fig. 32.5).

1. Calculate the min, max, average, and variance of the entire gray-scale image. The derived result and established rules can be used to classify gray-scale images to obtain global threshold T1, dynamic local threshold $\alpha, \Delta T$, and square size SW.
2. Scan the entire image and binarize each pixel: object $= 1$, and background $= 0$.
   a. If the gray-scale value of the processed pixel f(x, y) is larger than global threshold T1, after binarization, the processed pixel $b(x, y) = 0$.
   b. Else, the processed pixel is taken as the center, and the square size is $(2SW + 1) \times (2SW + 1)$. Calculate the average gray-scale value of pixels within the square (ave) and dynamic local threshold $T = \alpha \times \Delta T$. ΔT is the range of threshold adjustment [20–120].
   c. Extract the gray-scale values of 8 neighboring pixels. If any 3 serial pixels of the eight neighboring pixels have a gray-scale value $(Pi - f(x)) > T$, the processed pixel is binarized as $b(x, y) = 1$.
   d. Centered on the processed pixel, extract 3×3 sub images and compare them with the 225 straight line styles one by one. If the marked pixel's gray-scale

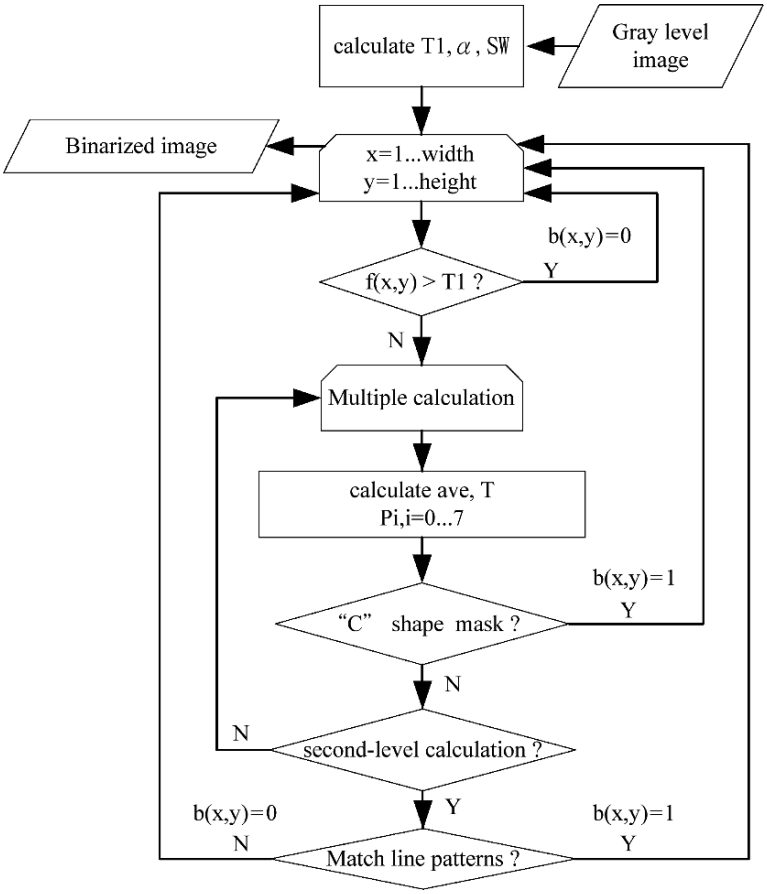


Fig. 32.5 The flowchart of the proposed method

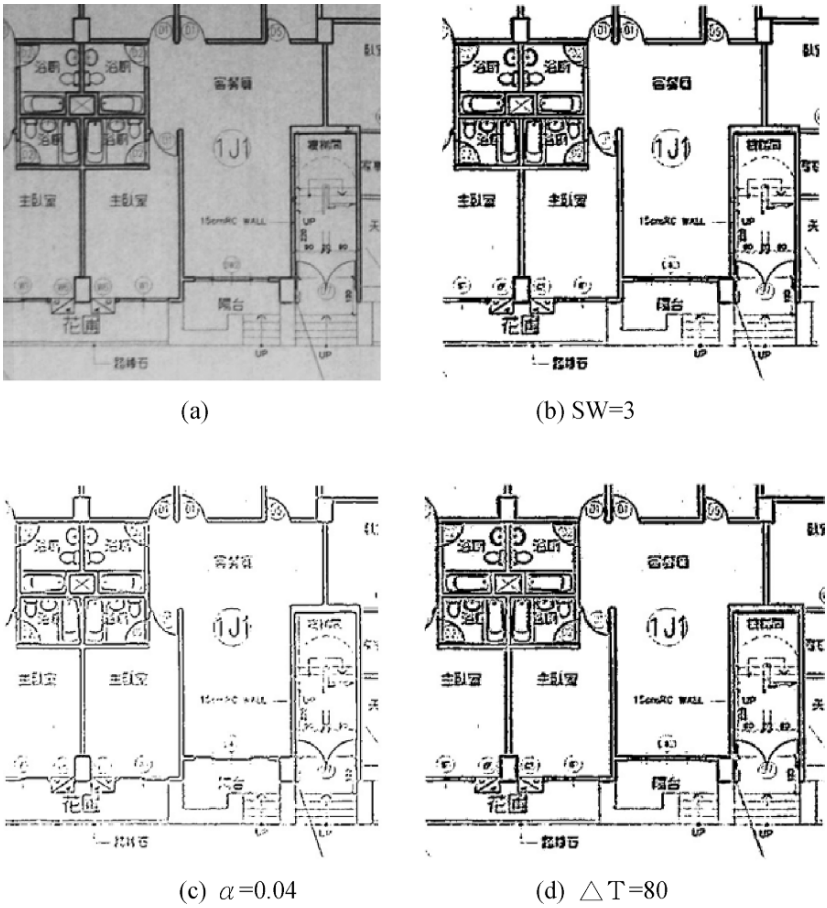
difference of any 3×3 straight line $>$ T, the processed pixel is binarized as $b(x, y) = 1$.

e. Else, the processed pixel is binarized as $b(x, y) = 0$.

3. If all the pixels are not processed, repeat Step 2; Else, output the binarized image and the task is completed.

32.4 Experiments

The simulation results of Yang [9], Zhao [10] and the proposed method show in Figs. 32.6 through 32.8 for processing the various kinds of engineering drawings.



(a) (b) SW=3

(c) α=0.04 (d) $\triangle$T=80

Fig. 32.6 Blueprints: **a)** original image, **b)** proposed method, **c)** Yang and Yan [9], and **d)** Zhao et al. [10]

The proposed method mainly improved the method [10] mechanism for processing thin lines and line intersections. To verify the ability of processing thin lines, first print the left corner of Fig. 32.8a where 3 parallel thin lines are perpendicular to other thin lines and capture a 15×19 sub image. The gray-scale value ranges from 168–251, and the average is 221. The variance is 20.3.

For further elaboration of the differences among these methods, a list of binaries derived from the original 15×19 gray-scale image on Fig.32.9a and the proposed method is presented on Fig. 32.10a and 10b, respectively. It can be observed from Fig. 32.7 and Fig. 32.8 that when a gray-scale image has a low quality, the proposed method can correctly differentiate all the thin lines and pixels of line intersections.

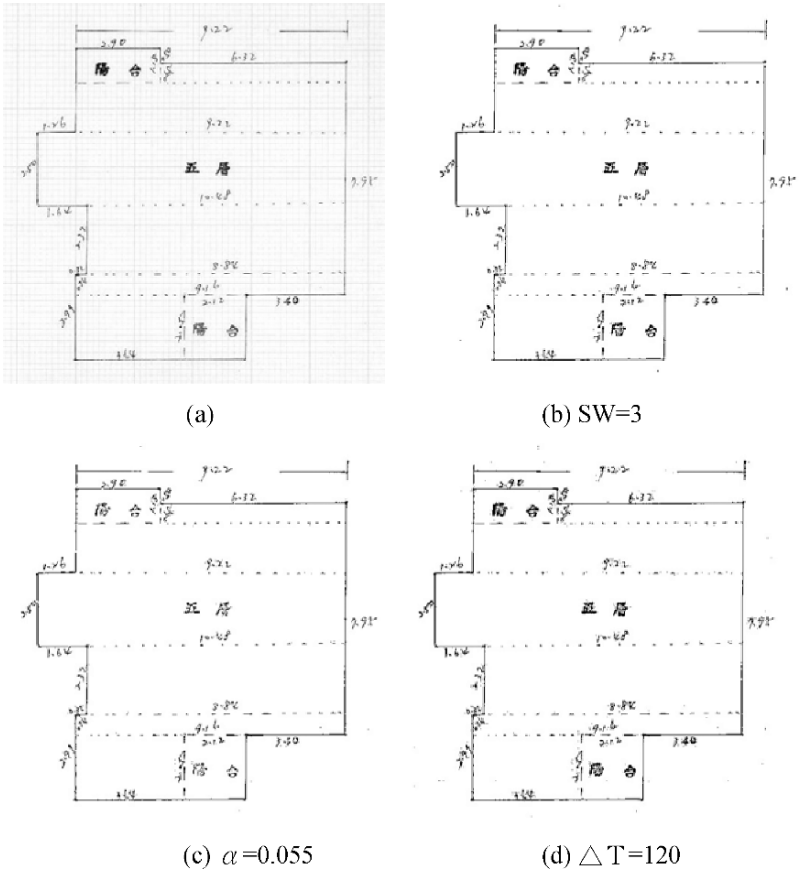


Fig. 32.7 Hand drawing: **a)** original image **b)** proposed method **c)** Yang and Yan [9], and **d)** Zhao et al. [10]

32.5 Conclusions

The proposed method not only improves the adaptive local threshold method [10] but also reinforces the ability of identifying thin lines and pixels of intersections. By analyzing histograms to recognize different engineering drawings, the proposed method can automatically apply the important calculation parameters, such as global threshold and local threshold values. The result of binarized histograms can be automatically acquired.

Through the verification with an exclusively developed application, the proposed method can still reasonably isolate thin lines and intersections when image quality is low. Broken lines and spaces can be effectively reduced, and the binarization result is more accurate than that derived by other threshold methods.

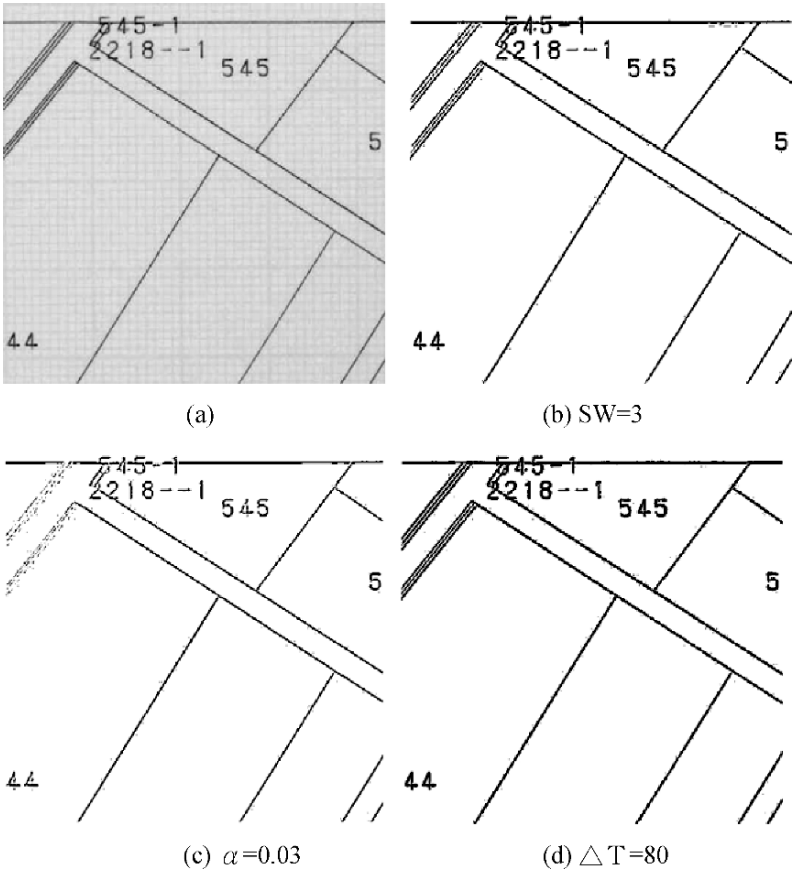


Fig. 32.8 Printed document image: **a)** original image **b)** proposed method **c)** Yang and Yan [9], and **d)** Zhao et al. [10]

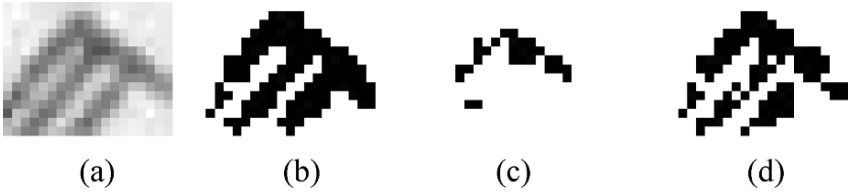
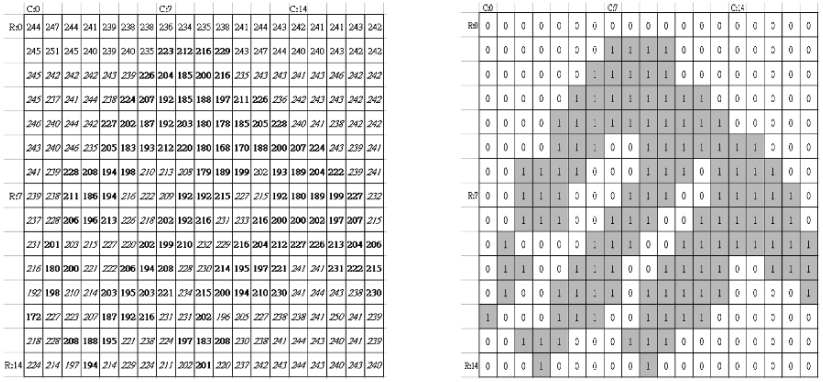


Fig. 32.9 Printed document image (15×19): **a)** original image, **b)** proposed method, **c)** Yang and Yan [9], and **d)** Zhao et al. [10]



(a) (b)

Fig. 32.10 Printed document image (1519): **a)** gray-level pixels, and **b)** binaried pixels (processed)

References

1. Deng W, Iyengar SS, Brener NE (2000) A fast parallel thinning algorithm for the binary image skeletonization. The International Journal of High Performance Computing Applications 14:65–81
2. Gatos B, Pratikakis I, Perantonis SJ (2004) An adaptive binarization technique for low quality historical documents. Marinai S, Dengel A (eds.) D AS, LNCS 3163, 2004, pp. 102–113
3. Huang Q, Gao W, Cai W (2005) Thresholding technique with adaptive window selection for uneven lighting image. Pattern Recognition Letters 26:801–808
4. Lee K-H, Cho S-B, Choy Y-C (2000) Automated vectorization of cartographic maps by a knowledge-based system. Engineering Applications of Artificial Intelligence 13:165–178
5. Navon E, Miller O, Averbuch A (2005) Color image segmentation based on adaptive local thresholds. Image and Vision Computing 23:69–85
6. O'Gorman L (1994) Binarization and multithresholding of document images using connectivity. CVGIP: Graphical Models and Image Processing 56:494–506
7. Sauvola., Pietikainen M (2000) Adaptive document image binarization. Pattern Recognition 33:225–236
8. Smith EHB (2002) Uniqueness of bilevel image degradations. SPIE Document Recognition and Retrieval VIII, San Jose, CA, pp. 20–25
9. Yang Y, Yan H (2000) An adaptive logical method for binarization of degraded document images. Pattern Recognition 33:787–807
10. Zhao M, Yang Y, and Yan H (2000) An adaptive thresholding method for binarization of blueprint images. Pattern Recognition Letters.21:927–943

Chapter 33
A Robust Watermarking Scheme for Digital Images Using Self Reference

Hsien-Wen Tseng and Chi-Pin Hsieh

33.1 Introduction

In recent years, transmission of digital multimedia via the Internet has become much easier and faster. The wide use of digital media has also led to an increase in digital piracy. In order to solve this problem, digital watermarking schemes that declare the ownership of digital media have received considerable attention. Digital watermarking is a process of permanently embedding a signature or a copyright message, such as a logo, or series number into digital media. Subsequently, the watermark can be extracted or detected from the watermarked media for copyright ownership identification.

There are a number of essential properties that a watermarking scheme must satisfy. The importance of each property depends on the requirement of the specific application. However, a watermarking technique should at least require the following properties.

1. *Imperceptibility*: After embedding watermark into cover media, it should not be noticeable to the viewer. The difference between original and marked medium is imperceptible to human eyes. The quality of the marked medium should not be degraded too much, and the attackers will not doubt of the marked medium. Moreover, for high robustness, it is desirable that attack tolerance be as high as possible. Thus, the tradeoff between imperceptibility and robustness is always considered when designing a watermarking scheme.
2. *Robustness*: The processing of digital images involves many manipulations and attacks, such as blurring, sharpening, compression, noising, cropping and so on. The embedded watermark must be strong enough to resist image processing

Hsien-Wen Tseng and Chi-Pin Hsieh
Department of Information Management Chaoyang University of Technology No. 168, Jifong E. Rd., Wufong Township, Taichxung County, Taiwan 41349, R.O.C
e-mail: {hwtseng,s9414624}@cyut.edu.tw

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

and attacks, otherwise it cannot protect the copyright to declare ownership. According to the degree of robustness, watermarks can be classified into three categories: robust, semi-fragile, and fragile. Unlike robust schemes, semi-fragile and fragile watermarks are designed for the purposes of authentication and content integrity verification.

3. *Security*: According to Kerckoff's principle, the security of a watermarking system should not depend upon the assumption that attacker does not know the watermarking algorithm. A cryptosystem achieves security by depending only on a secret key. Only authorized users who own the secret key can gain access to the watermark data.
4. *Unambiguity*: Retrieval of the watermark must precisely identify the owner of an image without ambiguity because an ambiguous result cannot provide accuracy of owner identification.

There have been numerous watermarking algorithms proposed in the last decade, which can be broadly classified into spatial domain and frequency domain. Some spatial-domain-based watermarking schemes [2, 11, 14, 17] mostly alter specific pixels in the cover image directly. However, spatial domain watermark methods achieve robustness by embedding a high intensity digital watermark, so it is difficult to achieve imperceptibility.

In order to achieve robustness, imperceptibility, and security, the frequency domain watermarking techniques often modify the coefficients to embed watermarks in transform domains such as discrete wavelet transform (DWT), discrete cosine transform (DCT), and discrete fourier transform (DFT). Cox et al. [6] embedded the watermark sequence into 1000 highest magnitude AC coefficients in the DCT transform domain. Since the watermark is embedded into the most significant regions of the cover image, it is strong enough to resist against common image processing and attacks. Hsu et al. [8] permutated a visually recognizable binary pattern and then shuffled it into every image block according to the variances of the image block. The middle-frequency values of the image are selectively modified to embed the permutated pattern. Joo et al. [10] obtained a reference image by using a low-pass filter. The watermark is embedded by repeatedly modifying the coefficients in the wavelet DC sub-band according to its reference image. However, these methods are very time-consuming. Chen et al. [4] presented a watermark scheme by integrating wavelet transform, digital signature, and digital time-stamp. Experimental results demonstrate that the watermark system is robust against common image processing and simple geometric distortions. A watermarking scheme based on subsampling is firstly proposed by Chu [5]. One watermark is inserted into one pair of values from two different subimages situated in the same DCT domain location. Hsieh et al. [7] embedded a visually recognizable pattern by modifying the qualified significant wavelet tree (QSWT). Temi et al. [16] also used QSWT to select locations in the embedding and extraction process. Watermark data is embedded by modifying the root of QSWT. Ni et al. [13] used fractal dimension to determine if the block of host image contains edges and textures. The watermark is embedded into the DCT middle-frequency coefficients of each feature block that contains edges and textures.

Nafornita et al. [12] computed each different threshold for each scale and subband based on the statistical analysis of the wavelet coefficients. If these coefficients are higher than a threshold, they are selected to embed the watermark. Yuan et al. [19] proposed a watermark system that embeds two watermarks into the most and the less significant subbands separately to resist all kinds of attacks.

In order to effectively modify the cover image without degrading subjective image quality seriously, human visual system (HVS) has been widely used to improve watermark invisibility and robustness. Barni et al. [1] modified the three highest frequency components in the DWT domain according to a weighing function considering the local sensitivity of the image to noise. Experimental results show that this scheme is resistant against JPEG and cropping but the attacker can remove the watermark by discarding these subbands. Xie et al. [18] overcame the major weakness by embedding the watermark into the most-resilient coefficients of all the detail subbands. The most-resilient coefficients are determined by comparing the JPEG compressed image with the host image. Reddy et al. [15] also used Barni et al.'s algorithm to calculate the weight factors.

Traditional watermark techniques mostly modify a selected portion of the cover image to embed the watermark. The distortion of the marked image is often small and imperceptible to human eyes, but it is not admissible in some sensitive applications, especially in military and medical image. Several vector quantization based watermark schemes without modifying the original image have been proposed recently [3, 9]. Unlike traditional watermark techniques, these kinds of watermark schemes do not modify a specific portion of the cover work at all. It is called zero-watermark.

In this chapter, a zero-watermark scheme that does not alter the original image is proposed. The reference image is used to produce the polarity table. Watermark data is embedded based on the polarity table. Experimental results demonstrate that the proposed method is robust against image processing or several attacks.

The rest of the chapter is organized as follows: In Sect. 33.2, a robust watermarking algorithm using a self-reference image is reviewed. The proposed embedding and extraction algorithms are described in Sect. 33.3. Experimental results and comparisons are presented in Sect. 33.4. Finally, in the last section, some conclusions are provided.

33.2 Related Work

Liu et al. [11] proposed a robust watermarking scheme using a self-reference image in 2006. The original image X is transformed into DWT domain by a one-level wavelet transform, and three subbands (LH_1, HL_1, and HH_1) are set to zero. After performing inverse wavelet transform, its reference image X' is obtained. The flowchart of reference image generation is shown in Fig. 33.1. The difference between the original image X and its reference image X' is computed by using Eq. (33.1),

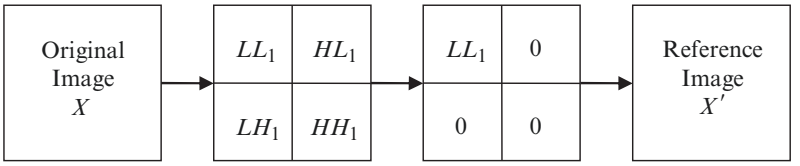


Fig. 33.1 Flowchart of reference image generation

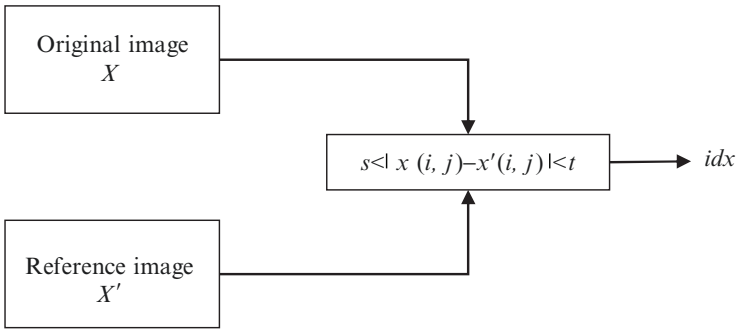


Fig. 33.2 Location information for watermark embedding

and then the embedding location $idx(i,j)$ in the watermark embedding process is obtained as shown in Fig. 33.2.

$$s < |x(i,j) - x'(i,j)| < t, \qquad s,t \in Z^{+}, \tag{33.1}$$

where x and x' denote pixel values of original image and its reference image, respectively.

A binary watermark sequence W of size n is generated by using a random number generator with specific seed. It is defined as Eq. (33.2). Finally, the watermark data is embedded into specific locations, which are randomly selected from the location information $idx(i,j)$.

$$W = \{w(k) | 1 \le k \le n, w(k) \in \{1,-1\}\} \tag{33.2}$$

The embedding algorithm is presented by using pseudo code as follows:

If $w(k) = 1 \& s < x(idx(i,j)) - x'(idx(i,j)) < t$

$x_w(idx(i,j)) = x'(idx(i,j)) + \alpha$

elseif $w(k) = -1 \& s < x'(idx(i,j)) - x(idx(i,j)) < t$

$x_w(idx(i,j)) = x'(idx(i,j)) - \alpha$

End

The weight factor α is defined as

$$\alpha = round\,[(s+t)/2] \tag{33.3}$$

From the above algorithm, it can be seen that if any conditions are not satisfied, the pixel $x(idx(i,j))$ is skipped, and the sequence idx of embedding locations is used for watermark extraction.

In the watermarking extraction process, the sequence of embedding location idx is used to extract the watermark. The reference image X'_w is obtained by performing the same procedure as that of the embedding process. According to the sequence of embedding location idx, the watermark can be extracted by comparing $x_w(idx(i,j))$ and $x'_w(idx(i,j))$. The extraction algorithm is presented as follows:

If $x_w(idx(i,j)) \geq x'_w(idx(i,j))$

$w'(k) = 1$

elseif $x_w(idx(i,j)) < x'_w(idx(i,j))$

$w'(k) = -1$

end

Here, x_w and x'_w denote pixel values of watermarked image and its reference image, respectively.

Experimental results show that the watermark scheme can resist JPEG compression, noising, cropping, and filtering. However, this watermark scheme is not a zero watermark scheme. Besides, Liu et al.'s scheme may not extract the embedded watermark completely under no attack since the reference image of watermarked image is not the same as the original one.

33.3 The Proposed Approach

The goal of the proposed scheme is to embed a robust watermark into the host image without modifying the host image. To achieve the goal, the reference image is used to calculate the polarity based on the mean of variances. Then, watermark data is embedded by using the polarity. In this section, the embedding and extraction process are illustrated in detail.

33.3.1 The Watermark Embedding Procedure

Step 1: Assume that the original image is a grayscale image of $w \times h$ pixels and watermark is a pseudo random bit stream. The original image X and watermark W are defined as follows.

$$X = \left\{ x_{i,j} | 0 \leq x_{i,j} \leq 255, 0 \leq i < w, 0 \leq j < h \right\} \tag{33.4}$$

and

$$W = \{ w_i | w_i \in \{-1, 1\}, 1 \leq i \leq n \} \tag{33.5}$$

In order to improve the security of our scheme, the permutation is implemented as follows. Firstly, each watermark bit in W is numbered from zero to $n-1$. Secondly, random numbers are generated by using a predetermined key, key_1. Finally, the permuted watermark W_p can be obtained by permuting W according to the random order.

$$W_p = permute(W, key_1) \tag{33.6}$$

Then, the permuted watermark W_p will be used to be embedded into the host image.

Step 2: The original image X is transformed into DWT domain by one-level wavelet transform and then three frequency subbands (LH_1, HL_1, and HH_1) are set to zero. Its reference image X' is obtained by performing inverse wavelet transform as shown in Fig. 33.1.

The reference image X' is divided into $\frac{w}{3} \times \frac{h}{3}$ non-overlapping image blocks B_k with size 3×3. The variance of each block is calculated by

$$\sigma^2(idx(k)) = \left(\frac{1}{9}\sum_{i=1}^{3}\sum_{j=1}^{3} b^2(i,j)\right) - \left(\frac{1}{9}\sum_{i=1}^{3}\sum_{j=1}^{3} b(i,j)\right)^2,\ 1 \le k \le \frac{w}{3} \times \frac{h}{3}, \tag{33.7}$$

where $idx(k)$ denotes the location information, and b denotes the pixel of block B_k.

Step 3: For a natural image, the neighboring pixels tend to be very similar and the closer the neighboring pixels, the higher the connection among them will be. The polarity P is generated by using this characteristic. After the average variance m of all blocks is calculated, the polarity based on the mean m and the predefined threshold T is constructed as follows:

$$p = \left\{ P(idx(k)) | P(idx(k)) \in \{-1, 1\},\ 1 \le k \le \frac{w}{3} \times \frac{h}{3} \right\}$$

where

$$P(idx(k)) = \begin{cases} 1,\ if\ \sigma^2(idx(k)) \ge m + T \\ -1,\ if\ \sigma^2(idx(k)) < m - T \end{cases} \tag{33.8}$$

Here, T is the weight factor for controlling the watermark strength, but it does not reflect the quality of stego-image at all.

Step 4: Among $idx(k)$, some locations are randomly selected for embedding. If ($W_p = 1$ and $P(idx(k)) = 1$), or ($W_p = -1$ and $P(idx(k)) = -1$), the location information $idx(k)$ is recorded. If the above conditions are not satisfied, the block is skipped. Finally, the sequence idx of embedding location will be collected for watermark extraction.

The flowchart of embedding procedure is shown as Fig. 33.3. The sequence idx of location information could be regarded as the secret key key_2 and send it to the legal receiver by a secure channel.

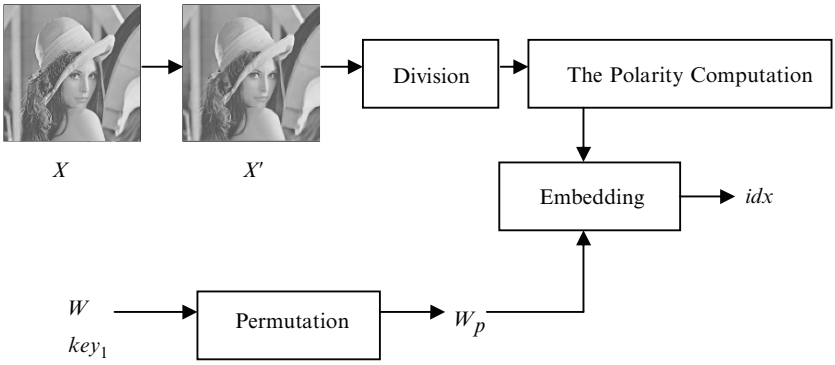


Fig. 33.3 Flowchart of the embedding procedure

33.3.2 The Watermark Extraction Procedure

Step 1: The watermarked image X_w is transformed into the DWT domain by one-level wavelet transform, and then three frequency subbands (LH_1, HL_1, and HH_1) are set to zero. The reference image X'_w is obtained by performing inverse
Step 2: The reference image X'_w is divided into $\frac{w}{3} \times \frac{h}{3}$ non-overlapping image blocks B'_k with size 3×3. Firstly, the variances $\sigma'^2(idx'(k))$ and the average variance m' are calculated as previously. Then the estimated polarity P' based on m' can be constructed by using Eq. (33.9).

$$P'(idx'(k)) = \begin{cases} 1, \ if \ \sigma'^2(idx'(k)) \geq m' \\ -1, \ if \ \sigma'^2(idx'(k)) < m' \end{cases} \tag{33.9}$$

Step 3: The sequence idx of the embedding location is used to extract the watermark data. If $P'(idx) = 1$, then the watermark $W'_p = 1$ is extracted. If $P'(idx) = -1$, then the watermark $W'_p = -1$ is extracted.
Step 4: Finally, the inverse permutation of Eq. (33.6) is performed to recover the watermark. The key key_1 is used to reverse-permute W'_p.

$$W' = inverse\_permute(W'_p, key_1) \tag{33.10}$$

The watermark extraction procedure is shown as Fig. 33.4. The receiver does not need the threshold T for extracting watermark.

33.4 Experimental Results

To evaluate the performance of the proposed method, two grayscale images (Lenna, and Baboon) are used to test the performance of our schemes in the experiments. The size of the grayscale image is 256×256. A pseudo-binary sequence is used as a

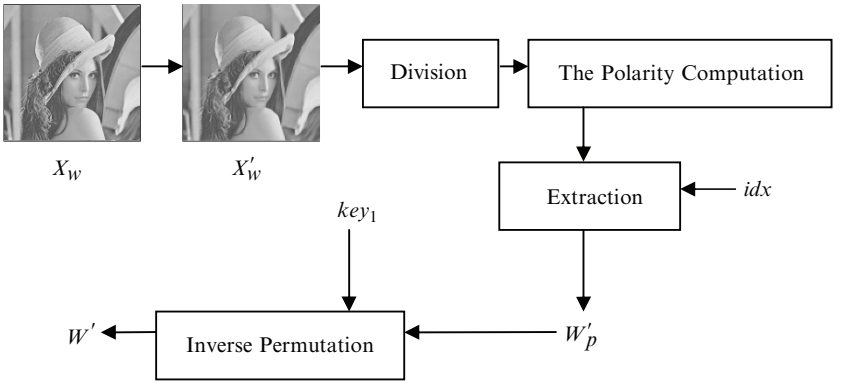


Fig. 33.4 Flowchart of the extraction procedure

watermark, and its length is 1000. The threshold T can be set appropriately by considering the robustness and embedding capacity. It is set to 114 in our experiments. The proposed algorithm is developed using MATLAB 6.5. PhotoImpact 11 and Stirmark are used to simulate typical numerical attacks and other image processing in our experiments.

In this chapter, the similarity measure is employed to evaluate the effectiveness of the proposed algorithm. The similarity between the original watermark W and the extracted one W' is defined as follows:

$$Sim(W,W') = \frac{W \cdot W'}{\sqrt{W' \cdot W'}} \tag{33.11}$$

If we acquire the higher similarity, it means the extracted watermark is more similar to the original one. Figure 33.5 shows the watermarked image and responses between the extracted watermark sequence and 499 other randomly generated watermarks. Responses show the extracted watermark only matches the original watermark clearly, and the highest similarity is 31.6228. The watermark can be extracted completely if the image suffers from no attack.

The performance of the proposed scheme is compared with that of Liu et al.'s scheme by performing several image manipulations and other attacks. Table 33.1 lists the comparison results. It can be seen that the embedding capacity and robustness of our scheme are evidently better than those of Liu et al., especially in JPEG compression. The similarity of our scheme in the low quality JPEG compression is higher than 26. Besides, other common attacks are tested in the following subsections.

33.4.1 JPEG Compression Attacks

In this experiment, JPEG compression with quality factors $QF = 80$, 50, 30, and 10 is applied on the watermarked image as shown in Table 33.2. Figure 33.6 shows

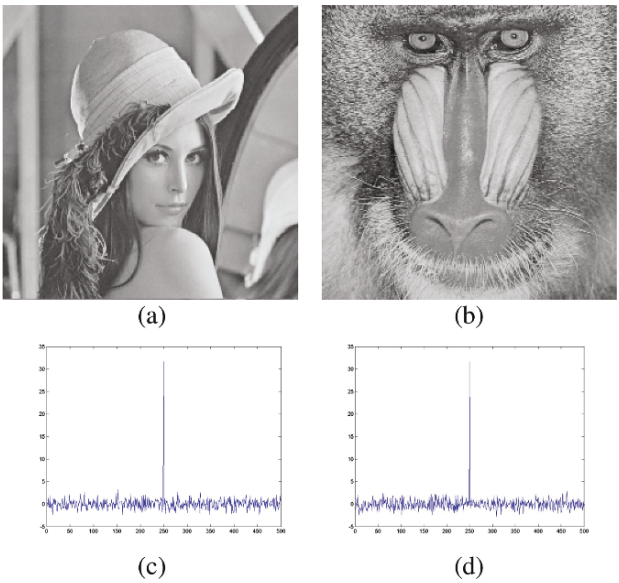


Fig. 33.5 The test images **(a)** and **(b)**; the similarities **(c)** 31.6228 and **(d)** 31.6228

Table 33.1 Comparisons between the proposed method and Liu et al.'s method

	Proposed method		Liu et al.'s method	
	Images			
Operations	Lenna	Baboon	Lenna	Baboon
No attack	31.6228	31.6228	31.24	31.37
JPEG(QF = 10)	29.4724	26.8161	17.9	19.98
Gaussian noise	27.4486	25.4880	16.76	17.7
Cropping(50)	14.0405	15.8114	13.22	16.95
Blurred	29.0930	19.3531	19.67	25.3
Sharpened	15.8114	12.5226	12.78	10.18

the images compressed by JPEG with $QF = 10$; the extracted watermarks still have high similarity—29.4724 and 26.8161, respectively. From these results, it can be seen that the proposed scheme is robust under JPEG compression. Additionally, JPFG2000 compression with compression ratios = 1 : 2, 1:8, and 1:20 is also performed to test the robustness of our method. The similarities of the extracted watermarks after JPEG2000 attacks are shown as Fig. 33.7 and Table 33.2. Experimental results demonstrate the extracted watermarks have high similarity—31.1168 and 20.2386—when the compression ratio is 1:20.

Table 33.2 Experimental results under various attacks

Attacks	Images	
	Lenna	Baboon
No attack	31.6228	31.6228
JPEG(QF=80)	31.6228	31.6228
JPEG(QF=50)	31.6228	31.4330
JPEG(QF=30)	31.5595	30.4211
JPEG(QF=10)	29.4724	26.8161
JPEG2000(cr=2)	31.6228	31.6228
JPEG2000(cr=8)	31.6228	29.2194
JPEG2000(cr=20)	31.1168	20.2386
Media filtering(5×5)	28.5237	15.7481
Scaling(0.75)	31.3698	29.9151
Scaling(0.50)	31.3065	28.9665

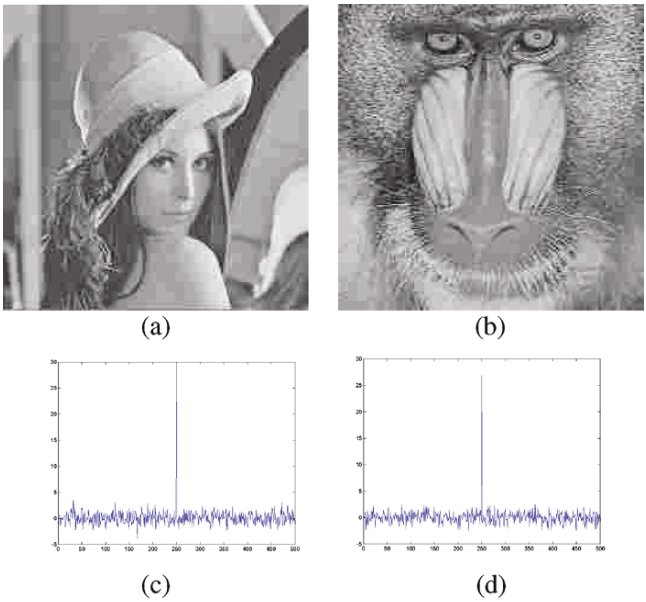


Fig. 33.6 JPEG compressed images with $QF = 10$ **(a)** and **(b)**; the similarities **(c)** 29.4724 and **(d)** 26.8161

33.4.2 Image Processing Attacks

Several image processing attacks, including Gaussian noise, image cropping, media filtering, blurring, and sharpening are performed on the watermarked image as shown in Figs. 33.8 through 33.12 and Tables 33.1 and Table 33.2. The watermarked image is also scaled down to evaluate the robustness of the algorithm. The results are given in Fig. 33.13 and Table 33.2. Additionally, multiple attacks are also tested.

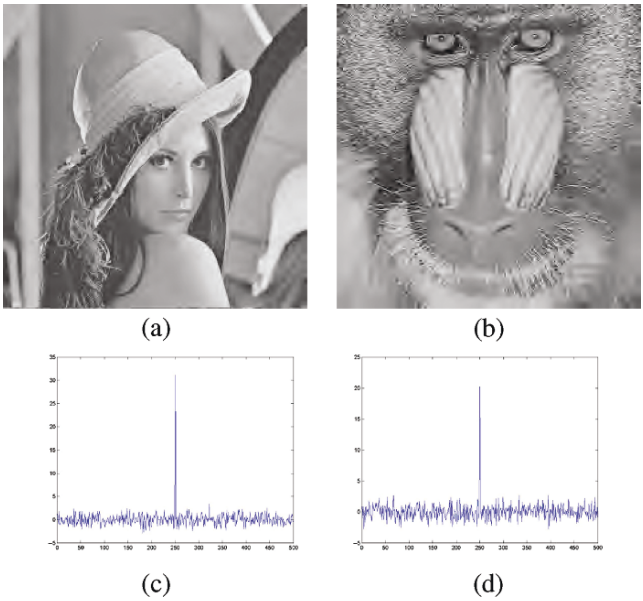


Fig. 33.7 JPEG2000 compressed images with a compression ratio = 1: 20 **(a)** and **(b)**; the similarities **(c)** 31.1168 and **(d)** 20.2386

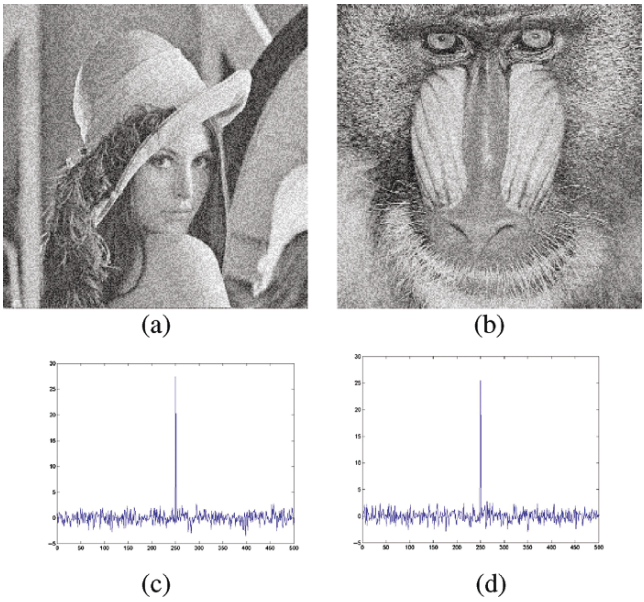


Fig. 33.8 Gaussian noising images with variance 15 **(a)** and **(b)**; the similarities **(c)** 27.4486 and **(d)** 25.4880

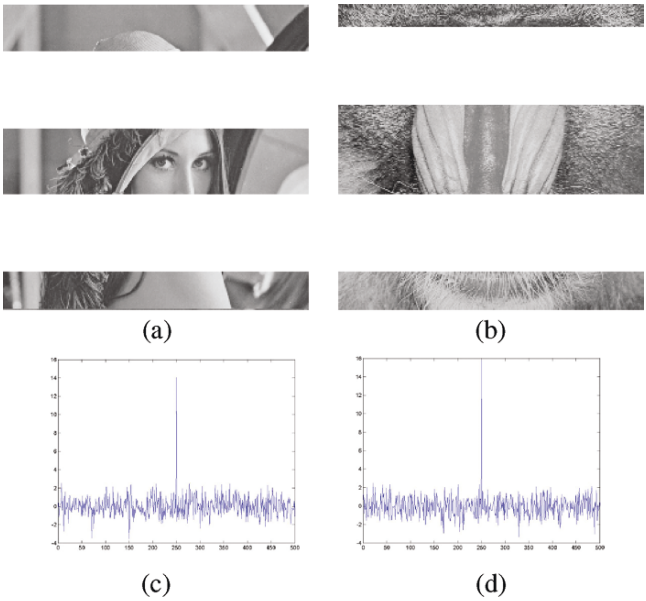


Fig. 33.9 50 percent cropped images **(a)** and **(b)**; the similarities **(c)** 14.0405 and **(d)** 15.8114

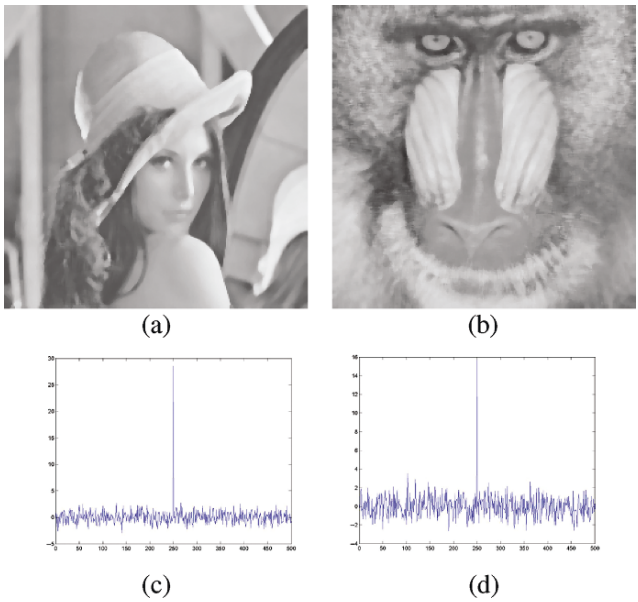


Fig. 33.10 Media filtering images with filter size 5 **(a)** and **(b)**; the similarities **(c)** 28.5237 and **(d)** 15.7481

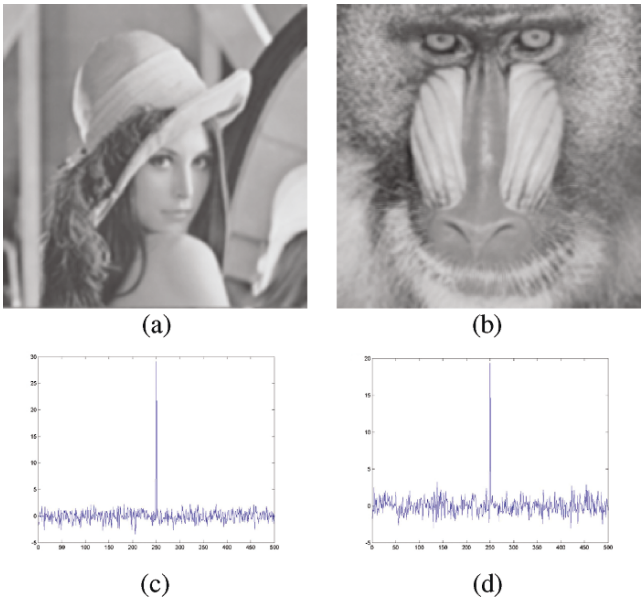


Fig. 33.11 Blurred images with a window size of 5×5 **(a)** and **(b)**; the similarities **(c)** 29.0930 and **(d)** 19.3531

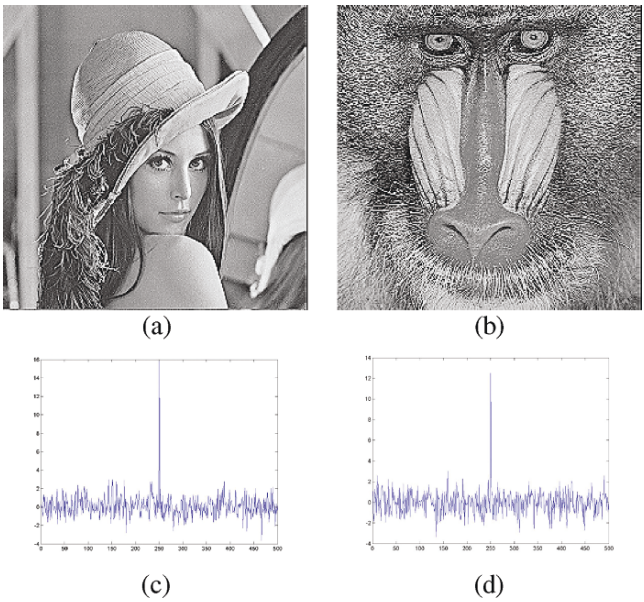


Fig. 33.12 Sharpened images with a level 5 **(a)** and **(b)**; the similarities **(c)** 15.8114 and **(d)** 12.5226

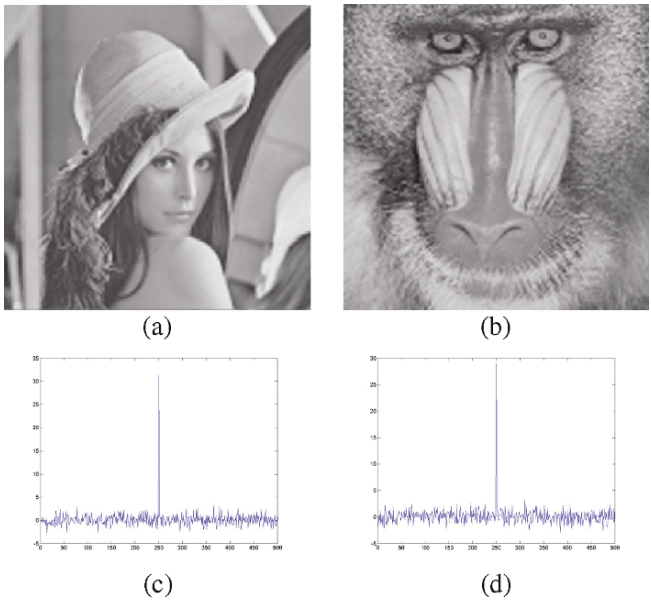


Fig. 33.13 The images with scaling factor 0.5 **(a)** and **(b)**; the similarities **(c)** 31.3065 and **(d)** 28.9665

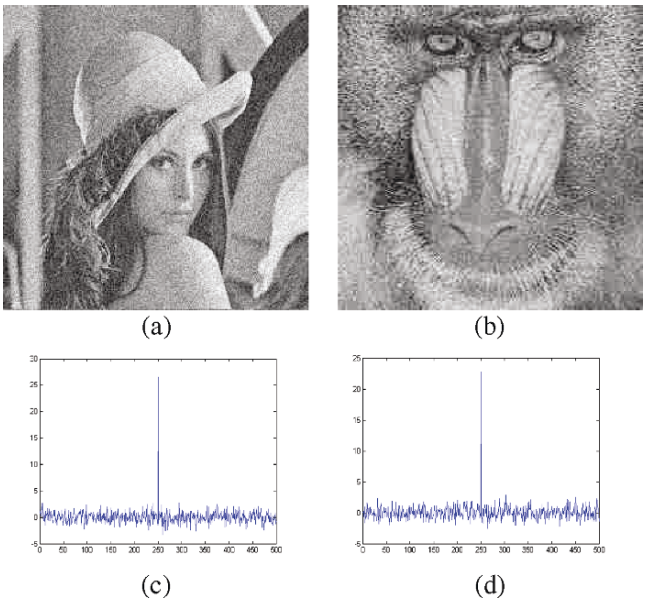


Fig. 33.14 The images with JPEG + Gaussian noise **(a)** and **(b)**; the similarities **(c)** 26.4366 and **(d)** 22.7684

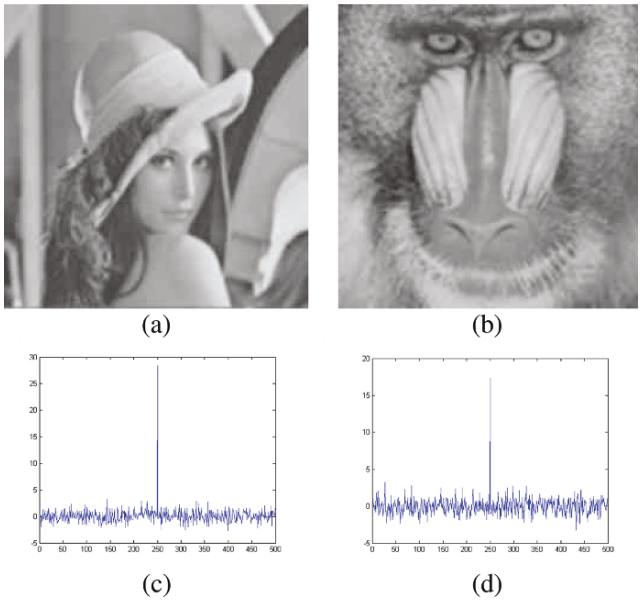


Fig. 33.15 The images with JPEG + blurring **(a)** and **(b)**; the similarities **(c)** 28.4605 and **(d)** 17.3293

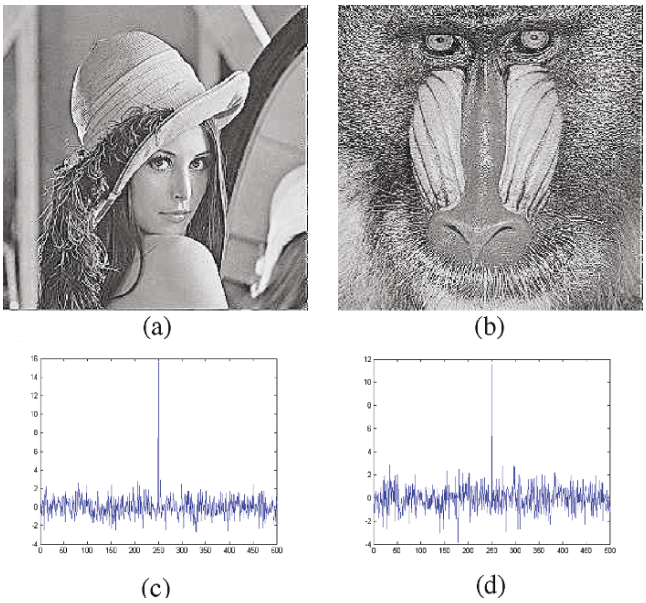


Fig. 33.16 The images with JPEG + sharpening **(a)** and **(b)**, the similarities **(c)** 15.6217 and **(d)** 11.5107

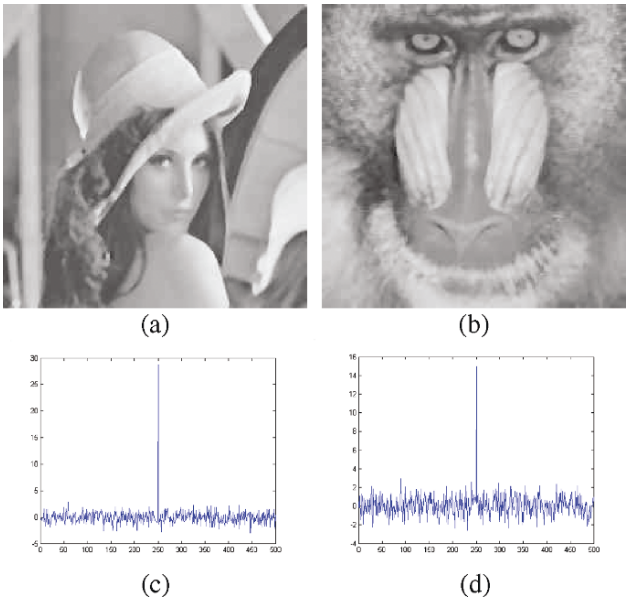


Fig. 33.17 The images with JPEG + media filtering **(a)** and **(b)**, the similarities **(c)** 28.1443 and **(d)** 14.9892

Table 33.3 Experimental results under multiple attacks

	Image	
Attacks	Lenna	Baboon
JPEG(30)+Gaussian noise	26.4366	22.7684
JPEG(30)+Blurring	28.4605	17.3293
JPEG(30)+Sharpening	15.6217	11.5107
JPEG(30)+Media Filtering	28.1443	14.9892

(JPEG with QF = 30) + (Gaussian noise, media filtering, blurring, sharpening) are applied to the test images as shown in Figs. 33.14 through 33.17 and Table 33.3. It can be seen that the embedded watermark can also survive these image processing attacks.

33.5 Conclusions

In this chapter, a zero-watermark scheme by using a reference image is proposed. By using the polarity based on the mean of variances, the proposed scheme can embed watermark data into the host image without modifying original image. Experimental results demonstrate the robustness of our method, especially in JPEG compression.

The proposed scheme has the following advantages. Firstly, the scheme is lossless without modifying the host image. It can be applied to some sensitive applications such as military and medical images. Secondly, the proposed scheme is secure. The watermark is permuted before embedding, and the embedding location information is unknown to the public.

References

1. Barni M, Bartolini F, Piva A (2001) Improved wavelet-based watermarking through pixel-wise masking. IEEE Trans. Image Processing 10(5):783–791
2. Bender W, Gruhl D, Morimoto N, Lu A (1996) Techniques for data hiding. IBM Syst. J 35(3&4):313–336
3. Chang CC, Tseng HW (2004) VQ-based image watermarking using anti-Gray coding. Informatica 15(2):147–160
4. Chen TH, Horng G, Lee WB (2005) A publicly verifiable copyright-proving scheme resistant to malicious attacks. IEEE Trans. Industrial Electronics 52(1):327–334
5. Chu WC (2003) DCT-based image watermarking using subsampling. IEEE Trans. Multimedia 5(1):34–38
6. Cox IJ, Kilian J, Leighton FT, Shamoon T (1997) Secure spread spectrum watermark for multimedia. IEEE Trans. Image Processing (12):1673–1687
7. Hsieh MS, Tseng DC, Huang YH (2001) Hiding digital watermarks using multiresolution wavelet transform. IEEE Trans. Industrial Electronics 48(5):875–882
8. Hsu CT, Wu JL (1999) Hidden digital watermarks in images. IEEE Trans. Image Processing 8(1):58–68
9. Huang HC, Wang FH, Pan JS (2002) A VQ-based robust multi-watermarking algorithm. IEICE Trans. Fund. E85-A(7)1719–1726
10. Joo S, Suh Y, Shin J, Kikuchi H (2002) A new robust watermark embedding into wavelet DC components. ETRI Journal 24(5):401–414
11. Liu JL, Lou DC, Chang MC, Tso HK (2006) A robust watermarking scheme using self-reference image. Computer Standards & Interface 28:356–367
12. Nafornita C, Isar A, Borda M (2005) Image watermarking based on the discrete wavelet transform statistical characteristics. EUROCON 2:943–946
13. Ni R, Ruan Q, Cheng HD (2005) Secure semi-blind watermarking based on iteration mapping and image features. Pattern Recognit 38:357–368
14. Nikolaidis N, Pitas I (1998) Robust image watermarking in the spatial domain. Signal Processing (3):385–403
15. Reddy AA, Chatterji BN (2005) A new wavelet based logo-watermarking scheme. Pattern Recognit. Lett. 26:1019–1027
16. Temi C, Choomchuay S, Lasakul A (2005) A robust image watermarking using multiresolution analysis of wavelet. International Symposium on Communications and information Technology (ISCIT) 1:623–626
17. Wang Y, Pearmain A (2004) Blind image data hiding based on self reference. Pattern Recognit. Lett. 25:1681–1689
18. Xie G, Shen H (2005) Toward improved wavelet-based watermarking using the pixel-wise masking model. IEEE Int. Conf. Image Process. 1:I-689–92
19. Yuan Y, Huang D, Liu D (2006) An integer wavelet based multiple logo-watermarking scheme. International Multi-Symposiums on Computer and Computational Sciences (IMSCCS) 2:175–179

Chapter 34
A Robust OCR for Degraded Documents

Kapil Dev Dhingra, Sudip Sanyal, and Pramod Kumar Sharma

34.1 Introduction

In the last two decades, many advances have been made in the field of document image analysis and recognition. In the recent past, several methods for recognizing Latin, Chinese, Japanese, and Arabic scripts have been proposed [7–9]. Until now, most of the OCR work has concentrated on high quality images and great success has been achieved by character recognition systems. Apart from these successes, there still exist two challenging problems in the field of recognition. The first one is optical character recognition (OCR) for low-quality images. Images having luminance variations, noise, and random degradation of text are difficult to read by OCR systems. The second open problem is that of recognizing off-line cursive handwritten character recognition [15]. Our work concentrates on the former one particularly for Devanagari script, which is the script for Hindi, Nepali, Marathi, and several other Indic languages. Together, these languages have a user base exceeding 500 million people.

A great deal of effort has been made towards the development of OCR for Indian scripts [1–3, 10]. This chapter is concerned with the recognition of degraded Devanagri text documents. A major contribution in the area of Devanagari OCR is the Hindi OCR system developed by Chaudhari [2]. As remarked earlier, a majority of the above mentioned work has concentrated primarily on good quality document images and little work has been reported so far for the development of OCR for degraded Devanagari document images. Jawahar [4] presented a scheme based on SVM classifier, but this work was not focused on degraded document images. A major contribution towards the development of OCR for degraded documents is OCR for Chinese characters [13]. Apart from the advances that have been made towards

Kapil Dev Dhingra, Sudip Sanyal, and Pramod Kumar Sharma
Indian Institute of Information Technology Allahabad, Universal Digital Library Research Lab, U.P, India - 211012
e-mail: kapil@ug.iiita.ac.in,ssanyal@iiita.ac.in,pramod@ug.iiita.ac.in

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

the development of Devanagri character recognition system (DCRS) [2], the development of DCRS for degraded text documents is still a challenge. The difficulties faced are from the following aspects: the books and documents scanned are very old and have been degraded due to various factors. They have been printed using primitive print technology. Moreover, the paper used for printing was of very low quality, which, after a considerable amount of time, starts to tear off. It also loses color and adds pores. The prime source of noise in old text documents is blotting or spreading of the ink due to the porous quality of the paper. The poor quality of ink used for printing leads to blotting of text in these documents, which results in character distortion. Examples of such characters are shown in Figs. 34.1 and 34.2. Due to noise introduced in documents by above mentioned reasons, general OCR on such documents produces poor results.

As explained in Sect. 34.2, the Indian languages in general, and the Devanagari script in particular, have a large number of characters. This is due to the use of vowel modifiers and conjuncts. Quite often, the number of classes is more than 400. This poses a serious challenge, particularly when we include additional complications like degradation and font styles.

In this chapter, we have presented an approach for the development of MCE classifier-based DCRS. Features used for classification are directly extracted by Gabor filters. Gabor filters have been successfully applied to Chinese OCR [13]. The MCE-based classifiers provide robustness to the system against random noise by adjusting the system feature space according to the loss function computed. We used the degradation model [16] to simulate the distortions caused due to the imperfections in scanning.

This chapter is organized as follows. In Sect. 34.2 we provide a brief overview of Devanagari script and explain the origin of the large number of classes. Section 34.3 gives the outline of the character extraction process from Devanagari documents.

Fig. 34.1 (a) Ideal character, and (b) character sample from our document image

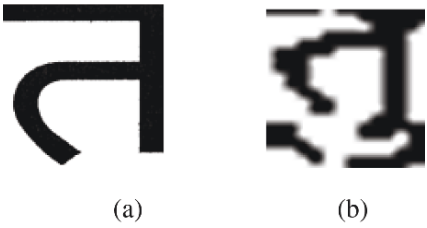


Fig. 34.2 (a) Full character, and (b) full character sample, with lower modifier, taken from degraded text document

In Sect. 34.4 we have described Gabor filters and their implementation in our OCR system. Section 34.5 includes the background and implementation of MCE classifier in our character recognition system. In Sect. 34.6, we present our results derived from various experiments. Section 34.7 includes the summary and conclusion of our work.

34.2 Brief Overview of Devanagri Script

Devanagari script is written in left to right and top to bottom format. It consists of 11 vowels and 33 basic consonants. Each vowel except the first one has corresponding modifiers that are used to modify a consonant. Thus, in addition to the original set of consonants, we also have consonants modified by the succeeding vowel. Some of these vowel modifiers can be extracted during the segmentation process, but others are difficult to segment. Thus, in those cases where we cannot segment the vowel modifier, we get a new class for our classification system.

In addition to the vowel modifiers, a consonant can also get modified by another consonant. These are called *conjuncts*. This occurs when we have two consonant sounds coming together without an intervening vowel sound. In these cases, only half of the first consonant is written to the left of the second consonant. Depending on the font style, it may or may not be possible to segment the half and full characters. Thus, the half characters, as well as the one-and-half characters, have to be treated as individual classes, in addition to the full characters. This leads to a very large number of classes.

All words in Devanagari script have a continuous line of black pixels for the whole word. This line is called Shirorekha. Based on Shirorekha, each character can be divided in to three parts. The first part includes components above Shirorekha that are called Upper modifiers. The second part includes characters that can be either full, half, or joint. In the third part, there are modifiers of vowels called Lower modifiers. Devanagari varnmala is shown in Fig. 34.3. As will be explained in Sect. 34.3, we can use the Shirorekha to segment words. However, segmenting characters creates problems, particularly when there is a degradation of the documents. We shall describe the preprocessing steps and the segmentation process in the next section.

34.3 Character Extraction from Scanned Document

Efficient preprocessing of scanned documents is necessary to improve recognition. Because most of these steps are fairly standard, we shall briefly describe the preprocessing and segmentation steps in this section.

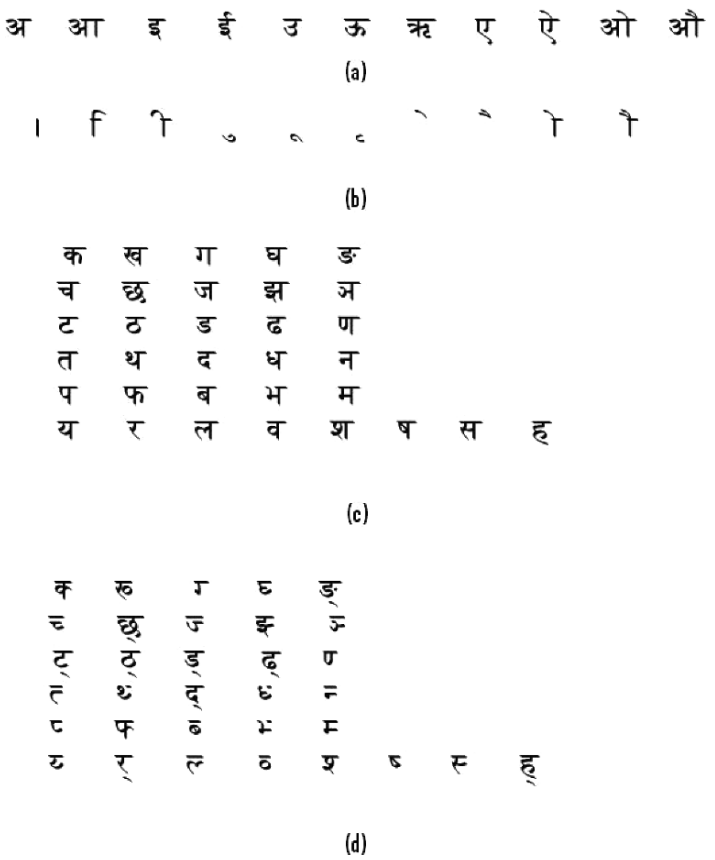


Fig. 34.3 (a) Vowels, (b) modifiers, (c) full character, and (d) half characters

34.3.1 Preprocessing Steps

Document pages used in our system were scanned using a flat-bed scanner at 600 dpi. In the first step of preprocessing process, median filtering is applied to the scanned documents. This removes some of the noise, particularly of the salt and pepper variety. The second step constitutes binarization. Binarization is performed by obtaining the histogram of the page finding suitable threshold values. After binarization, the scanned documents are skew corrected. Our system performs skew correction in the range of $+/-20$.

34.3.2 Segmentation

To extract individual characters from scanned documents, line segmentation and then word segmentation are carried out. Line segmentation can be done efficiently

and accurately using vertical profiles of the rows and finding the zeros in the profile. As mentioned in the previous section, the Shirorekha provides an easy tool for recognizing the word boundaries. The vertical profile of the segmented lines has a clear white run length at the ends of the words. Thus, word boundaries can be detected easily.

The character level segmentation is performed next. Details of the segmentation method can be seen in Chaudhuri and Pal. The method they suggest is to look at the horizontal profile of the word and identify the rows corresponding to the Shirorekha. The pixels corresponding to the Shirorekha are *stripped off*. Thus individual characters form separate connected components and can be segmented using a variety of methods like vertical profiling or connected component labeling. While the described method works well for non degraded texts, it fails for our cases. Due to blotting of text full character often gets overlapped with its lower modifier, as can be seen in the case of Fig. 34.2 above. The overlapping is very complex by nature, because it is random. Moreover, the extent of overlapping also depends on the font style. Thus the segmentation process fails to separate the character and modifier as well as the half characters that are joined with the full characters. This can become a major cause of OCR errors. Therefore, in our DCRS, we include all possible combinations of full characters and their lower modifiers as separate classes. While this reduces the segmentation errors, it results in a large number of target classes.

To handle the case of a large number of classes, we needed a high dimensional feature space. The high dimensionality of our feature space reduces the possibility of overlapping among classes. Each of the extracted characters is normalized to a standard size after segmentation. We obtain our high dimensional feature space using Gabor filters, which are described in the next section.

34.4 Gabor Filters

Gabor filters have been used successfully in texture analysis and texture segmentation, edge detection, and many pattern recognition applications such as face recognition [11] and finger print recognition. In our case, some noise-resilient method was required to deal with degraded text documents and Gabor filters are found to be least sensitive to noise, scaling, and rotation [5]. Gabor feature-based extraction has also been applied to character recognition [13]. However, it will be time inefficient in case of high-varying noise, as in the case of ours.

The two-dimensional (2D) Gabor filter is a complex sinusoidal modulated Gaussian function in spatial domain. It is given by

$$h(x,y;\lambda,\phi,\sigma_x,\sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left\{-\frac{1}{2}\left[\frac{V_1^2}{\sigma_x^2}+\frac{V_1^2}{\sigma_y^2}\right]\right\}\times\exp\left\{i\frac{2\pi V_1}{\lambda}\right\} \quad (34.1)$$

where,

$V_1 = x\cos\phi + y\sin\phi$

$V_2 = -x\sin\phi + y\cos\phi$

In spatial frequency domain, it is given by

$$H(u,v;\lambda,\phi,\sigma_x,\sigma_y) = K\exp\left\{-2\pi^2\left[\sigma_x^2\left(F_1 - \frac{1}{\lambda}\right)^2 + \sigma_y^2(F_2^2)\right]\right\} \tag{34.2}$$

where,

$F_1 = u\cos\phi + v\sin\phi$

$F_2 = -u\sin\phi + v\cos\phi$, K = constant

If $I(u,v)$ is the Fourier transform of the original image, then we define

$$f(x,y) = FFT^{-1}[I(u,v)\cdot H(u,v)] \tag{34.3}$$

The flexibility of Gabor filters lies in the fact that we can perform the convolution using various choices of scale (like wavelets), as well as orientations. Moreover, we can center the Gabor function at various points on the image. In our experiments, we used eight orientations and two scales to cover the spatial frequency plane. Thus, if we scale the image to a size of 20×20, we have a feature space with a dimensionality of $20 \times 20 \times 2 \times 8$. This high dimensionality means that the classes are likely to be well-separated in the feature space. After obtaining the feature vectors of the samples, we next need to perform the classification. This is described in Sect. 34.5.

34.5 Minimum Classification Error Technique

To provide stability and robustness to the classifier, against random noise present in sample training characters, we implemented the minimum classification error (MCE)-based technique. The technique has been primarily established in the automatic speech recognition (ASR) area. It has been also studied in the OCR area [6]. Recently, it has been implemented in online handwriting recognition [14]. This technique finds the discriminant functions for each of the classes. The discriminant function was designed using a simple approach to make the system computationally efficient, and also to make it stable against the random noise.

Let us assume N character classes C_i, thus, $\{C_i\}_{i=1}^{N}$. Each class C_i is being trained by K training samples, represented as X_{ik}. Thus, $\{X_{ik}\}_{k=1}^{K}$ represents the training set. After Gabor feature extraction, each training sample X_{ik} can be represented by a feature vector m_{ikd} of length D—i.e., $\{m_{ikd}\}_{d=1}^{D}$ where $D = \lambda \times \phi \times M \times M$. Here, λ and ϕ are the total number of scales and orientations, respectively, used in Gabor filtering. $M \times M$ is the dimension of training samples. For each character class C_i we define a mean vector, V_i, such that

$$V_i = \frac{\sum_{k=1}^{K} X_{ik}}{K} = \frac{\sum_{k=1}^{K}\sum_{d=1}^{D} m_{ikd}}{K} \tag{34.4}$$

For pattern classification, a discriminant function is evaluated. It can be defined as the distance between X_{ik} and V_i. One, of course, has a choice of distance metric, and the accuracy of the final classifier is strongly dependent on this choice. In Sect. 34.6 we shall give results for specific choices of the metric. If we use the Manhattan distance metric, the discriminant function can be defined as

$$g_n(X_{ik}, V_n) = |V_n - X_{ik}| = \sum_{d=1}^{D} abs(m_{ikd} - V_{nd}) \tag{34.5}$$

When discriminant functions are used, the classification rule is defined as: The class that gives the maximum value of the discriminant function for a given sample X_{ik} is considered to be the recognized class—i.e.,

$$X \in C_j \; if \; j = argmax \; g_n(X_{ik}, V_n) \tag{34.6}$$

On the basis of the definition of discriminant function in Eqn. 34.5 and the classification rule in Eqn. 34.6, we can define a misclassification measure [12] for each training sample X_{ik} as

$$d_{ik}(X_{ik}, V_i) = \frac{g_i(X_{ik}, V_i) - g_q((X_{ik}, V_q)}{g_i(X_{ik}, V_i) + g_q((X_{ik}, V_q)} \tag{34.7}$$

where

$q = argmax \; g_n(X_{ik}, V_n)$ *and* $q \neq i$

We can now define the loss function for each training sample as

$$L(X_{ik}) = \frac{1}{1 + \exp -\alpha(d_{ik}(X_{ik}, V_i) + \beta)} \tag{34.8}$$

where α and β are two control parameters. We will update m_{ik} and m_{qk} according to the following rules

$$m_{ik}^{t+1} = m_{ik}^{t} - \varepsilon_t \upsilon \frac{g_i}{(g_i + g_q)^2}(m_{ik}^{t} - V_i^{t}) \tag{34.9}$$

$$m_{qk}^{t+1} = m_{qk}^{t} - \varepsilon_t \upsilon \frac{g_q}{(g_i + g_q)^2}(m_{qk}^{t} - V_q^{t}) \tag{34.10}$$

where

$g_i = g_i(X_{ik}, V_i^t)$,

$g_q = g_q(X_{qk}, V_q^t)$,

$V_i^t = \frac{\Sigma_{k=1}^{K} X_{ik}}{K}$,

$\upsilon = L(X_{ik})(1 - L(X_{ik}))$

Furthermore, the index t in the superscript of the relevant variables in the above equations represents the cumulative number of training samples presented so far. One iteration of all the training samples is called an *epoch*. The learning rate ε_t is

$$\varepsilon_t = \varepsilon_o \left(1 - \frac{t}{N_{total} * N_{epoch}}\right) \tag{34.11}$$

where N_{total} is the total number of training samples—i.e., $N \times K$—and N_{epoch} is the total number of epochs. ε_o is the initial learning rate that will be evaluated from the experiments.

34.6 Experiments and Results

As is true in most pattern recognition tasks, there are a number of parameters that have to be fixed in order to get a good recognition system. In this section, we describe the experiments performed to fine-tune our system and then present a comparison of the proposed system with that of other methods. In our experiments, 20,000 training and 10,000 test characters were used. These characters include joint, single, half, and lower modifier characters equally.

34.6.1 Experiment on Distance Metrics

In our first experiment, we used a simple centroid-based classifier and compared its classification efficiency on Euclidean and Manhattan distance metrics. In centroid-based classifiers, each class forms a separate region in the feature space. Distance between any two classes is defined by the distance between their centroids. To measure the classification efficiency of a centroid-based classifier on a certain distance metric, the following method was used: Ratio R_{ij} is defined between classes i and j as

$$R_{ij} = \frac{D_{ij}}{R_i + R_j} \tag{34.12}$$

where D_{ij} is the distance between the centroids V_i and V_j of classes i and j. R_i and R_j are the radii of the classes i and j, respectively. The radius of a class is defined as the maximum distance of the training samples from the centroid. From Eqn. 34.12, if R_{ij} is greater than 1, then classes i and j do not overlap in the feature space and will be successfully classified by the classifier on the given distance metric. If R_{ij} is less than 1, then the classes overlap and the classifier may fail to classify i and j on the given distance metric.

From the experiment, we found that the Manhattan distance metric is more efficient than the Euclidean for classification. The results of using these two distance metrics are given in Table 34.1. The average ratio column in Table 34.1 gives the average ratio, R_{avg}, of all the ratios R_{ij} obtained using the corresponding distance metric for 250 classes.

34.6.2 Experiment on Character Image Scales

In this experiment, we evaluated the character recognition efficiency of DCRS on various character image scales. Character images were scaled and passed as the input to DCRS in each iteration. The corresponding character recognition rate of DCRS at that scale was computed. The experiment was done on centroid-based classifiers and involves the same training and test sets that were used in the first experiment. The best recognition rate was obtained at character image dimension of 20×20 as shown in Fig. 34.4. Thus, we found that just by optimizing the scale of character images to an optimum scale, we can boost the character recognition rate of the OCR.

Table 34.1 Classification efficiency on different distance metrics

Distance Metric	Average Ratio (R_{avg})	% of Classes having $R > 1$
Euclidean	0.72	41
Manhattan	1.05	76

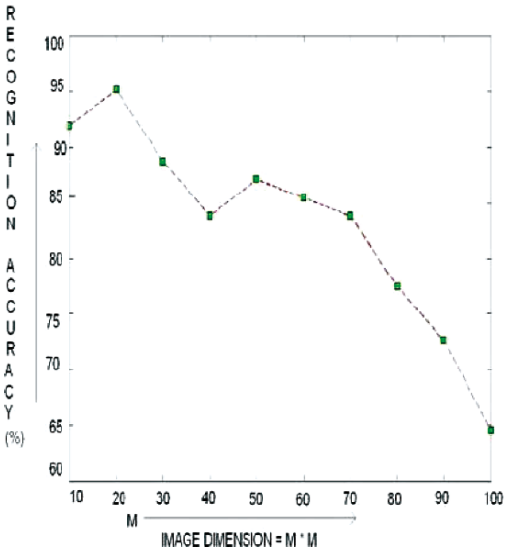


Fig. 34.4 Character Recognition accuracy (%) as a function of Character Image Dimension ($M \times M$)

34.6.3 Experiment on Classifiers

We improved the efficiency of our classifier by use of the MCE-based classification technique as compared to the centroid-based method used in the first two experiments. The training set was the same as that used in above experiments. The character images of the training set were scale normalized to 20×20. The parameters α and β of Eqn. 34.8 were both set to 1, and the total number of epochs, N_{epoch} of Eqn. 34.11 was set to 15. From the experiment, DCRS obtained higher character recognition rate and more robustness. A comparison of the centroid-based classifier used in our previous experiments, the MCE classifier as applied above KNN and SVM classifiers tested on degraded Hindi text documents [4] is shown in Table 34.2.

34.6.4 Experiment on Various Degradation Levels

As a final experiment, we tested the accuracy of our classifier on various levels of degradation. A controlled amount of degradation was obtained using a ground truth tool. The tool introduces noise using Kanungo model [16]. Further details about the tool can be seen in Appendix 34.7. Approximately 5,000 training and 2,000 test sets were generated for each degradation range being tested. The same degradation model is applied both to the training and test sets. The size 2 disk was used for flipping the pixel values with a probability threshold of 10^{-4}. Performance of OCR on various degradation levels is shown in Table 34.3. As can be seen from the results, the performance of our OCR degrades gradually with increasing degradation. However, even when the level of degradation is very high—i.e., at the 20–40 level—there

Table 34.2 Character recognition rate on different classifiers

Classifier	Character Recognition Rate (%)
Centroid-based	95
MCE	98.5
KNN	94.6
SVM	97.6

Table 34.3 Character recognition rate on various degradation levels

Degradation Range (%)	Character Recognition Rate (%)
10–35	98.2
15–30	95.8
20–40	86.66

is a significant accuracy. In practical situations, the type of degradation observed in documents that are around 60 years old was similar to the synthetic degradation produced by the ground truth tool when the degradation parameter is in the range of 10–35. As can be seen, the recognition rate is above 98 percent for these cases.

34.7 Conclusion and Future Work

In this chapter, an MCE-based classifier is presented to recognize degraded documents. Experiments and results show that the proposed system produces better results as compared to the KNN and SVM classifiers. The future extensions are to make the system robust so that it can read characters in different fonts and styles. Even though the work presented in this document is focused on degraded documents for Devanagari script, the proposed method can be easily adapted to recognize printed text documents from other scripts. Moreover, the use of Gabor filters results in a very high dimensional feature space that is able to accommodate a large number of classes that are typical of many languages of Indian origin.

Acknowledgements The authors gratefully acknowledge financial assistance from the Ministry of Communication and Information Technology, Government of India.

Appendix 1: Synthetic Character Degradation

A ground truth tool was developed to model real world degradation similar to degradation found in old books. The tool is based on a stochastic document degradation model that closely resembles the model of Kanungo [16]. The prime characteristic of the model is that it simulates aging and ink spreading mechanisms. This closely represents the degradation found in the old documents that were used in our experiments.

The degradation model has six parameters $(\eta, \beta, \alpha, \alpha_0, \beta_0, k)$. These parameters are used to degrade an ideal binary image as follows:

1. Compute the distance d of each pixel from the character boundary.
   Flip each foreground pixel with a probability,

$$p(0|1, d_r, \alpha_0, \alpha) = \alpha_0 \times \exp -\alpha (d_r)^2 + \eta \tag{34.13}$$

   Flip each background pixel with a probability,

$$p(1|0, d_r, \beta_0, \beta) = \beta_0 \times \exp -\beta (d_r)^2 + \eta \tag{34.14}$$

2. Divide the pixel-inverted image into small equal-sized subimages.

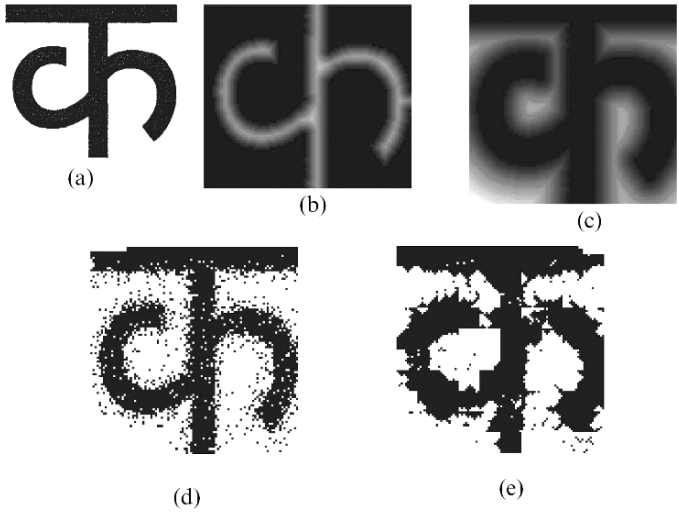


Fig. 34.5 (a) Ideal noise free Devanagri character image, (b) distance transform of foreground, (c) distance transform of background, (d) pixel inverted image, and (e) final synthetic degraded image with parameters (0,1.5,2.0,2.5,2.0,2) and ink spreading 80%, and aging of 20%

3. Perform a morphological closing and opening operation on these sub-images based on user-defined ink spreading and aging effects with a disk-structuring element of diameter k where:

$\alpha_0, \beta_0 =$ Initial values for exponentials
$\alpha, \beta =$ Control the decay speed of exponentials
$\eta =$ Constant probability of flipping for all pixels
$d =$ Closest distance of pixel from character boundary, computed using distance transform algorithm [18]
$d_r =$ Randomly generated distance within the neighborhood of d, with the neighborhood depending on the size of the image.

These steps are illustrated in Fig. 34.5.

References

1. Bansal V, Sinha RMK (2001) A Devanagari OCR and a brief review of OCR research for Indian scripts. Proceedings of STRANS01
2. Chaudhari BB, Pal U (1997) An OCR system to read two Indian Languages scripts. Proc. of 4th Int. Conf. on Document Analysis and Recognition, 1011–1015
3. Atul Negi, Chakravarthy Bhagvati, Krishna B (2001) An OCR System for Telugu, ICDAR, 1110
4. Jawahar CV, Pavan Kumar MNSSK, Ravi Kiran SS (2003) A Bilingual OCR for Hindi-Telugu Documents and its Applications, ICDAR. 408–412

5. Xuewen Wang, Xiaoqing Ding, Changsong Liu (2002) Optimized Gabor Filters Based Feature Extraction for Character Recognition, Proc.16th International Conference on Pattern Recognition, 223–226
6. Qiang Huo, Yong Ge and Zhi –Dan Feng, (2001) High Performance Chinese OCR Based on Gabor Features, Discriminative Feature Extraction and Model Training. Proc. IEEE International Conference on Accoustic, Speech and Signal Processing, 1517–1520
7. Mantas J (1986) An Overview of Character Recognition Methodologies, Pattern Recognition 19:425–430
8. Bozinovic RM, Srihari SN (1989) Offline Cursive Script Word Recognition. IEEE Trans on Pattern Analaysis and Machine Intelligence 11:68–83
9. Mori S, Suen CY, Yamamoto K (1992) Historical Review of OCR Research and Development. Proc. of IEEE 80:1029–1058
10. Nagy G (2000) Twenty Years of Document Image Analysis in Pattern Analysis and Machine Intelligence. IEEE Trans. on Pattern Analysis and Machine Intelligence 22:38–62
11. Zhang J, Yan Y, Lades M (1997) Face recognition: Eigenface, Elastic Matching, and Neural Nets. Proc. of IEEE 85:1423–1435
12. Juang BH and Katigiri S (1992) Discriminative Learning for Minimum Error Classification Paper Title. IEEE Trans. on Signal Processing 4:3043–3054
13. XuewenWang, Xiaoqing Ding and Changsong Liu (2005) Gabor Based Feature Extraction for Character Recognition. Pattern Recognition 38:369–379
14. Alain Biem (2006) Minimum Classification Error Training for Online Handwriting Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 28:1041–1051
15. Plamondon R, Srihari SN (2000) On-line and Off-line Handwriting Recognition: A Comprehensive Survey. IEEE. Trans. on Pattern Analysis and Machine Intelligence 22:63–84
16. Kanungo T. et al. (2000) A Statistical, Nonparametric Methodology for Document Degradation Model Validation. IEEE Trans. on Pattern Analysis and Machine Intelligence 20:1209–1223
17. Chaudhuri BB and Pal U (1997) Skew Angle Detection of Digitized Indian Script Documents. IEEE Trans. on Pattern Analysis and Machine Intelligence 19:182–186
18. Maurer CR, Qi R, Raghavan V (2003) A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions. IEEE Trans. on Pattern Analysis and Machine Intelligence 25:265–270

Chapter 35
Distortion Tolerant Color Pattern Recognition by Using Mach-Zehnder Joint Transform Correlator with Image Encoding Technique

Chungcheng Lee and Chulung Chen

35.1 Introduction

During the past decade, a liquid crystal spatial light modulator (LCSLM) being researched in optical pattern recognition has aroused lots of attention. We can perform a Fourier transform in parallel and real-time easily with an LCSLM and a lens, which is carried out based on the correlation. Furthermore, the VanderLugt (or matched filter based) correlator (VLC) [11] and the joint transform correlator (JTC) proposed by Weaver and Goodman [12] have been widely used for the optical pattern recognition systems originally. The basic distinction between these techniques is that the VLC depends on Fourier domain matched filtering, whereas JTC depends on spatial domain filtering. Compared with VLC, JTC does not have the stringent filter-alignment problem. Consequently, it is more suitable for real-time processing and more robust in term of environment disturbances. Hence, JTC has become a hot research field in recent years.

One of the main deficiencies in classical JTC (CJTC) is that the joint transform power spectrum (JTPS) has a spatially varying average which leads to a large zero order term. The desired cross-correlation peaks were almost overshadowed and the output detector was often saturated by the auto-correlation term. Accordingly, several research groups have studied how to improve the performance of JTC [1, 7, 15]. To reduce or eliminate the zero order diffraction and enhance the recognition ability, Lu et al [9] applied a phase-shifting technique to the CJTC and Li et al. [8] proposed the joint transform power spectrum (JTPS) subtraction strategy to accomplish the nonzero order JTC (NOJTC). However, the removal of the zero order term needs two or more time-consuming steps in the preprocessing of the input objects before applying the second Fourier transform operation. In this work, a Mach-Zehnder based NOJTC (MZJTC) proposed by Cheng and Tu [2] was utilized to remove all

Chungcheng Lee and Chulung Chen
Department of Electrical Engineering, Yuan Ze University, 135 Yung Tung Road, Taoyuan 32026, Taiwan, Republic of China

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

unneeded peaks at the output plane. It can directly remove the zero order spectrum in one step. Recently, in order to reduce the correlation sidelobes as low as possible for a rotated image, Chen et al. [3, 4, 13] adopted the minimum average correlation energy (MACE) method [10] based on the Lagrange multipliers to design a reference function that can result in sharp correlation peaks.

In this study, we use an MZJTC with an image encoding technique for performing color pattern recognition. The concept of image encoding led to the evolution of optical pattern recognition. In order to obtain a novel encoded pattern, we utilize recombination of red, green, and blue grayscale images in the input image. Moreover, the encoded pattern will be synthesized into the reference function by using the MACE impulse response. Additionally, the multichannel single-output NOJTC [5] system has many advantages over other classical systems, especially with distortion tolerant capability [14]. The technique also can yield sharp correlation peaks. Sequentially, the system can deal with detection of multiple targets with distorted target and non-target images. Furthermore, the test image is added by some random noise. These noisy images are used to test this system for noise ability. Numerical results are shown to illustrate the performance of this system. Finally, a hybrid optical fulfillment of the proposed system has excellent recognition ability.

35.2 Analysis

The implementation setup of the MZJTC system with interlaced scanned color components is shown in Fig. 35.1. It consists of a laser as a source, 2 beam splitters (BSs), 3 liquid crystal spatial light modulators (LCSLMs), 3 Fourier lenses (FLs), 3 CCD cameras, 1 electronic subtractor (ES) to remove the zero order term of JTPS, and a personal computer (PC) to control the whole system.

The PC transfers grayscale images of red, green, and blue color components of both the reference image and the target image in the input plane through LCSLM1 and LCSLM2, respectively. The LCSLM1 and LCSLM2 can display the grayscale images of color components for the reference encoded image and the target encoded image, which are denoted by $h(x+a,y)$ and $e(x-a,y)$, respectively. a is the distance of the reference channel and target channel away from the vertical central lines of the LCSLM.

We propose a novel space arrangement method with image encoding technique for color images. In this method, each color target image is transformed to a grayscale image from red, green, and blue color components, as shown in Fig. 35.2, which is displayed on a single, monochromatic input plane.

The image encoding process is listed as follows:

1.

$$\text{Let } r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}, g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}, \text{and } b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \tag{35.1}$$

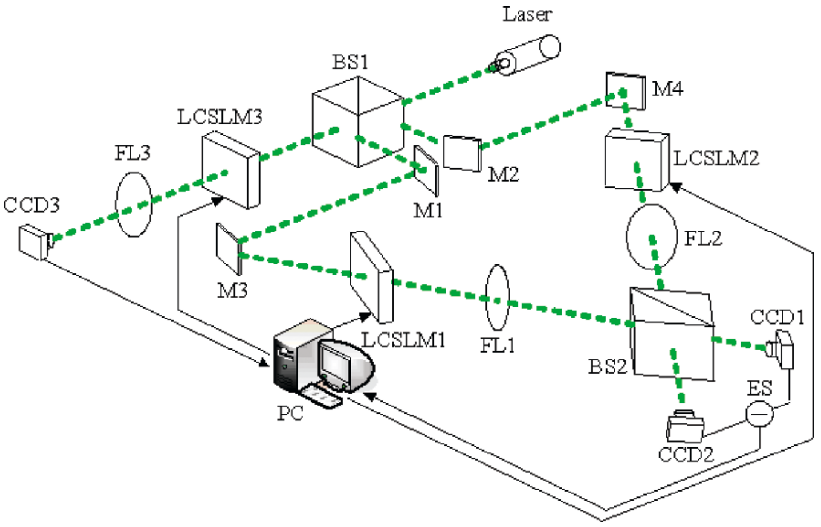


Fig. 35.1 Implementation setup of a MZJTC system with interlaced scanned color components

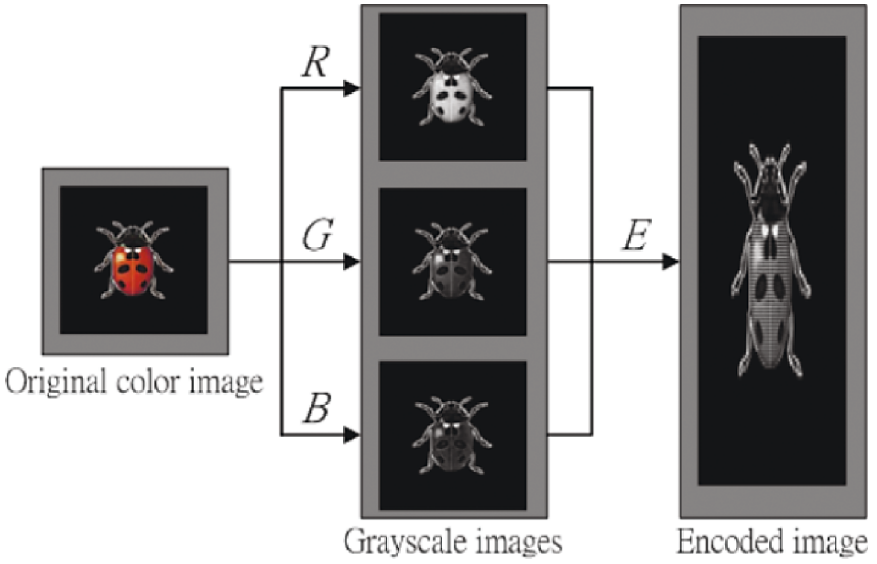


Fig. 35.2 The image encoding process

where r, g, and b represent the matrices of red, green, and blue images, respectively; n denotes the number of the maximum row vector.

2. Insert one-by-one respectively each r, g, and b row vector into a matrix.

$$e = \begin{bmatrix} r_1 \\ b_1 \\ g_1 \\ \vdots \\ r_n \\ b_n \\ g_n \end{bmatrix}. \tag{35.2}$$

Hence, the new matrix e is an encoded image, which includes all color elements. This method is different from multichannel single output JTC. The encoded image can represent the combination of red, green, and blue grayscale images at the input plane. The system could be smaller, since the channel number of the input image has been reduced to a single channel.

We put the reference encoded image and the target encoded image into LCSLM1 and LCSLM2, respectively. A collimated coherent beam illuminates both the LCSLM1 and LCSLM2, then it passes through the Fourier lens. We can represent the light fields W_1 and W_2 including the reflected and transmitted components as follows:

$$\begin{aligned} W_1(u,v) = {} & \alpha_1 \cdot H(u,v) \cdot \exp\{j2\pi(au)\} \\ & + \beta_2 \cdot E(u,v) \cdot \exp\{-j2\pi(au)\} \end{aligned} \tag{35.3}$$

and

$$\begin{aligned} W_2(u,v) = {} & \alpha_2 \cdot E(u,v) \cdot \exp\{j2\pi(au)\} \\ & + \beta_1 \cdot H(u,v) \cdot \exp\{-j2\pi(au)\}, \end{aligned} \tag{35.4}$$

where α and β denote the reflection and transmission coefficients of the BS2, respectively; $H(u,v)$, $E(u,v)$ are the Fourier transforms of $h(x,y)$ and $e(x,y)$, respectively; u and v are mutually independent frequency domain variables. Both of the outputs of CCD1 and CCD2 are connected to an ES. The resultant output of the ES due to the square-law detection is:

$$\begin{aligned} I_s = {} & |W_2(u,v)|^2 - |W_1(u,v)|^2 \\ = {} & (|\alpha_1|^2 - |\beta_1|^2) \cdot \{E(u,v) \cdot \exp[-j2\pi(au)]\} \cdot \{E(u,v) \cdot \exp[-j2\pi(au)]\}^* \\ & + (|\beta_2|^2 - |\alpha_2|^2) \cdot \{H(u,v) \cdot \exp[j2\pi(au)]\} \cdot \{H(u,v) \cdot \exp[j2\pi(au)]\}^* \\ & + (\alpha_1\beta_2^* - \beta_1\alpha_2^*) \cdot \{E(u,v) \cdot \exp[-j2\pi(au)]\} \cdot \{H(u,v) \cdot \exp[j2\pi(au)]\}^* \\ & + (\beta_2\alpha_1^* - \alpha_2\beta_1^*) \cdot \{H(u,v) \cdot \exp[j2\pi(au)]\} \cdot \{E(u,v) \cdot \exp[-j2\pi(au)]\}^*. \end{aligned} \tag{35.5}$$

According to Stokes relations [6], let $\alpha_2 = -\alpha_1$, $|\alpha_1| = |\beta_1|$, and $|\alpha_2| = |\beta_2|$. We can obtain:

$$\begin{aligned} I_s = 2\,|\beta_2\alpha_1^* + \alpha_1\beta_1^*| \cdot |H(u,v)| \cdot |E(u,v)| \\ \cdot \cos\{2\pi[2au] + \phi + \phi_H(u,v) - \phi_E(u,v)\}, \end{aligned} \tag{35.6}$$

where ϕ is the phase of $\beta_2\alpha_1^* + \alpha_1\beta_1^*$; ϕ_H and ϕ_E are the phases of $H(u,v)$ and $E(u,v)$, respectively. Finally, we can obtain the Mach-Zehnder joint transform power spectrum (MZJTPS) without zero order. Then we put I_s into LCSLM3. Finally, we can obtain the intensity distribution without an autocorrelation term at the output correlation plane via the CCD3 detector, which is:

$$\begin{aligned} I_o(x,y) = |c(-x,-y)|^2 \otimes \delta(x+2a,y) \\ + |c(x,y)|^2 \otimes \delta(x-2a,y), \end{aligned} \tag{35.7}$$

where $c(x,y) = h(x,y) \circ e(x,y)$, which is the complex cross-correlation between h and e; $\circ$ and $\otimes$ denote the correlation and convolution operations, respectively.

Moreover, we can minimize the average correlation energy for all training images by using the Lagrange multipliers method and constrain the correlation peak to be a desired value P to achieve optimal recognition ability. The optimum solution can be depicted as follows:

$$H_c = D^{-1}E\left[(E^*)^T D^{-1} E\right]^{-1} P^*, \tag{35.8}$$

where the superscript T denotes the transpose and the asterisk * represents the complex conjugate operation; D is a real-valued $(z \times z)$ diagonal matrix whose diagonal entry is given by calculating $\sum_{k=1}^{N} |E_k(u,v)|^2 \Big/ N$; E is a $z \times N$ data matrix whose kth column denotes the raster-scanned Fourier transform coefficients of the kth training image; P is the correlation peak requirement vector of size N, which can be determined by the user as the same constant to yield equal correlation peaks in response to all training images. The solution in Eqn. (35.8) is a vector representation in the frequency domain. The column vector H_c is rearranged as a square matrix $H(u,v)$. Finally, $h(x,y)$ can be obtained by an inverse Fourier transform of $H(u,v)$; that is

$$h(x,y) = \Im^{-1}\{H(u,v)\}. \tag{35.9}$$

In order to estimate the recognition ability, we use some measurement criteria to evaluate the performance. The CPI is the correlation peak intensity at the correlation output plane. This value is the fundamental criterion for other definitions of the performance evaluation. The CPI is defined as

$$\mathrm{CPI} = |c(0,0)|^2. \tag{35.10}$$

To inspect the detection ability, one of the important aspects must be the accuracy of the detection, which can be determined by the sharpness of the correlation profiles in optical pattern recognition. The peak response is defined to be the intensity of the desired peak occurring in the expected position $(0,0)$ of the output plane. The

sharpness of the correlation peak can be measured by the peak to correlation energy (PCE) ratio at the desired area, which is defined as:

$$\mathrm{PCE} = \frac{|c(0,0)|^2}{\sum_{x,y}|c(x,y)|^2}. \tag{35.11}$$

The PSR, which is defined as the ratio between the primary correlation peak energy and the secondary peak energy, is another important measurement criterion of pattern recognition, which is expressed as:

$$\mathrm{PSR} = \frac{|c(0,0)|^2}{\max\limits_{x,y\in\Theta}\left\{|c(x,y)|^2\right\}}, \tag{35.12}$$

where $\Theta = \{(x,y)\,|\,|x|>2, |y|>2\}$. The secondary sidelobe is defined as the highest intensity point in the correlation plane at a point at least two pixels away from the CPI. In the definitions of the PCE and PSR, we have assumed that the target image does not shift. Hence, the primary peak is located at the point (0, 0). A higher PSR value implies higher pattern detection. Accordingly, high values of PCE and PSR at output show good performance of our system. In reality, there might be some random noise in real images that leads to reduction in recognition ability. The input signal power to noise power ratio (SNR) is used to quantify the amount of noise added to the target image, as expressed by:

$$\mathrm{SNR} = \frac{\sum_{x,y}|e_n(x,y)|^2}{B\cdot\sigma^2}, \tag{35.13}$$

where σ^2 is the variance of each pixel and B is the pixel number. Adding noise to the target image creates noisy patterns, which will be utilized as the test images. Moreover, we can evaluate system recognition ability in a noisy environment by metrics such as the average PCE (APCE) and average PSR (APSR) in the region of interest (ROI). Each value is the mean of the CPI over 100 random noise tests. The definitions of the APCE and APSR are stated as the following equations:

$$\mathrm{APCE} = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{|c_i(0,0)|^2}{\sum\limits_{x,y}|c_i(x,y)|^2}\right\}, \tag{35.14}$$

and

$$\mathrm{APSR} = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{|c_i(0,0)|^2}{\max\limits_{x,y\in\Omega}\left\{|c_i(x,y)|^2\right\}}\right\}. \tag{35.15}$$

35.3 Simulation Results

The performance of the MZJTC with image encoding technique is investigated. We chose the colorful ladybug image with 128 × 128 pixels as the target image. Furthermore, we utilized the combination of red, green, and blue grayscale images to form an encoded image with 384 × 128 pixels. The original training images set contains 15 color images with different angles of in-plane rotation, and the rotational distortion range from -14° to 14° is considered (in steps of 2°), as shown in Fig. 35.3(a).

Figure 35.3(b) illustrates some examples of the training set encoded images for *r*, *g*, and *b* channels. The synthetic reference function was produced for each encoded image.

Therefore, we synthesize the reference image from a training encoded images set with the MACE approach. In order to test the validity of the proposed configuration, we use the Matlab software package to simulate the execution of the proposed system. Compared with this monochromatic counterpart, polychromatic JTC has a large input image plane. However, the encoded image is composed of red, green, and blue grayscale images. For this reason, the size of the input scene could be smaller than the initial polychromatic JTC. The training data was fast Fourier transformed and the synthetic reference function has been designed. The input scenes involve two parts. The LCSLM1 shows the synthetic encoded image of the reference image, while the LCSLM2 contains the target encoded image of the input test scene. The object under test is the noiseless, 0° rotated training encoded image. The sizes of the LCSLMs are chosen to be 768 × 768 pixels in the input planes. The input planes for

Fig. 35.3 (a) The standard training images for original test image with rotation angles from -14° to 14° (in steps of 2°)

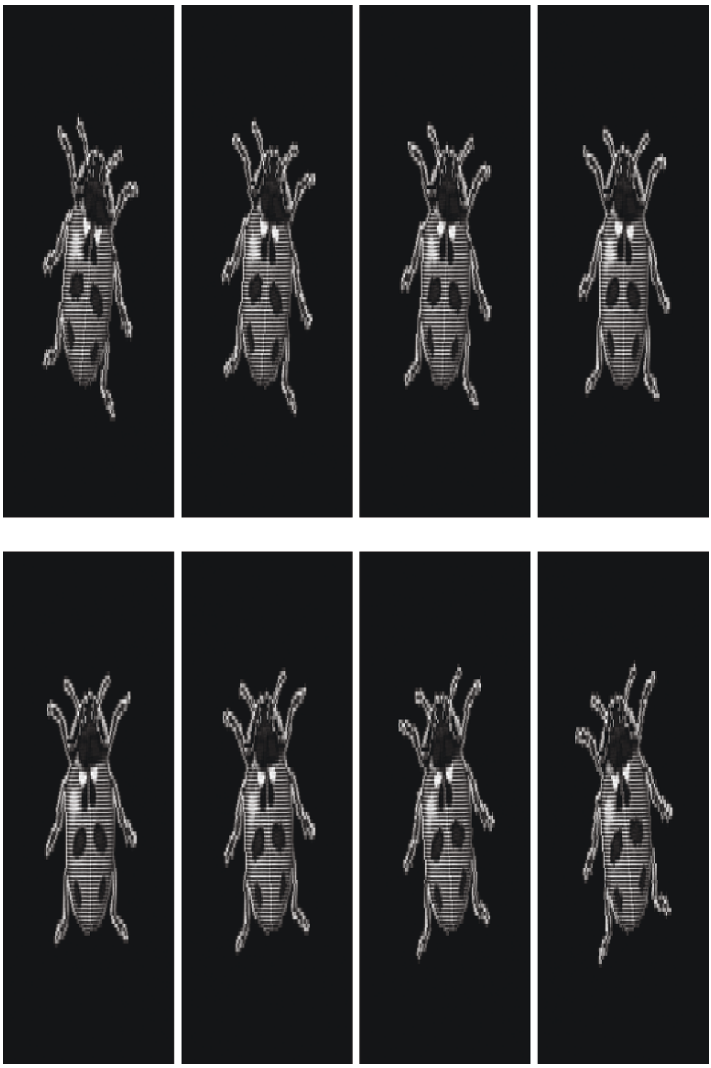


Fig. 35.3 (b) Eight color encoded images organized with *r*, *g*, and *b* components from the training set, shown in grayscale

the MZJTC with image encoding technique are shown in Figs. 35.4(a) and 35.4(b), respectively. From the previous analysis, the ROI at the output plane of MZJTC are the desired areas around $(2a,0)$ and $(-2a,0)$.

Finally, the corresponding 3D output profile is obtained by this MZJTC, as shown in Fig. 35.4(c). In the 3D output profile, the desired sharp correlation peaks can be seen clearly. Therefore, the MZJTC system with image encoding technique yields acceptable results to detect targets.

Fig. 35.4 (a) Synthetic reference image in the spatial domain of LCSLM1

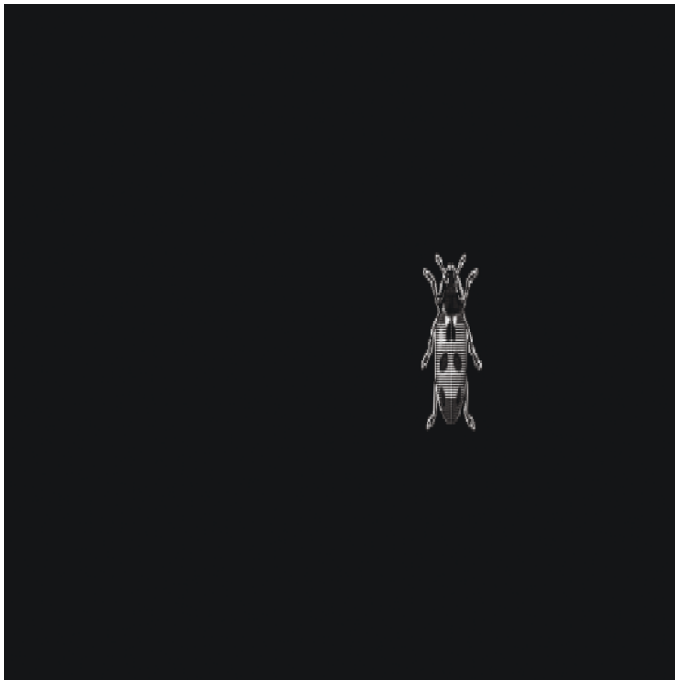


Fig. 35.4 (b) The input plane of LCSLM2

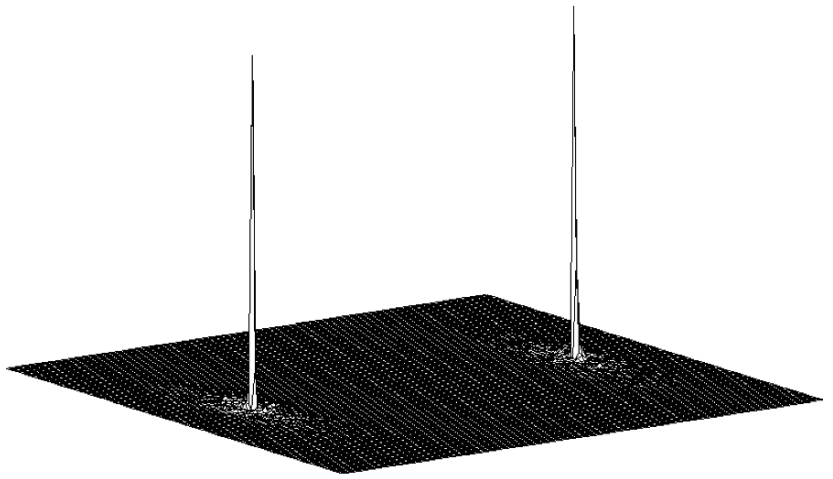


Fig. 35.4 (c) The corresponding three-dimensional output profile

In order to test our system for multiple targets detection, an original colorful ladybug image is used as the non-target image. Its corresponding encoded images are shown in Fig. 35.5(a). The input test scene of the LCSLM2 for multiple targets is shown in Fig. 35.5(b). We selected the 0° and 14° rotated images to be the target images, and put a non-target encoded image in the middle of LCSLM2. The corresponding 3D correlation intensity profile is shown in Fig. 35.5(c). From the 3D output profile, the desired sharp correlation peaks can be seen clearly. Therefore, the system yields acceptable results for multiple targets detection.

In addition, we investigated the performance of each training encoded image. The training encoded images are the angles from -14° to 14° in steps of 2°. The CPI, PCE, and PSR versus the rotation angle for each training image are illustrated in Figs. 35.6(a), 35.6(b), and 35.6(c), respectively.

In Figs. 35.6(a) and 35.6(b), the curves of the CPI and PCE versus different angles of rotation are smooth and steady for each training image. Notice that the curves of PSR for the MZJTC with image encoding method are oscillating and higher than a certain level in Fig. 35.6(c). From Figs. 35.6(a–c), we can observe that our system has quite a good performance for each encoded image.

In addition, we have also addressed the problem of noise in this system. Consequently, some random noise is added to the test encoded image. The degree of noise addition is set according to Eqn. (35.13). Figure 35.7 depicts some of the encoded patterns with different SNR values from -5 dB to 10 dB in steps of 5dB.

The test encoded target pattern, which is the unrotated target image, had Gaussian random noise added. The entire process was repeated over 100 times at each SNR value. The values of ACPI, APCE, and APSR as the function of the input SNR are plotted in Figs. 35.8(a), 35.8(b), and 35.8(c), respectively. The range of the SNR was chosen from -5 to 10 dB. The ACPI values are quite close to each other in the

given SNR range, as illustrated in Fig. 35.8(a). Moreover, we can also observe that the variation of the ACPI is slightly higher than 80 percent over 0 dB. The trends for the curves of APCE and APSR versus input SNR are also exactly what we expect to see. The APCE and APSR values are rising as the input SNR increases. Therefore, the ability of the proposed system to counter noise is quite good.

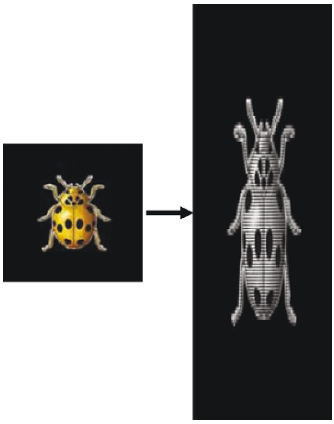


Fig. 35.5 (a) The multiple targets data for the non-target image

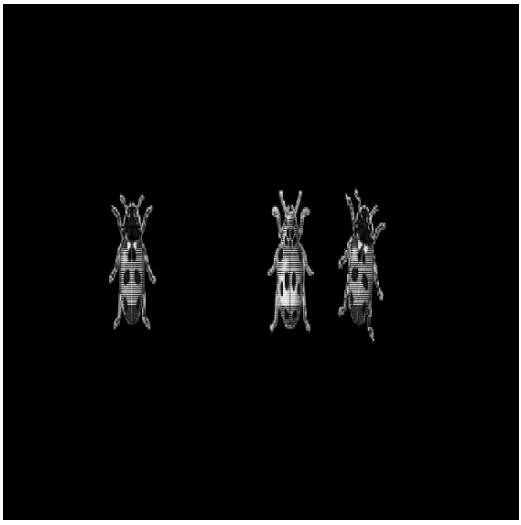


Fig. 35.5 (b) The input plane of LCSLM2 with multiple target encoded images

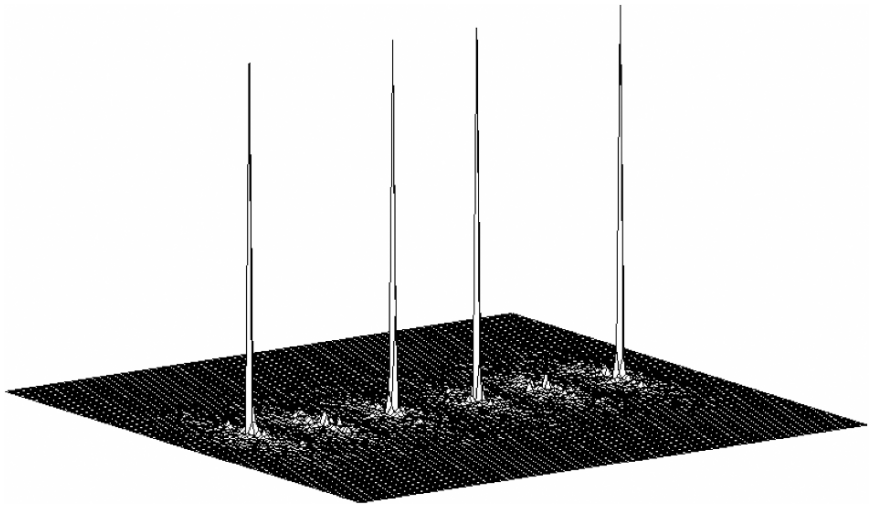


Fig. 35.5 (c) The corresponding 3D output profile for multiple targets detection

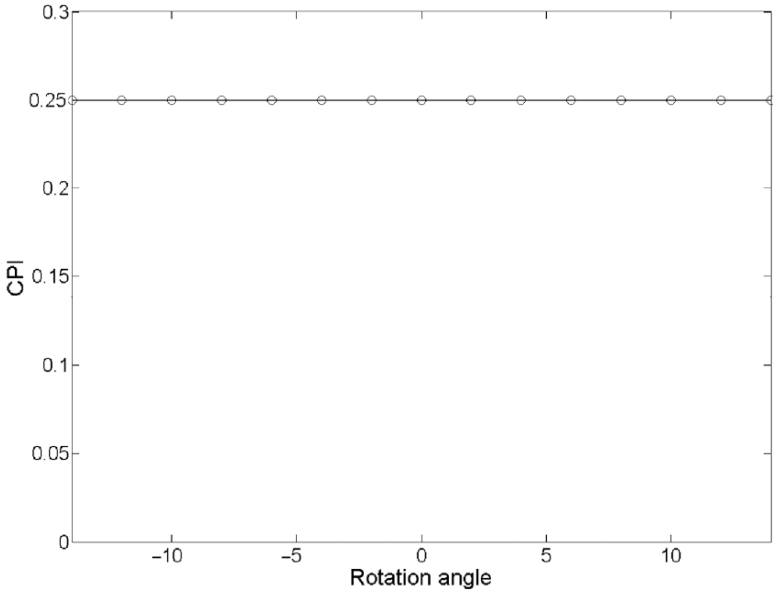


Fig. 35.6 (a) CPI versus the rotation angle

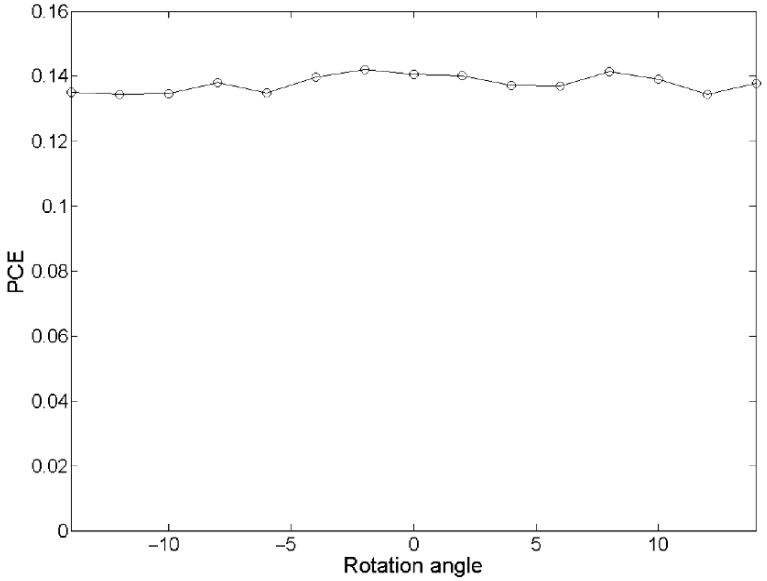


Fig. 35.6 (b) PCE versus the rotation angle

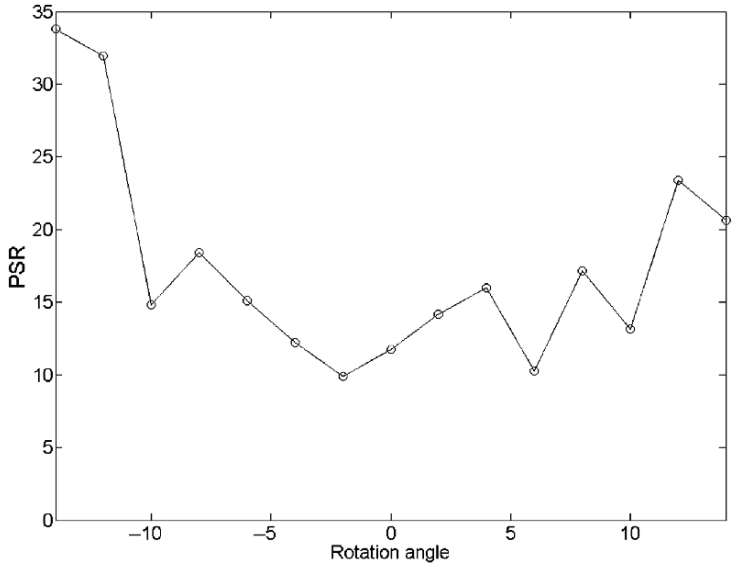


Fig. 35.6 (c) PSR versus the rotation angle

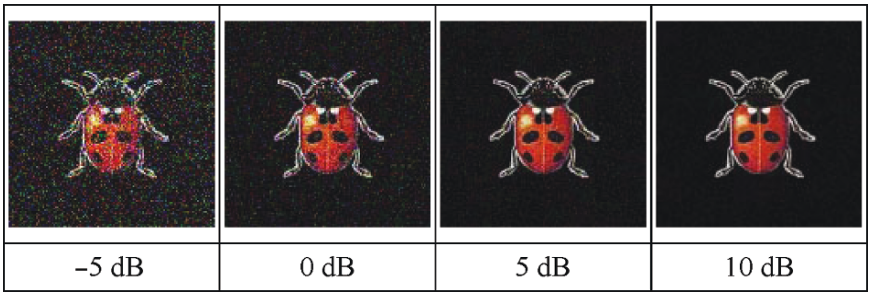


Fig. 35.7 Encoded patterns with random noise from −5 dB to 10 dB

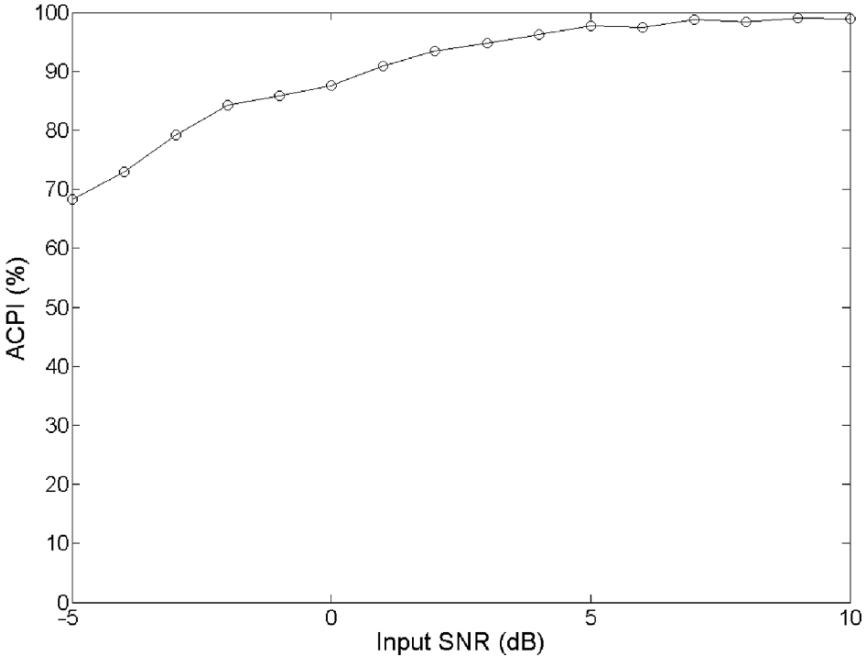


Fig. 35.8 (a) ACPI versus input SNR from −5 to 10 dB

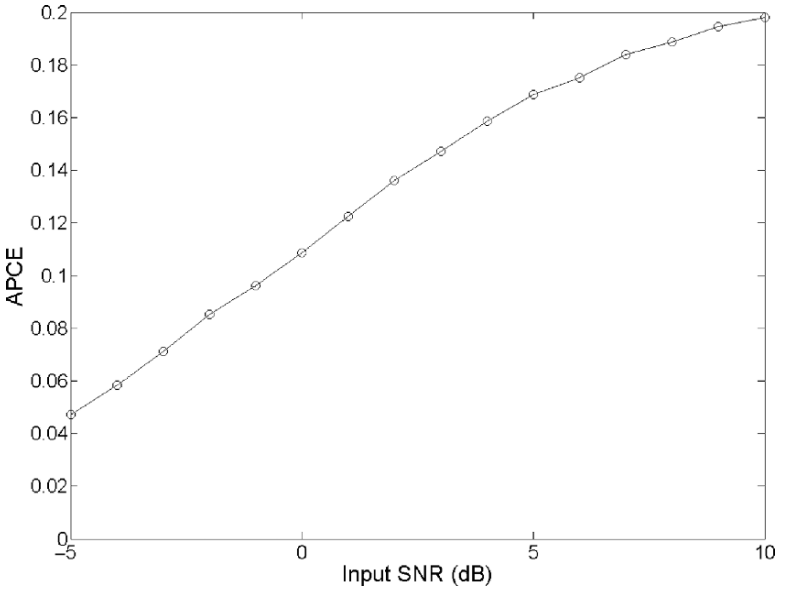


Fig. 35.8 (b) APCE versus input SNR from −5 to 10 dB

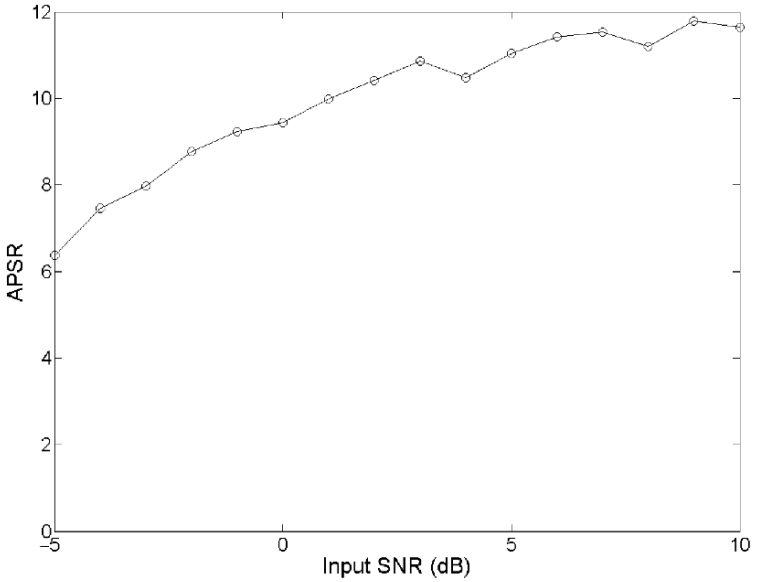


Fig. 35.8 (c) APSR versus input SNR from −5 to 10 dB

35.4 Conclusions

In this article, we have presented a novel optoelectronic system with image encoding technique for invariant color pattern recognition based on the Mach-Zehnder joint transform correlation. To better utilize the large space-bandwidth product of the JTC input plane and to use the parallel nature of optics, many researchers have proposed a JTC setup capable of performing elementary correlations simultaneously. Compared to multichannel JTC in the joint input plane, the proposed multichannel single-output MZJTC of image encoding can be easier and the system can be smaller, since the input image can be reduced to a single channel. In addition, our MZJTC system can remove the zero order term in one step. For real-time processing, the joint input plane is displayed on the LCSLMs. In the tests, the algorithm is robust enough to realize the real-time process and distortion invariant. Furthermore, numerical results confirm the effectiveness of the proposed technique for noisy images. It is expected that the proposed method will be promising in practical applications. We have proposed a novel system, which offers an attractive alternative for color pattern recognition. To summarize, the overall performance on optical color pattern recognition ability is good in our system.

References

1. Chen C, Chen C, Lee C, Chen C (2004) Constrained optimization on the nonzero order joint transform correlator with the Mach-Zehnder configuration. Opt Commun 231:165–173
2. Cheng C, Tu H (2002) Implementation of a non-zero order joint transform correlator using interferometric technique. Opt Rev 9:193–196
3. Chen C, Fang J (2000) Cross-correlation peak optimization on joint transform correlators Opt Commun 178:315–322
4. Chen C, Fang J (2001) Chinese seal imprint recognition with joint transform correlators. Opt Eng 40:2159–2164
5. Deutsch M, Garcia J, Mendlovic D (1996) Multi-channel single-output color pattern recognition by use of a joint-transform correlator. Appl Opt 35:6976–6982
6. Hecht E (2002) Optics. 4th edition, Reading, Mass., Addison-Wesley, pp 136–137
7. Javidi B, Kuo CJ (1988) Joint transform image correlation using a binary spatial light modulator at the Fourier plane. Appl Opt 27:663–665
8. Li C, Yin S, Yu FTS (1998) Non-zero order joint transform correlator. Opt Eng 37:58–65
9. Lu G, Zhang Z, Wu S, Yu FTS (1997) Implementation of a non-zero order joint transform correlator by use of phase shifting techniques. Appl Opt 36:470–483
10. Mahalanobis A, Kumar BVKV, Casasent D (1987) Minimum average correlation energy filters. App Opt 26:3633–3640
11. VanderLugt A (1964) Signal detection by complex spatial filtering. IEEE Trans Inf Th IT-10:139–146
12. Weaver CS, Goodman JW (1966) A technique for optically convolving two functions. Appl Opt 5:1248–1249
13. Wu C, Chen C, Fang J (2002) Constrainted optimization for color pattern recognition with a non-zero order joint transform correlator. Microwave Opt Tech Lett 33:385–388

14. Wu C, Chen C (2003) Performance comparison between multichannel polychromatic and conventional joint transform correlators. Opt Eng 42:1758–1765
15. Yu FTS, Lu G, Lu M, Zhao D (1995) Application of position encoding to complex joint transform correlator. Appl Opt 34:1386–1388

Chapter 36
Approach to the Chinese Seal Registration

Yung-Sheng Chen

Registration becomes difficult when the concerned pattern—e.g., a Chinese seal pattern—is noisy and rotated. An approach to Chinese seal registration is presented. For a color image, a self-organization feature map algorithm and a simple color filtering process are applied for seal segmentation. Then, a contouring analysis is performed on the segmented seal to detect its principal orientation. Finally, the registration is achieved by combining the orientation difference and the center region translation for two seals. Experiments on seals with rotations, heavy noise, and different resolutions confirm the feasibility of the proposed approach.

36.1 Introduction

In Chinese society, seals are usually used instead of signatures [1], which are adopted for proving identity in western countries on many types of articles—e.g., paintings, proposals, money withdrawing lists, checks, receipts, etc. Figure 36.1(a) shows a partial scanned color image containing a RED-type seal from a Chinese check, for example.

In the past, many pattern recognition techniques have been proposed for the seal identification—e.g., use of fluency function approximation and relaxation matching method [2] and implementation of "hardmatching" principle [3], etc. It has been shown that an effective identification is strongly related to the quality of seal imprint that may result from paper, setting pressure, slant setting, and paint. Thus it is a reasonable requirement that the seal is set on paper with enough care so that the "geometrical property" of the resultant seal imprint is preserved enough for identification [3]. Based on the hardmatching principle on the characteristics of seal

Yung-Sheng Chen
Department of Electrical Engineering, Yuan Ze University, Chung-Li, Taoyuan 320, Taiwan, ROC
e-mail: eeyschen@saturn.yzu.edu.tw

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

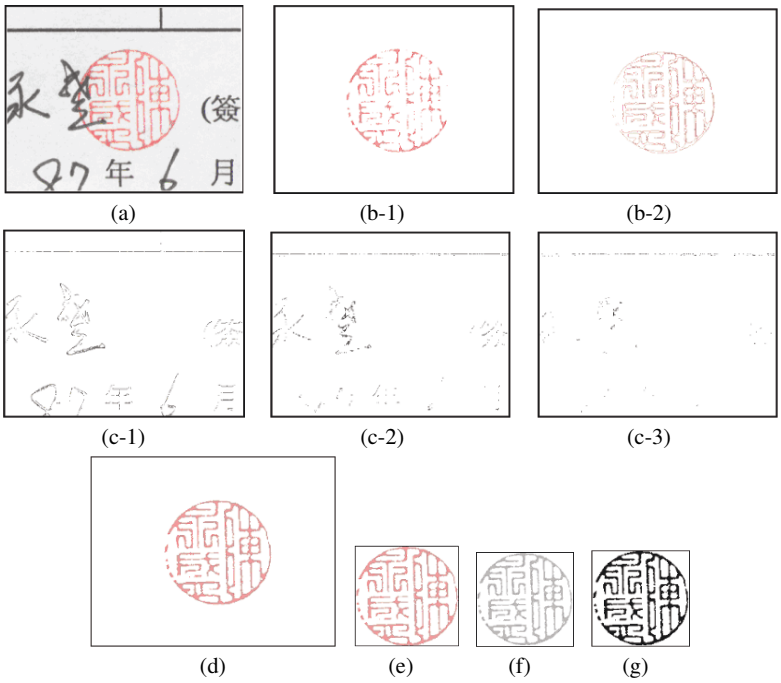
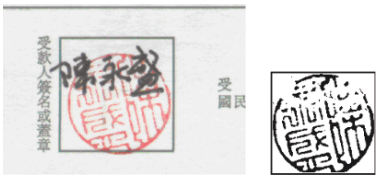


Fig. 36.1 (a) shows a scanned color image containing a seal (300 dpi). (b-1)–(b-2) and (c-1)–(c-3) show two clusters for the color segmentation. (d) is the desired RED-type seal image. (e) and (f) show the final seal image and its gray image, respectively. (g) is the binary image

identification, any preprocessing for the seal image, such as the feature vectorization that will affect the original geometrical property is not recommended.

The hardmatching usually performed by human visual inspection may be stated as follows [3]: Let R and I be two seal imprints as reference 1 and input 1, respectively. R is preregistered as a reference in a personal dictionary of a bank. If I appearing in a check as shown in Fig. 36.1(a) is given, the teller of the bank will take the original R and compare them by focusing their centroids, rotating I, and looking for the corresponding details between them. If the details between them are satisfactory to the agreement of the teller's subjective judgment, then the input I is accepted. According to the human visual inspection, the hardmatching can be divided into the following steps—i.e., segmenting or positioning the seal imprint, finding its centroid, rotating the seal, as well as matching in detail and picking out the maximum match for the final decision. In these suggested processing steps, three main problems may occur. First, if the seal part is interfered with by other information such as a signature or background colors, the segmentation will become difficult. Secondly, the decided centroid of a seal will be unstable due to the noise involved in a seal information such as Fig. 36.2 shows. If the centroid of a seal cannot be found correctly, the original approach will fail. Thirdly, the automatic seal identification is usually not preferred in practical applications due to the security consideration. The



(a) (b)

Fig. 36.2 (a) A scanned seal image (300 dpi) for testing, and (b) its binarized image of the same type of seal as shown in Fig. 36.1(a)

visualized matching result providing a fast visual inspection for a bank's teller will be appreciated instead. Accordingly, the goal of this paper is to present an effective registration approach to overcome these problems.

Before presenting our approach for seal registration, some requirements and corresponding operations are summarized as follows:

16. A seal is composed of a set of RED-type colors (i.e., RED is stronger than GREEN and BLUE). This requirement is practically used in Chinese society for the purpose of person identification. In addition, it will make it easy for seal segmentation.
- To distinguish the seal from the background, it is reasonable to have the condition that background colors are NOT of seal's colors.
- The final segmented seal is converted and represented by a BINARY image, which will facilitate the registration of two seals.
- Because it is difficult to define the matching score under the security concern, the bank's teller is usually preferred to check the details between input and reference seals. Thus, a very important process of the proposed approach is to display the registration result by overlapping two seals for further inspection.

36.2 Proposed Approach

According to the foregoing statements, an approach is presented in this section to segment the seal image and perform the seal registration effectively. It is mainly composed of the following parts: segmentation of a seal image, detection of principal orientation of the seal, and registration.

36.2.1 Segmentation

A partial check image containing a Chinese seal as shown in Fig. 36.1(a), where the resolution is 300 dpi, can be readily scanned via a color scanner. The segmentation of seal image is the first step in our approach and performed by a (self-organization

feature map) (SOM) algorithm, spatial distance computation, range confirmation, and binarization. The details of the first two algorithms can be found in Chen and Hsu (1994, 1995), which have been confirmed as an effective scheme for the color image segmentation and may be briefly reviewed as follows.

36.2.1.1 Presegmentation

The image obtained by a color scanner is represented as a two-dimensional (2D) color intensity function **f**; where each pixel is composed of red, green, and blue components, and can be denoted by a three-element color vector. The seal segmentation is a process of pixel classifications—the image is segmented into subsets (called color planes here) by assigning the individual color vectors to classes, where each color plane having the same image size contains one set of pixels all having approximately the same color vector, and the other set of pixels all having the zero vector. Therefore, if the K color vectors of $\mathbf{f}$ have been found, then the original $\mathbf{f}$ can be approximately represented by the linear combination of $\mathbf{f}_k$, $k = 1, 2, ..., K$—i.e.,

$$\mathbf{f} \approx \mathbf{f}_1 + \mathbf{f}_2 + \cdots + \mathbf{f}_K = \sum_{k=1}^{K} \mathbf{f}_k. \tag{36.1}$$

To perform the pre segmentation, a one-dimensional (1D) SOM algorithm is used with similar implementation in Chen and Hsu (1994, 1995). Our SOM algorithm has also been successfully applied to segment a natural color image for the investigation of human visual perception [6].

36.2.1.2 Spatial Distance

Because the seal part in the image may be composed of similar color planes, the spatial distance between any two color planes is used to measure their nearness. The closer the two color planes are, the higher the possibility of both belonging to the same class is. According to the information of distances, the segmentation of CBP is therefore reduced to classify color planes into the seal part and non-seal parts, and Eqn. 36.1 can be rewritten as

$$\mathbf{f} \approx \mathbf{f}_{seal} + \mathbf{f}_{non-seal}. \tag{36.2}$$

The $\mathbf{f}_{seal}$ and $\mathbf{f}_{non-seal}$ may be respectively expressed by

$$\mathbf{f}_{seal} = \sum_{k \in seal} \mathbf{f}_k, \tag{36.3}$$

and

$$\mathbf{f}_{non-seal} = \sum_{k \in non-seal} \mathbf{f}_k. \tag{36.4}$$

The nearness measurement can be achieved by computing the spatial distance d_{hk} between any two color planes, $\mathbf{f}_h$ and $\mathbf{f}_k$ ($h \neq k$). The distance d_{hk} is defined as follows.

Let the distance between two non zero pixels, $\mathbf{f}_h(i,j)$ and $\mathbf{f}_k(u,v)$, be Euclidean as

$$d(i,j;u,v) = \sqrt{(i-u)^2 + (j-v)^2} \tag{36.5}$$

and the distance between the non zero pixel $\mathbf{f}_h(i,j)$ and the color plane $\mathbf{f}_k$ is defined as

$$d(i,j;\mathbf{f}_k) = \min_{\forall \mathbf{f}_k(u,v) \neq \mathbf{0}} d(i,j;u,v) \tag{36.6}$$

then the distance between color planes $\mathbf{f}_h$ and $\mathbf{f}_k$ may be expressed as

$$d_{hk} = \frac{1}{n(\mathbf{f}_h)} \sum_{\forall \mathbf{f}_h(i,j) \neq \mathbf{0}} d(i,j;\mathbf{f}_k) \tag{36.7}$$

Here, $n(\mathbf{f}_h)$ denotes the number of non zero pixels belonging to the color plane $\mathbf{f}_h$. It performs a normalizing factor. By the computation d_{hk}, we can measure the nearness from $\mathbf{f}_h$ to $\mathbf{f}_k$, and vice versa. Because $\mathbf{f}_h$ and $\mathbf{f}_k$ are different, $n(\mathbf{f}_h)$ and $n(\mathbf{f}_k)$ are also different. This implies that d_{hk} will not be equal to d_{kh} in usual cases. Hence, a more appropriate measure is computed by the average $\bar{d}_{hk} = (d_{hk} + d_{kh})/2$, and $\bar{d}_{hk} = \bar{d}_{kh}$. This spatial distance computation has also been effectively applied to solve the correspondence problem on the three-dimensional (3D) surface measurement [7].

After performing the algorithms, we can obtain the cluster containing RED seal information as shown in Fig. 36.1(b-1) and 36.1(b-2), whereas the other cluster containing the partial signature information as shown in Fig. 36.1(c-1)–36.1(c-3). Figure 36.1(d) shows the combined RED seal image using the information in Fig. 36.1(b-1) and 36.1(b-2).

36.2.1.3 Range Confirmation

Because only the seal information is needed, we use the following procedure to re-find the range of a single seal from all clusters.

1. Find the possible range of the single seal from all the clusters.
2. Fill the horizontal and vertical line-gaps $\leq$3 pixels.
3. Remove the clusters having information of null, large background, and noise.
4. Re-find the range of the single seal from the useful clusters.

Based on the re-found range, the final single seal image and its gray image are obtained as shown in Fig. 36.1(e) and 36.1(f), respectively.

36.2.1.4 Binarization

To facilitate the further process of registration, the gray seal image should be converted into a binary image. Because the distribution of gray information (related to

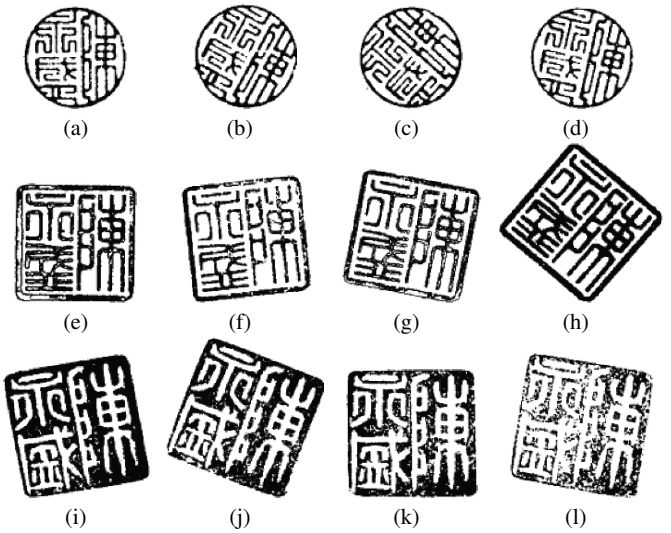


Fig. 36.3 Three types of binarized seals of the author's, own are used for experiments. (a)–(d), (e)–(h), and (i)–(l) are represented for Types 1, 2, and 3, respectively. Note here that heavy noise may appear in the seals of Type 2 and 3

the original RED information) for a seal image is not well-shaped. We need an adaptive threshold, TH as expressed below, to perform the binarization. Let *mean* and *median* of the gray distribution of the seal image be pre-computed. The TH may be obtained by

$$TH = 2T_{\min} - T_{\max} \tag{36.8}$$

where $T_{\min} = \min(mean, median)$ and $T_{\max} = \max(mean, median)$, respectively. For the current example, *mean* = 237 and *median* = 255, thus TH = 219 and we can obtain the binary seal image as shown in Fig. 36.1(g). Figure 36.2 gives the other result of the same type of seal image, which shows that some seal information disappears. Figure 36.3 gives three types of binarized seals of the author's own, where the circular-male engraved seals are regarded as Type 1, the rectangular-male engraved seals are regarded as Type 2, and the rectangular-female engraved seals are regarded as Type 3, respectively. Note that heavy noise may appear in the seals of Type 2 and 3.

36.2.2 Detection of Principal Orientation

Because the binary seal image often contains noise due to its inherent property, the detection of its relative orientation corresponding to the original one will be not easy. Many techniques have been proposed to perform such a task, such as the moment-based method [8] and Fourier descriptor [9]. However, the moment invariants are

only in reasonably close agreement to geometrically modified versions of the same object. It is not suitable to be applied for the current case of heavy noise. Similarly, the Fourier descriptor is usually applied to shape analysis possessing a continuous closed curve. To develop a feasible but simple algorithm for dealing with this problem, the following procedure based on contouring analysis is presented to determine the so-called principal orientation (PO) of a seal.

1. Input a binary seal image.
2. Perform median process on the binary seal image.
3. Find the contour of the seal image.
4. Calculate angle information, ranging from 0° to 179°, for each contour pixel in the selected contours containing enough contour pixels.
5. Produce histogram for the angle information and normalize it with the maximum pick.
6. Produce cumulative distribution function (CDF) for all angle information between two picks, located at the adjacent left side and right side of the found maximum pick. The angle information in the range is dominated by these two picks. Take the angle indicating 50 percent CDF as the principal orientation of the seal.

The key steps 4-6 in the above procedure are detailed as follows. Our binary seal image can be described by its contour information, and represented by a finite set of contour chains, C_i, $i = 1,2,\cdots,m$. The kth contour pixel of the ith contour chain is denoted by $C_i(k)$, whose coordinate is denoted by $(x_{C_i(k)}, y_{C_i(k)})$. The length of a contour chain can be obtained by counting the number of pixels along the contour. According to the geometrical property of seal processing mentioned previously, the significant seal information should be preserved for identification [3]. This implies that a contour chain having longer length possesses more significant seal information. Hence, we define a threshold (=50 was used in our experiments) to filter out the contour chains whose length is smaller than the threshold. The set of contour chains will thus be reduced to C_i, $i = 1,2,\cdots,n \leq m$. Based on these definitions, we let $TA_i(k)$ denote the tangent angle for the kth contour pixel in the ith contour chain, which may be obtained and expressed by

$$TA_i(k) = \tan^{-1}\left(\frac{y_{C_i(k+\Delta)} - y_{C_i(k-\Delta)}}{x_{C_i(k+\Delta)} - x_{C_i(k-\Delta)}}\right) \tag{36.9}$$

where $C_i(k+\Delta)$ and $C_i(k-\Delta)$ denotes the next Δth and the previous Δth contour pixel, respectively, based on the current contour pixels. To produce the angle histogram, and for further detection of principal orientation, all the TAs are transformed into integer degrees ranging from 0° to 179°. Figure 36.4(a) and 36.4(b) show two examples for the contour information and the angle histogram, where the color information is used to label the corresponding TA. The top point, namely (θ, h), of each bar in the angle histogram denotes the existance h contour pixels having $TA = \theta$. Note that each angle histogram has been normalized such that the maximum peak is $h_p = 255$.

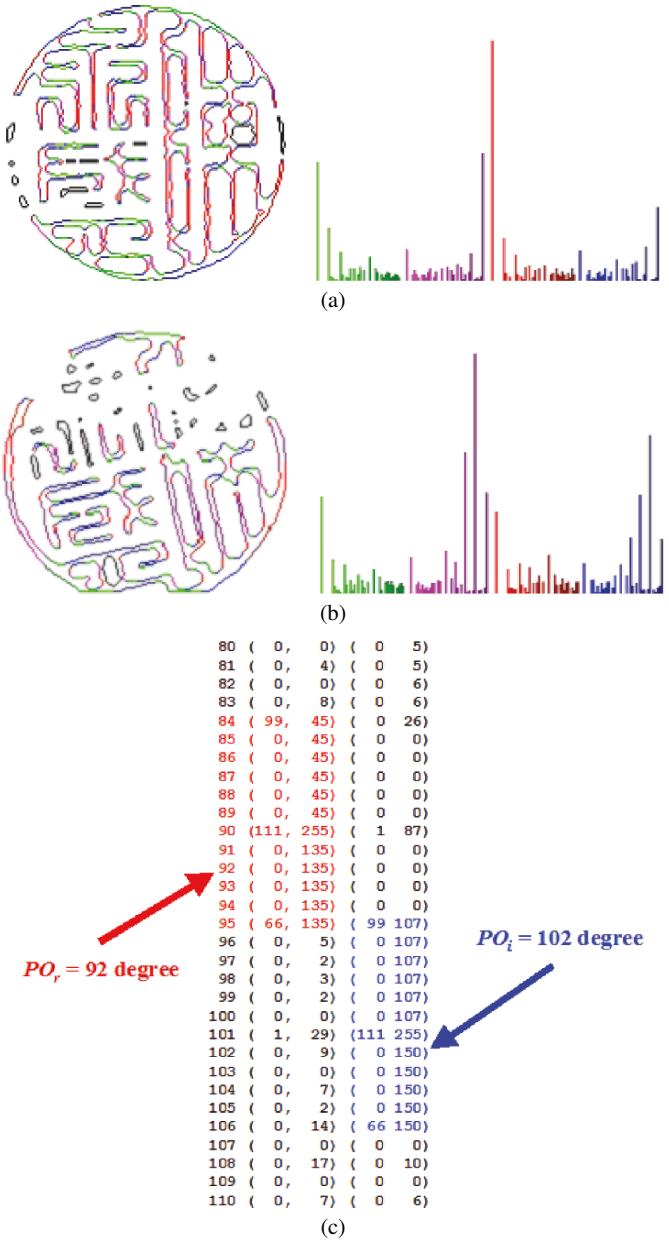


Fig. 36.4 (a) Contour and angle information of the seal in Fig. 36.1(g), and (b) those in Fig. 36.2(b). Their partial CDFs and the determined POs are shown in (c)

From the angle histogram of Fig. 36.4(a) and 36.4(b), a peak is apparently observed due to the stroke property for a seal. However, the principal orientation (PO) of a seal image may be affected and changed within a local range due to the effect of

"seal noise." Hence, to determine a more reasonable PO for a seal image, the angle information in a local angle range $[\theta_l, \theta_r]$ is considered. Here, θ_l and θ_r denote the second maximum peak at the left side (toward to 179°) and right side (toward to 0°), respectively, neighboring to the maximum peak θ_p. To filter out the small values, we take the mean value $h_{th} = (\sum_{\forall\theta} h(\theta))/180$ as a threshold. The candidate of second maximum peak should be larger than h_{th}.

According to the property of angle histogram for a seal image, PO $\in [\theta_l, \theta_r]$, and to make the PO information more robust, we modify the angle histogram in the range $[\theta_l, \theta_r]$ as below

$$\widetilde{h}(\theta) = \begin{cases} h(\theta_l), & \theta_l \geq \theta > \theta_p, \\ h(\theta_p), & \theta = \theta_p, \\ h(\theta_r), & \theta_p > \theta \geq \theta_r, \end{cases} \tag{36.10}$$

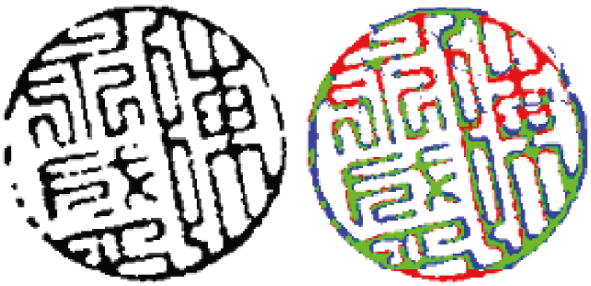
and construct the cumulative distribution function (*CDF*) for the new angle histogram in the range $[\theta_l, \theta_r]$. The PO of a seal is thus defined by the θ having the 50th percentiles of the CDF, that is,

$$PO = \left\{ \theta_{50\%} \middle| CDF(\theta_{50\%}) = \left(\sum_{\forall\theta\in[\theta_l,\theta_r]} \widetilde{h}(\theta) \right)/2 \right\}. \tag{36.11}$$

Now let the seal in Fig. 36.1(g) be reference 1 denoted as BS_r, and that in Fig. 36.2(b) be input 1 denoted as BS_i. The corresponding contour and angle information, as well as partial CDFs are shown in Fig. 36.4. From the results, we obtain $PO_r = 92°$ and $PO_i = 102°$, for reference and input seals, respectively. Thus the difference rotation angle from reference 1 to input 1 can be obtained by

$$\Delta PO = PO_r - PO_i. \tag{36.12}$$

For example, $\Delta PO = -10°$ in this case states that it is possible to have a match with input 1 when reference 1 is rotated $-10°$ in counter-clockwise as Fig. 36.5(a) shows. The rotated version of BS_r is denoted as RBS_r.



(a) (b)

Fig. 36.5 (a) shows the new seal RBS_r of Fig. 36.1(g), and (b) shows the final registration result

36.2.3 Registration

To complete the registration task, the translation vector $\vec{V}$ from reference 1 to input 1 should be determined. Because the range containing a seal has been found in the segmentation stage such as Fig. 36.1(g) and Fig. 36.2(b) display, it is possible to find a centered window in RBS_r that is possibly matched to the specified centered range in BS_i using template matching.

Let W and H be the width and height of RBS_r, while $W/2$ and $H/2$ are used for the centered window so that an enough seal information may facilitate for template matching. In addition, let $(x_r^{center}, y_r^{center})$ be the center point of RBS_r, and the found best-match window in BS_i will have its center point at $(x_i^{center}, y_i^{center})$. Thus, the translation vector $\vec{V}$ defined from $(x_r^{center}, y_r^{center})$ to $(x_i^{center}, y_i^{center})$ can be obtained. Based on $\vec{V}$ and the previously detected ΔPO, our registration procedure may be summarized as follows and illustrated in Fig. 36.6.

1. Give BS_r and BS_i.
2. Detect PO_r of BS_r and PO_i of BS_i, respectively.
3. Rotate BS_r by ΔPO degrees to form a new seal RBS_r.
4. Find the translation vector $\vec{V}$.
5. Translate RBS_r based on $\vec{V}$ to form a new seal $TRBS_r$.
6. Overlap and display the two seals, $TRBS_r$ and BS_i, as a registration result.

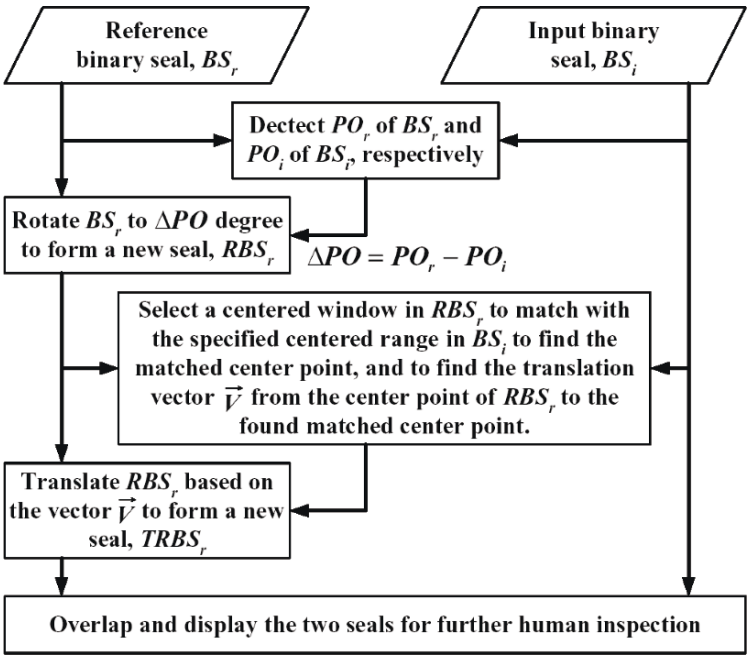


Fig. 36.6 Flowchart of our seal registration algorithm

Based on the found $\Delta PO = -10^\circ$ in the illustration of Fig. 36.4, RBS_r of Fig. 36.1(g) is obtained as shown in Fig. 36.5(a). After performing the registration procedure for the seals of Fig. 36.1(g) and Fig. 36.2(b), the registration result is shown in Fig. 36.5(b). Figure 36.7 gives the other results with a Type 3 seal to further demonstrate the feasibility of the proposed approach, where the seals in Fig. 36.3(i) and Fig. 36.3(l) are regarded as reference 1 and input 1, respectively. Some early investigation using contour analysis can be found in Chen (2003), in which a mean registration error with 1.13° was reported.

36.3 Results and Discussion

To further confirm the feasibility of the proposed approach, three types of binarized seals of the author's own shown in Fig. 36.3 are used for experimentation, where 36.3(a)–(d), 36.3(e)–(h), and 36.3(i)–(l) are represented for Types 1, 2, and 3 seal images, respectively. The resolution of these images are 300 dpi. Figure 36.8 presents the well-resultant registration performed by the proposed approach.

Because our method is strongly related to the contour information, it is worthy to investigate the effect of changing resolution. Intuitively, the lower the resolution is, the more rough the contour information should be. This can be discussed by applying our approach to the seals with different resolutions (100 dpi and 200 dpi are used for illustrations). Figure 36.9 shows the results using 100 dpi images including Type 1, 2, and 3 seals from top to bottom. We let (a) be a reference 1 and (b) the test 1, respectively. Figure 36.9(c) shows that the contour information of a low resolution seal image and thus the registration using the proposed method will be affected significantly. The acceptable results of using the same set of images but with 200 dpi resolution are shown in Fig. 36.10. This experiment confirms the fact that if a higher resolution seal image is used, the registration will approach to be more correct. Accordingly, if a seal image is scanned with a sufficient resolution for registration, the proposed method will be a feasible approach.

36.4 Conclusion

An automatic registration method using a contouring analysis to deal with a heavy-noisy and rotated seal pattern has been demonstrated. An effective scheme combined with an SOM algorithm for segmentation has also been involved in the proposed approach. Our experiments show that the proposed approach can obtain a considerable and valuable seal registration result under the sufficient resolution image provided, which can be further applied to a practical application such as bank security system in Chinese society.

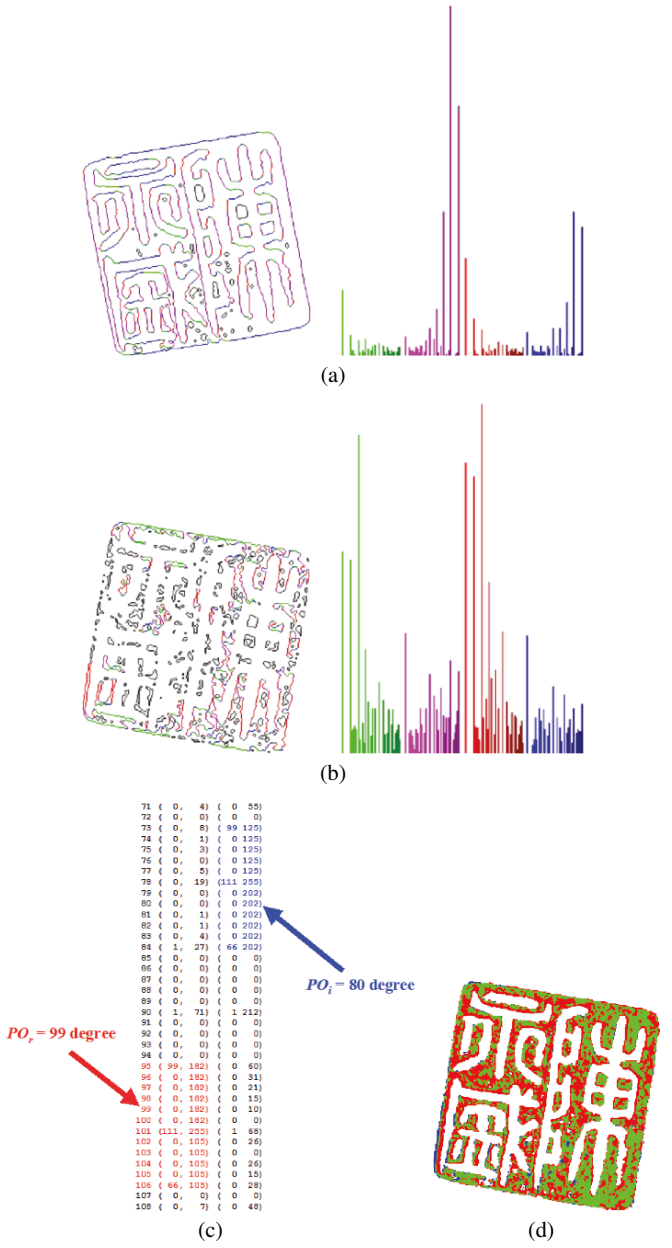


Fig. 36.7 (a) Contour and angle information of the seal in Fig. 36.3(i), and (b) those in Fig. 36.3(l). Their partial CDFs and the determined POs are shown in (c). Based on this information, we have $\Delta PO = PO_r - PO_i = 19^\circ$. After performing the registration procedure for the seals of Fig. 36.3(i) and Fig. 36.3(l), (d) shows the registration result

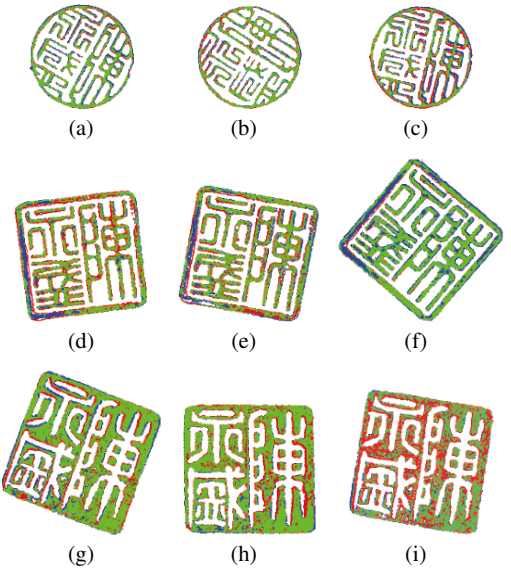


Fig. 36.8 Take the image in Fig. 36.3(a), 36.3(e), and 36.3(i) as reference seals, respectively. Others are regarded as input seals. (a)–(c), (d)–(f), and (g)–(i) show the corresponding registration results

Fig. 36.9 Results of applying our method to 100dpi seal images, where (a) are reference seals, (b) test seals, and (c) the registration results. These results show that the contour information of a low resolution seal image, and thus the registration, will be affected significantly using the proposed method

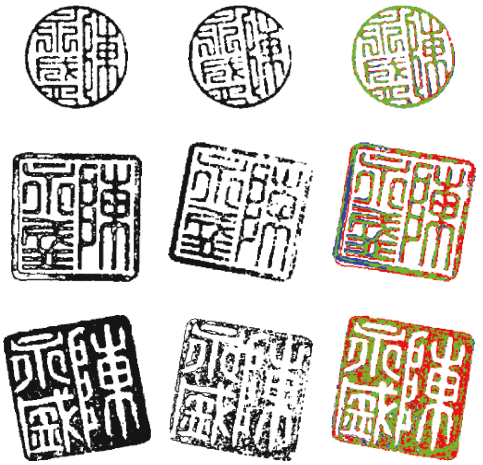


Fig. 36.10 Results of applying our method to 200dpi seal images, where (a) are reference seals, (b) test seals, and (c) the registration results. The registration will approach to be correct if a higher resolution seal image is used

Acknowledgements This work was supported in part by STARTEK Engineering Inc., Taiwan, ROC.

References

1. Brault JJ, Plamondon R (1993) IEEE Trans Pattern Anal Mach Intell 15:953–957
2. Yang Y, Horiuchi T, Toraichi K (1993) Proc. 2nd ICDAR 786–789
3. Chen YS (1996) Pattern Recognit 29:1807–1820
4. Chen YS, Hsu YC (1994) Appl Optics 33:6818–6822
5. Chen YS, Hsu YC (1995) Image Vis Comput 13:463–478
6. Huang HY, Chen YS, Hsu WH (2002) J Electron Imaging 11:136–148
7. Chen YS, Chen BT (2003) Appl Optics 42:1958–1972
8. Shapiro LG, Stockman GC (2001) Computer Vision, Prentice Hall, Inc., New Jersey
9. Gonzalez RC, Woods RE (1992) Digital Image Processing, Addison Wesley, New York
10. Chen YS (2003) LNCS 2749:255–261

Chapter 37
Automatic Localization and Segmentation of Blood Vessels, Optic Disc, and Macula in Digital Fundus Images

S. Balasubramanian, Anantha Vidya Sagar, G. Vijaya Saradhi, and V. Chandrasekaran

37.1 Introduction

One of the major areas of medical research is the design and implementation of intelligent decision support systems for medical professionals. In this context, digital medical image analysis plays an important role in building computational tools to assist physicians in quantification and visualization of pathology and anatomical structures. Such tools will help the medial community to diagnose disorders and treat patients more effectively than before.

Digital fundus images are the images of the fundus occuli acquired using a fundus camera where the optic system of the camera is connected to a CCD. The visible part of the image consists of the retina with its vascular network and the optic nerve head. Study of digital fundus images is important in relation to the diagnosis of diabetic retinopathy (DR), a leading cause of blindness among diabetic patients.

DR is an eye disease that is prevalent among the diabetic population. The WHO projects the increase in the number of diabetics from 130 million to 250 million over the next 25 years. Further, everyone who has diabetes is at risk of developing DR. DR does not exhibit any visual defects until it has proliferated to a larger area of the retina, abruptly resulting in blindness. If detected at an early stage, laser treatment is effective in controlling DR. Hence regular mass screening for DR is important. It potentially reduces the risk of blindness in DR patients by 50 percent [27].

Mass screening for DR would produce a large number of retinal images to be examined by ophthalmologists. Since the ratio of the number of ophthalmologists to the number of images obtained in screening programs is very small, the cost of manual examination is expensive. This calls for an automated system for detection of DR that would distinguish normal retinas from abnormal ones. Such a system

S. Balasubramanian, Anantha Vidya Sagar, G. Vijaya Saradhi, and V. Chandrasekaran
Department of Mathematics and Computer Science, Sri Sathya Sai University, Prasanthi Nilayam, Andhra Pradesh, India 515134
e-mail: bala.ssbals@gmail.com

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

would significantly reduce the workload on ophthalmologists, who could then spend their time and energy only on abnormal images.

An automated DR screening system would examine retinal images as an ophthalmologist would. First, the system would locate various anatomical features like retinal vasculature, OD, and macula. Later these locations would be used as landmarks for the detection of symptoms of DR. A variety of reasons, including variation of the shape of the OD due to retinal disorders, a DR symptom called exudates exhibiting characteristics similar to OD, the possibility of the macula being affected by DR (called maculopathy), the acquisition of fundus images from multiple angles—all make the localization of the aforementioned anatomical features nontrivial. In this chapter we describe a system to locate the anatomical structures from a given digital retinal fundus image automatically. The chapter is arranged as follows:

- The anatomy of the human retina and the effect of DR on the retina are described in Sect. 37.2.
- A literature review of existing methods to locate and segment retinal vasculature and the OD is presented in Sect. 37.3. Once the OD is located, the macula can be localized (as explained later) since it is at a fixed distance from the OD.
- Novel methods to locate and segment retinal vasculature, the OD and the macula are explained in Sect. 37.4. At the end of each method, the results are demonstrated.
- Conclusions and suggestions for future research are mentioned in Sect. 37.5.

37.2 The Retina and Diabetic Retinopathy

The retina is a part of the brain, having been sequestered from it early in development, but having kept its connections with the brain through a bundle of fibers called the optic nerve. It consists of three layers of nerve cell bodies separated by two layers containing synapses. The tier of cells at the back of the retina contains the light receptors, the rods and the cones. Rods, which are far more numerous than cones, are responsible for our vision in dim light and do not work in bright light. Cones do not respond to dim light but are responsible for our ability to see fine detail and colored light.

When light enters the eye and strikes the retina, the photoreceptors convert light into nerve impulses, which are then processed by the retina and sent through nerve fibers to the brain. The nerve fibers exit the eyeball at the OD and reach the brain through the optic nerve where the visual image is formed. The OD measures about 1.5 to 2 mm in diameter and it is 3mm nasal to the fovea. At the OD, the retina contains no photoreceptors. The retinal vessels branch out from the OD toward the retinal periphery. Normally, the vessels end about 1 or 2 mm before the retinal periphery. Two of the large vessels branching out of the OD are located around the center of the retina and are called the vascular arch. The center of the retina is called the fovea, which is the region of high visual acuity and cone cell density. The area in and around the fovea is called the macula. It is made up mostly of cone cells

and it is responsible for central vision and contains the fovea. The macula is less than 5 percent of the total retina. The anatomy of the eye and a gross anatomy of the retina are displayed in Figs. 37.1 and 37.2, respectively [32–35].

Symptoms of DR influence the appearance of the human retina. The progress of DR can be subdivided into four stages.

- Early non-proliferative diabetic retinopathy (NPDR): At this earliest stage, few microaneurysms occur. They are small areas of balloon-like swelling in the retina's tiny blood vessels.
- Moderate NPDR: More microaneurysms occur at this stage. Some blood vessels that nourish the retina are blocked. Exudates form which are the leakage of the surrounding capillaries and microaneurysms. Exudates may have characteristics similar to the OD in terms of color, shape, and size.
- Very advanced NPDR: Many more blood vessels are blocked, depriving several areas of the retina from their blood supply. These areas appear as white/yellowish

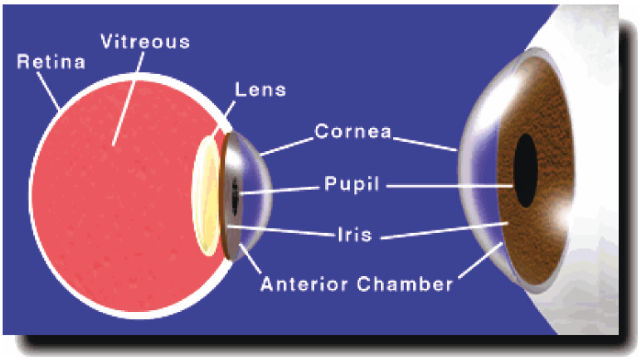


Fig. 37.1 The anatomy of the eye [35]

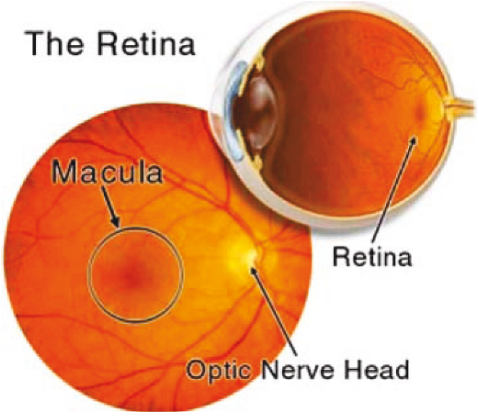


Fig. 37.2 Location and appearance of the retina [34]

spots, called cotton-wool spots. The cotton-wool spots send signals to the body to grow new blood vessels for nourishment.

- Proliferative diabetic retinopathy (PDR): At this advanced stage, the signals sent by the retina for nourishment trigger the growth of new blood vessels. These new blood vessels are abnormal and fragile. They grow along the retina and along the surface of the clear, vitreous gel that fills the inside of the eye. By themselves, these blood vessels do not cause symptoms or vision loss. However, they have thin, fragile walls. If they leak blood, severe vision loss and even blindness can result.

Some of the symptoms of DR are depicted in Fig. 37.3. Exudates may exhibit characteristics similar to the OD. Further, the OD, the macula and the retinal vasculature may undergo structural changes in a diseased retina. Therefore, localization and segmentation of the OD, macula, and blood vessels in the retina is an important prerequisite for automatic detection of DR.

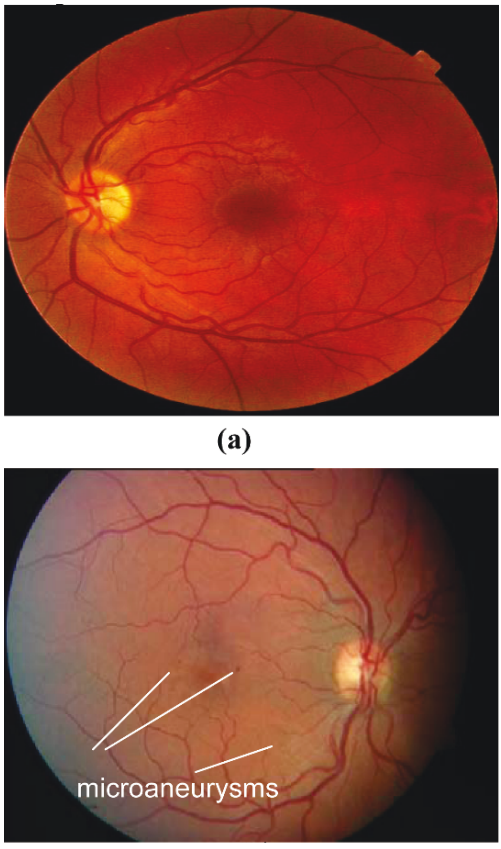
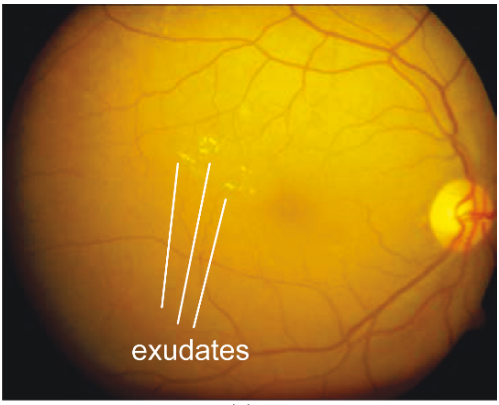
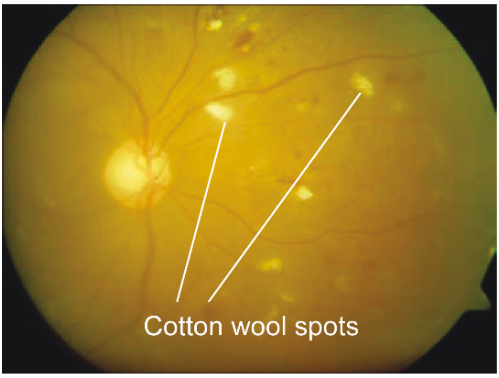


Fig. 37.3 Normal and abnormal DR images **(a)** a normal retinal image **(b)** presence of microaneurysms **(c)** presence of exudates **(d)** presence of cotton wool spots



(c)



(d)

Fig. 37.3 Continued.

37.3 Literature Review

37.3.1 OD Localization and Segmentation

Reliable and efficient OD localization and segmentation are important tasks in an automated DR screening system. It is required as a prerequisite in many algorithms applied for identification of other anatomical structures like blood vessels and macula, and pathologies like diabetic lesions in the retinal images. In blood vessel tracking approaches, the blood vessels in the neighborhood of the OD are used as seeds for vessel tracking [2]. The constant distance between the OD and the macula can be used as a priori method for locating the macula [16]. Segmentation of the OD and its removal as a false positive is an essential step in the exudates detection [17]. For these reasons, OD localization has attracted a lot of contributions to the literature of retinal image analysis.

A broad view of the literature shows that the OD is localized based on the following properties:

1. It is the largest region that consists of pixels with the highest gray levels.
2. It is the area with the highest variation of intensity of adjacent pixels.
3. Its shape is circular.
4. It is the convergence point of the major blood vessels.

Various approaches in the literature can be classified as follows:

- methods which use the intensity values of the image alone, and exploit the first two properties;
- methods based on the structure of the OD; and
- methods based on the geometrical location, which essentially uses the last property.

37.3.1.1 Methods Based on Intensity Values Alone

Following local contrast enhancement, Sinthanayothin et al. [24] localized the OD as the area with the highest variation in the intensity of adjacent pixels. A variance image is determined by evaluating the variance of an 80×80 subimage at each pixel. An average filter of size equal to that of the variance filter is applied on the resultant image. Location of maximum of this image is taken as center of OD. 99.1 percent of sensitivity and specificity are reported. In [22] Osareh et al. employ template matching to localize the OD. A fixed number of manually segmented OD regions from a variety of retinal images are averaged to compute the template. A correlation image is obtained by moving the template across the image as a window and recording the correlation value at each pixel. The maximum of the correlation image is estimated to be the approximate center of OD. The author reports 100 percent success on their database of 75 images. A method elucidated in [2] pursues the following steps:

1. Pixels with the highest 2 percent gray levels are selected.
2. A simple clustering mechanism is used to form clusters.
3. Finally, the centroid of the largest cluster is chosen as the OD center.

Principal component analysis (PCA)-based localization of the OD is expressed in [12, 16]. Candidate regions are first determined by identifying the areas with the highest variation in intensity of adjacent pixels. PCA is applied to the candidate regions. The minimum distance between the original retinal image and its projection on to the disc space is located as the center of the OD.

37.3.1.2 Methods Based on the Structure of OD

Lowell et al. [17] proposed a method based on the observation that the OD consists of a high intensity near-circular rim, with a significant vertically oriented, roughly

centrally located band of low intensity blood vessels. A Laplacian of the Gaussian template with a vertical channel in the middle to correspond to the major vessel band is correlated with the intensity component of the image. The location recording the maximum correlation is estimated to be the approximate center of the OD. The authors report a detection performance of 99 percent on their database consisting of 90 images. In [19] candidate regions are obtained by automatic thresholding where the roundness of the candidate regions is a parameter that determines the threshold. A Hough transform is then applied to extract the circular candidate regions. Subsequently the center of the circle with the highest average intensity is chosen as the center of the OD. A success rate of 90.25 percent on the DRIVE [36] database is reported. Gray level morphological closing is a preprocessing operation to suppress the blood vessels in a similar approach as [1]. A Gauss minimum blur morphological edge detection and Hough curve fitting are the key techniques employed in [28] to localize the OD. Kaupp [13] exploited split-and-merge segmentation, followed by feature based classification. The features used for classification include region intensity and shape.

37.3.1.3 Methods Based on Geometric Location

This approach essentially relies on the information provided by the vessel structure, i.e., the fact that all retinal vessels originate from the OD. In [14], Hough transform, eigen image analysis, and geometrical analysis based on a vasculature model are the three tiers that aid in tracking the OD in optical coherent tomography (OCT) images. In [10], the blood vessels are broken into segments at their branch points. Each of these is modeled as fuzzy segments and is used to determine the area of convergence of the blood vessels. This method was tested on 31 healthy images with 100 percent success and 50 images of diseased retinas with 88 percent success. An algorithm for localization of the OD in low-resolution images is portrayed in [15]. The design of this algorithm relies on the combination of two procedures: (i) a Hausdorff-based template-matching technique on the edge map; (ii) a pyramidal decomposition for large scale object tracking. The performance against a database of 40 images of: (i) various visual quality and retinal pigmentation and (ii) normal and small pupils resulted in an average error of 7 percent on the OD center positioning, with no false detection. The OD can be located based on the preliminary detection of the main retinal blood vessels. To describe the general direction of retinal vessels at any given position in the image, a geometric parametric model is proposed in [8], where two of the model parameters are the coordinates of the OD center. Model parameters are found by simulated annealing. This method achieved 95 percent success on 81 images. In [20], the position of the OD is identified by inference from the location of a set of landmarks placed on various anatomical structures. The algorithm relies on a specific energy function that combines global and local cues. A success rate of 91 percent in 1100 images was achieved.

37.3.2 OD Boundary Detection

Following localization, the OD must be segmented accurately. Two approaches for the segmentation of the OD are found in the literature. In the first approach, detection of the OD boundary is based on treating the OD as a circle or an ellipse. Here a Hough transform as in [24] is used to obtain the estimated circle of the OD. Alternatively, a Hausdorff-based matching between the detected edges and circular templates of different sizes [15] is performed. The second approach is based on deformable contour models. A genetic algorithm explores the combinatory space of possible contours (solutions) by means of crossover and mutation, followed by the evaluation of fitness and the selection of a new set of contours [2]. The cumulative local gradient is used as a fitness function to find the fittest contour. Pallawala et al. [23] presented an algorithm to detect the OD based on wavelet processing and ellipse fitting. The Daubechies wavelet transform approximates the OD region. Then an abstract representation of the OD is obtained using an intensity-based template. An ellipse-fitting algorithm is subsequently utilized to detect the OD contour from this abstract representation. Additional wavelet processing is performed on the more complex cases to improve the contour detection rate. An accuracy of 94 percent on 279 images was reported. Mendels et al. [18] relied on morphological preprocessing and active contours. The contour was driven by a novel external image-derived field called the gradient vector flow (GVF). This reduces the need for accurate initialization of the contour. Results are demonstrated for only nine images. In [17], James Lowell et al. propose a global elliptical model and a local deformable model with variable edge strength dependent stiffness. Though this method is robust, its performance is not perfect because of the variable nature of the images and the presence of distracter boundaries concentric with the desired rim, which may be located either inside or outside the rim. The difficulties in these methods are a result of occlusion of the OD disc boundary due to the convergence of blood vessels in the OD disc region. To overcome the above problem, Osareh et al. [22] define a lab space color morphology preprocessing step which completely removes the blood vessels and performs segmentation by active contours using GVF. This method results in accurate segmentation to a large extent. But it fails in cases where nonhomogeneity of the OD still persists. The accuracy of OD boundary detection using active contours is based on the homogeneity of the OD region.

Once the OD is segmented, the macula can be easily localized since it is at a fixed distance from the OD. In [24] the macula is determined by a matching correlation, together with the characteristic of the fovea that it is the darkest area in the vicinity of the OD. The macula is localized in [9] by locating the darkest pixel in the coarse resolution image following a priori geometric criteria based on the eye's anatomy. Since blood vessels are also dark, and are present throughout the image, finding the darkest pixel may not always give the correct result.

37.3.3 BVD

Digital fundus retinal images are used by doctors for diagnosis and treatment of various retinal disorders. In this context, robust segmentation of blood vessels is of the utmost importance. First, it can be used for early detection of diabetic retinopathy by calibrating it with the normal blood vessel diameters. Second, it can be used in the diagnosis of severe NPDR, which can be treated by laser surgery. Furthermore, blood vessel segmentation is a prerequisite for localizing the OD [8, 10].

A survey of vessel extraction techniques and algorithms is presented in [4]. Algorithms for detecting blood vessels are generally grouped as pixel processing-based methods and tracking methods [3]. The pixel processing is divided into kernel-based and classifier-based methods. Other classification schemes are proposed in the literature, such as the rule-based methods and supervised methods mentioned in [25]. Design of 2D matched filters for BVD is explained in detail in [6]. This approach is computationally expensive due to the large size of the convolution kernel. Other matched filter approaches using global [5] and local [11] thresholding strategies have been reported for the segmentation of retinal vessels.

Zana et al. [29] proposed an algorithm that combines morphological filters and cross-curvature evaluation to segment vessel-like patterns in retinal images. Another mathematical morphology approach to retinal vessel segmentation which combines the detection of centerlines with multiscale morphological enhancement is described in [3].

Neimeijer et al. presented a vessel segmentation algorithm based on pixel classification using a simple feature vector [21]. A method based on the extraction of ridges, called the primitive-based method, was proposed by Staal et al. in [25]. A supervised multilayer perceptron neural network has been employed for BVD in [13]. In [9] a tracking-based approach with recursive dual edge tracking and connectivity recovering is applied for BVD. This procedure has been tested only on 5 images. The method for BVD in [7] is based on quadtree decomposition and postfiltration of edges. However, this approach does not address the problem of noise reduction.

37.4 Proposed Methods for Automatic Detection of Anatomical Structures in Digital Fundus Images

As is evident from the literature, the OD could be localized based on the heuristic that all the major blood vessels converge in the OD, or that it is the largest region that consists of pixels with the highest gray levels, or a combination of both. We propose a novel method for BVD based on morphology. We subsequently utilize the method to localize the OD. In our localization of the OD, we use both heuristics in combination. Once the OD is localized, we make use of the method prescribed in [26] that homogenizes the region of the OD and employs the active contour to

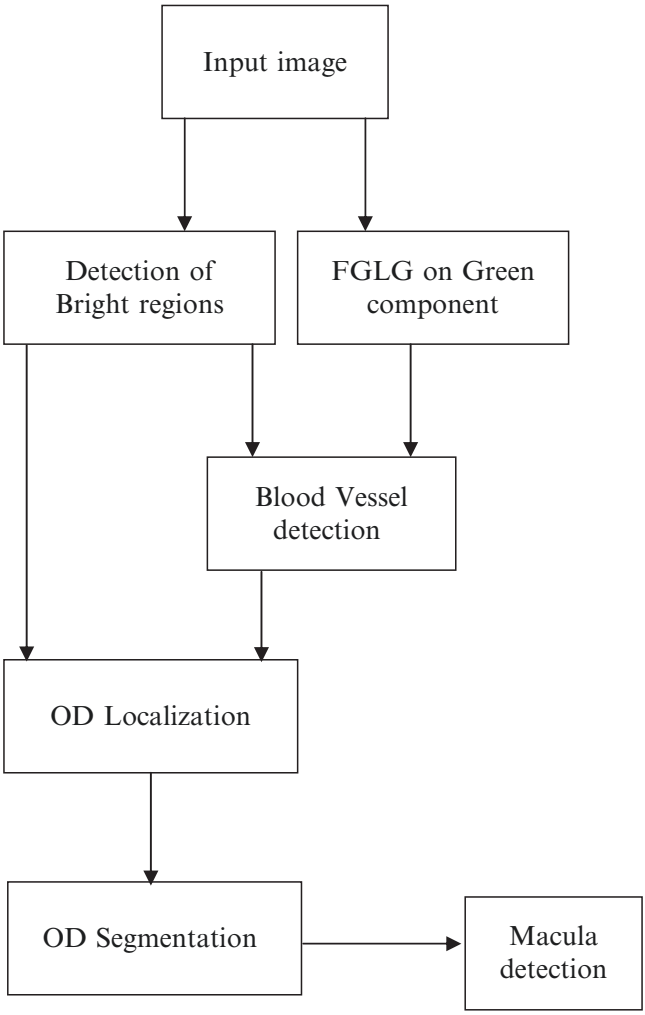


Fig. 37.4 Schematic diagram of the proposed system

detect the boundary of the OD. Following OD segmentation, the macula is localized. The schematic diagram of the proposed system is shown in Fig. 37.4. The proposed approach is implemented in MATLAB (version 7) on a P4 machine (3.00 GHz). It was tested on the DRIVE database [36] and on our own database that includes 100 normal and 480 diseased images of resolution 565×584.

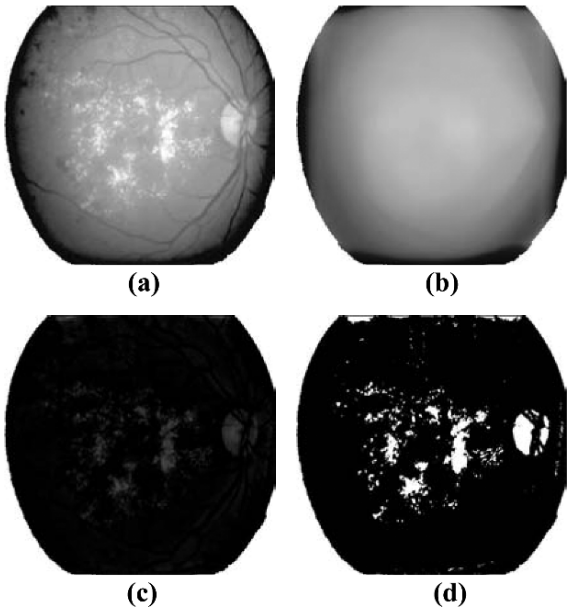


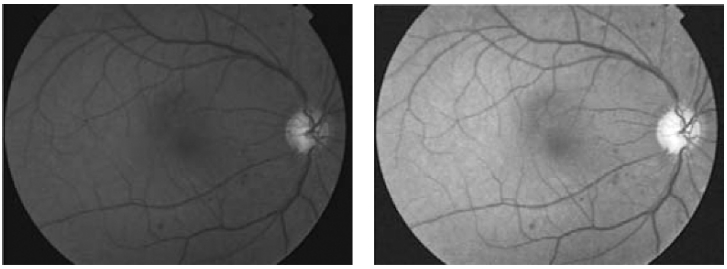
Fig. 37.5 Detection of bright regions: **(a)** green channel of the test image, **(b)** median filtered image, **(c)** difference map of the images in **(a)** and **(b)**, **(d)** detection of bright regions

37.4.1 Detection of Bright Regions

It is noted that the intensity of the OD is much higher than the surrounding retinal background. Therefore a common method to localize the OD is to find the largest cluster of pixels with the highest gray level. This will not always succeed, because the manifestations of DR in the form of exudates have a brightness similar to that of the OD. Hence we first determine all the bright regions in the image. The green channel of the RGB image is chosen for this purpose, since the contrast of the image is maintained well in this channel. A smoothing operation is performed to remove noise. Then large size median filtering (we used a 100×100 filter) is applied to the smoothed image. Subtracting the median filtered image from the green channel of the original image highlights the bright regions. This difference image is thresholded to obtain the bright regions. We set the threshold at 40. Figure 37.5 shows the result on a sample image.

37.4.2 Gray Level Grouping (GLG) for Contrast Enhancement

There is degradation in the quality of contrast as we move from center towards the boundary of the fundus image. Also, blood vessels usually have poor local contrast. So a preprocessing method is applied for contrast enhancement. The green channel



(a) **(b)**

Fig. 37.6 FGLG for contrast enhancement: **(a)** green channel of a test image, **(b)** image after contrast enhancement

of the RGB space is chosen for this purpose because blood vessels appear most contrasted in this channel. Conventionally, methods such as histogram equalization, adaptive histogram equalization, and histogram specification are used for contrast enhancement. For our purpose, we make use of gray level grouping (GLG) [30], as it is an automatic method unlike histogram specification, and shows improved performance over existing methods.

The basic procedure for gray level grouping is as follows.

1. First, the histogram components of the input image are grouped into a proper number of gray level bins according to their amplitudes.
2. These groups of histogram components are redistributed over the grayscale, so that each group occupies a grayscale segment of the same size as the other groups, and the concentrated histogram components spread out and image contrast is increased.
3. Ungrouping the previously grouped gray levels and mapping the gray level values of the pixels in the input image to the desired values in the output image is performed.

A variation of GLG called fast gray level grouping (FGLG) [30] has been applied to the images. Fig. 37.6 shows a sample of the results.

37.4.3 BVD

Some basic morphological operations are defined as follows.

Let $D_f \subseteq Z^2$ and $T = \{\min\ t, \ldots, \max\ t\}$ be an ordered set of gray levels. A gray level image f can be defined as a function f: $D_f \subseteq Z^2 \rightarrow T = \{\min\ t, \ldots,\ \max\ t\}$. Let B be a subset of Z^2 and $s \in N$ a scaling factor; the morphological operations are defined as follows.

Erosion: $[\varepsilon^{(sB)}(f)](x) = \min_{b \in sB} f(x+b)$
Dilation: $[\delta^{(sB)}(f)](x) = \max_{b \in sB} f(x+b)$

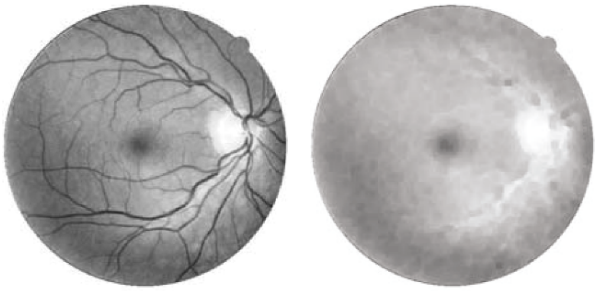


Fig. 37.7 Closing: **(a)** input image, **(b)** result of morphological closing

Opening: $\gamma(\mathrm{sB})(\mathrm{f}) = \delta^{(\mathrm{sB})}[\varepsilon^{(\mathrm{sB})}(\mathrm{f})]$
Closing: : $\varphi^{(\mathrm{sB})}(\mathrm{f}) = \varepsilon^{(\mathrm{sB})}[\delta^{(\mathrm{sB})}(\mathrm{f})]$

Our method for BVD is as follows. Let g represent the green channel of the test image after contrast enhancement. First, blood vessels are eliminated by a closing operation. A square structuring element $\mathrm{s_1B}$ is used for the purpose. $\mathrm{s_1}$ is chosen such that $\mathrm{s_1B}$ is larger than the maximal width of the blood vessels (we used an 11×11 structuring element).

$$\mathrm{h_1} = \varphi^{(\mathrm{s}1^{\mathrm{B}})}(\mathrm{g}) \tag{37.1}$$

where $\mathrm{h_1}$ is the closed image. Figure 37.7 shows a sample result of the closing operation.

Then image g is subtracted from the closed image $\mathrm{h_1}$.

$$\mathrm{h_2} = \mathrm{h_1} - \mathrm{g} \tag{37.2}$$

As blood vessels are relatively darker compared to the background, the values at the locations of vessel pixels in $\mathrm{h_2}$ will be higher compared to the values at the locations of non-vessel pixels. Hence thresholding $\mathrm{h_2}$ at level $\mathrm{t_1}$, where $\mathrm{t_1}$ is chosen appropriately, identifies blood vessel pixels (we set $\mathrm{t_1}$ at 20). Post processing is performed for reduction of noise. The noise caused by bright regions whose size matches that of blood vessels is removed. Small isolated regions of pixels misclassified as blood vessels are removed using connected component labeling, choosing size as the criterion.

Our intention was to detect major blood vessels that could be subsequently utilized to localize the OD. More specifically, on a DRIVE test set [36] our test reported sensitivity and specificity of 70.14 and 96.44 percent, respectively, and on a STARE test set [37] it reported sensitivity and specificity of 64.34 and 97.08 percent, respectively. Figure 37.8 shows the result of our approach compared to the one in [7].

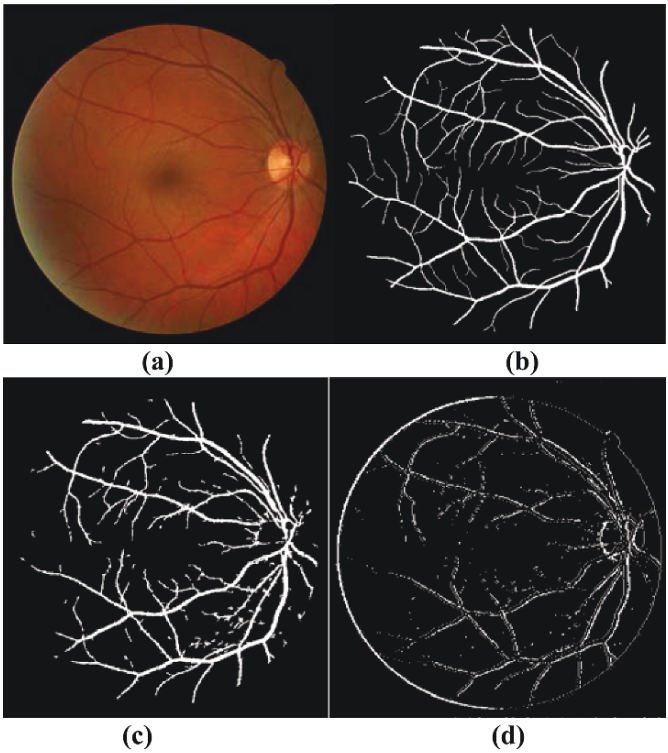


Fig. 37.8 Results of BVD: **(a)** test image, **(b)** manually labeled image, **(c)** BVD by our method, **(d)** BVD by [7]

37.4.4 OD Localization and Segmentation

Since the diameter of the OD is in the range of 65–100 pixels in a retinal image of size 565×584, our aim in this step is to find a square of size 110×110 pixels that contains the OD. We worked on the green channel of the RGB image for this purpose. Figure 37.9 shows the green channel of two retinal images. As can be observed, the OD is a bright region with a high density of blood vessels. There are other regions such as exudates which are also bright, but the density of blood vessels in these regions is low. We exploit this property for OD localization.

Let the original image be denoted by A. Then a count matrix C whose size equals that of A is created such that:

1. for all noncandidate pixels C(i, j) equals zero, and
2. for all candidate pixels C(i, j) equals the number of blood vessel pixels in the 110×110 square with A(i, j) as the center.

Since the density of blood vessel pixels is high in the OD region, it is observed that the count C(i, j) is large for all the bright pixels in the OD. We propose that the

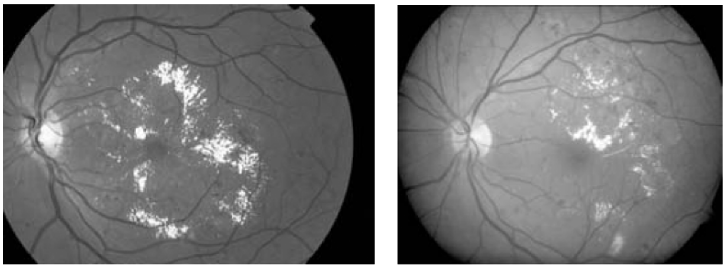


Fig. 37.9 Test images: **(a)** test image 1, **(b)** test image 2

110×110 square S for which the sum of C(i, j)'s of pixels is the maximum contains the OD. Our method is justified as follows. In the case of normal images, the OD is the only bright region, which justifies the approach. In the case of diseased images, bright lesions such as exudates also get detected as candidate regions. As can be observed in Figs. 37.9 and 37.10, the density of blood vessels in these lesions is less compared to that of the OD region. Because blood vessels converge at the OD, the bright pixels in the OD are divided into a number of clusters rather than a single cluster. This can be noticed in Fig. 37.5. It can be noted that the count C(i,j) is large for each pixel in the OD region as compared to pixels in other areas of the image, and that all the clusters of OD region can fit into a square of size 110×110. Therefore the proposed approach leads to OD localization even on diseased images.

On normal images OD was localized 100 percent correctly, with an average execution time of 9 sec. On diseased images OD was localized correctly 93.75 percent of the time, with an average execution time of 17 seconds. Our algorithm is simple yet robust, reliable, and fast. The fuzzy convergence approach to OD localization [10] reports only an 89 percent success rate on diseased retinal images. The PCA-based method in [12] has been tested on the same diseased images and it achieves a success rate of only 60 percent. Also, the average execution time of this algorithm is 8 minutes, compared to 17 seconds for our algorithm. Figure 37.10 shows the results of our algorithm on some images.

Once the OD is localized, we follow our method proposed in [26] to detect the boundary of the OD. Here the blood vessels converging in the OD region are removed and subsequently homogenized. Following this, the boundary of the OD region is determined using GVF snakes [26]. Figure 37.11 displays a couple of results. We compared our OD segmentation routine in [26] to the method proposed in [22] where the OD is not completely homogenized. Our method outperformed the one described in [22]. For details readers are referred to [26]. A set of comparative results is shown in Fig. 37.12.

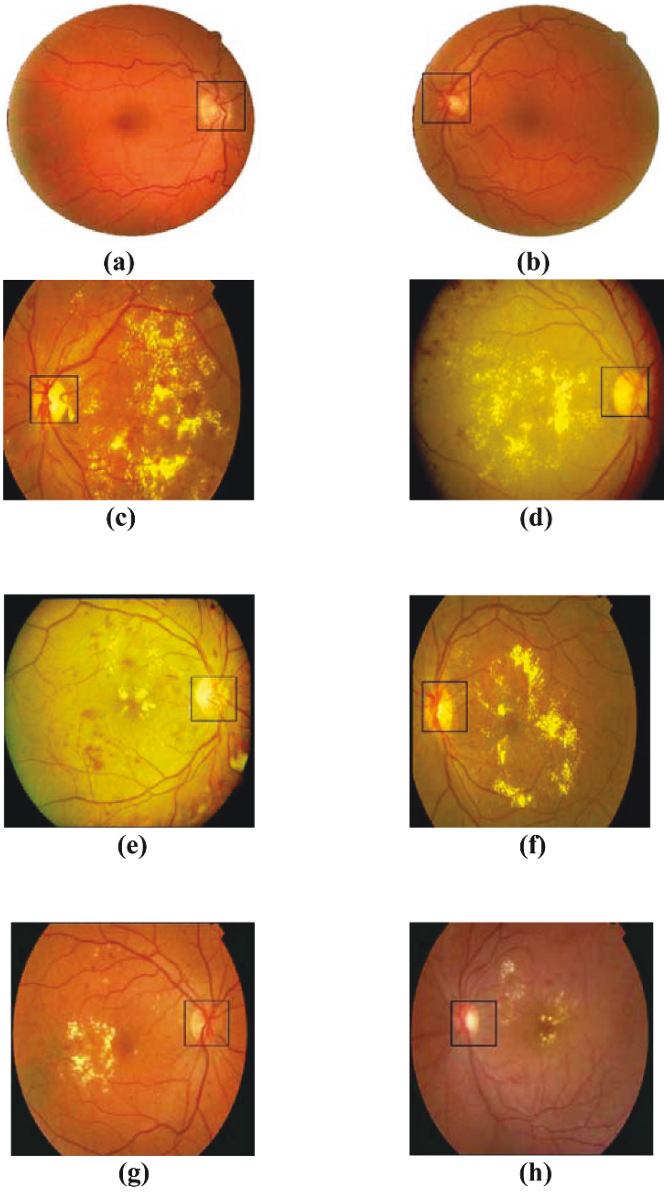


Fig. 37.10 Results of OD localization. The black square represents the localized OD: **(a)–(b)** results on normal images, **(c)–(h)** results on diseased images

37.4.5 Macula Detection

The fovea is usually located at a distance from the center of the OD of approximately 2.5 times the diameter of the OD [24]. As the OD is already localized

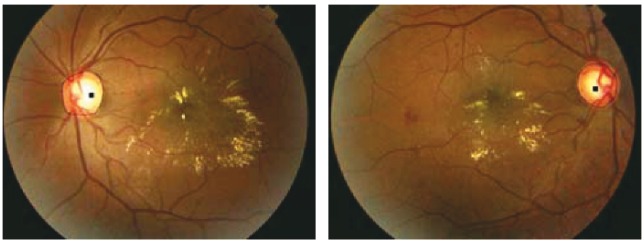


Fig. 37.11 OD segmentation. The black square represents the center of the OD estimated after OD localization

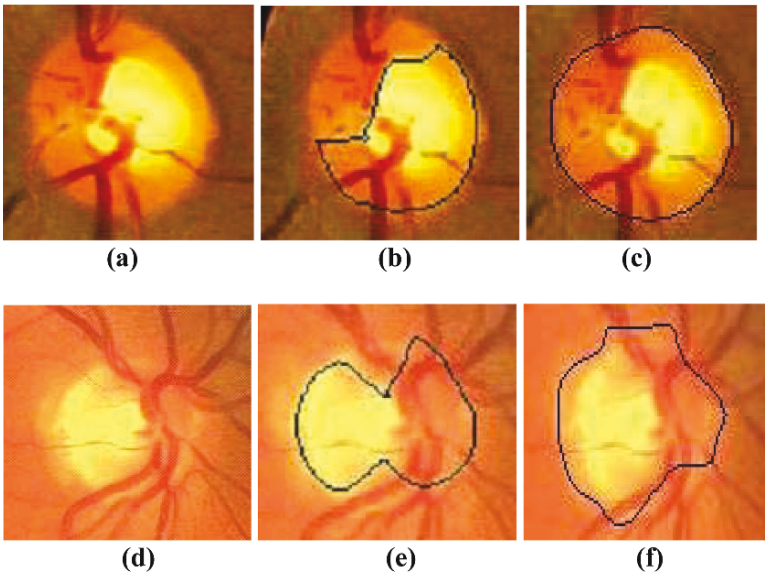


Fig. 37.12 OD segmentation: **(a)** and **(d)** input images, **(b)** and **(e)** segmentation by [22], **(c)** and **(f)** segmentation by [26]

and segmented, the approximate diameter of the OD is obtained from its boundary. Let the diameter of the OD be d. The proposed method for macula detection is as follows:

First the region of interest (ROI) is determined from the geometric criteria based on the eye's anatomy. The ROI contains only those pixels in the image whose radial distance r from the center of the OD satisfies the following equation:

$$(1.5 \times \mathrm{d} < r < (3.5 \times \mathrm{d}). \tag{37.3}$$

The algorithm assumes that it is known a priori whether the image is centered on the macula or not. If the image is centered on the macula then the ROI contains pixels that are oriented towards the center of the image and satisfy (3). Otherwise the ROI contains all the pixels in the image which satisfy (3). Figure 37.13 shows the

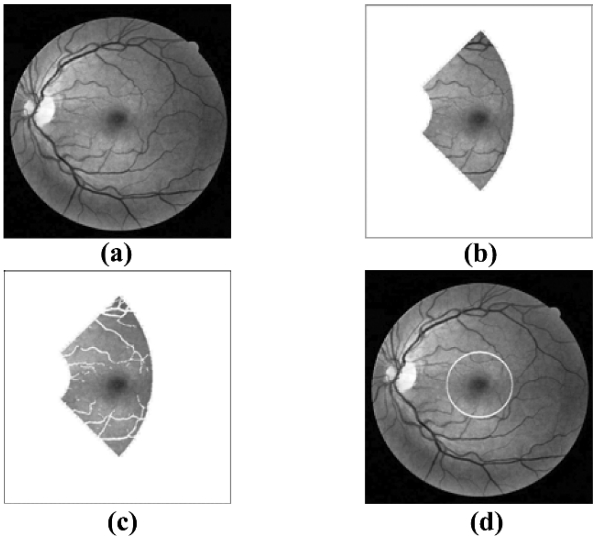


Fig. 37.13 ROI detection: **(a)** test image, **(b)** result of ROI detection, **(c)** masking out the blood vessels, **(d)** result of macula detection

ROI for a sample image. Pixels which do not belong to the ROI are marked white. The result of the BVD is used to mask out vessel pixels in the region of interest. Then we determine the darkest 1 percent of pixels in the ROI. These pixels are clustered together, and the largest cluster is localized as the macula. A circle with diameter equal to twice the diameter of the OD is drawn to represent the macula region. This method reported a success rate of 96 percent.

37.5 Discussion, Future Research, and Conclusion

Figure 37.14 shows another set of results of our approach for the automatic detection of anatomical structures in the retina.

An automatic system developed for screening background DR identifies symptoms like microaneurysms and exudates. For this purpose, anatomical structures including the OD are initially segmented out. BVD can be employed as a precursor to OD localization. In this case, it suffices if the BVD algorithm detects all the major vessels, even if some of the minor capillaries are missed. Other applications like hypertensive retinopathy characterized by vessel tortuosity, macular avascular region detection may require accurate detection of all the blood vessels in the retina. This is an important point to be noted since it may reduce a lot of the computational burden of the BVD routine. Localization and segmentation of the OD is significant from the point of view of the detection of exudates. The area of the OD need not be considered for exudates detection, thereby reducing the number of false positives.

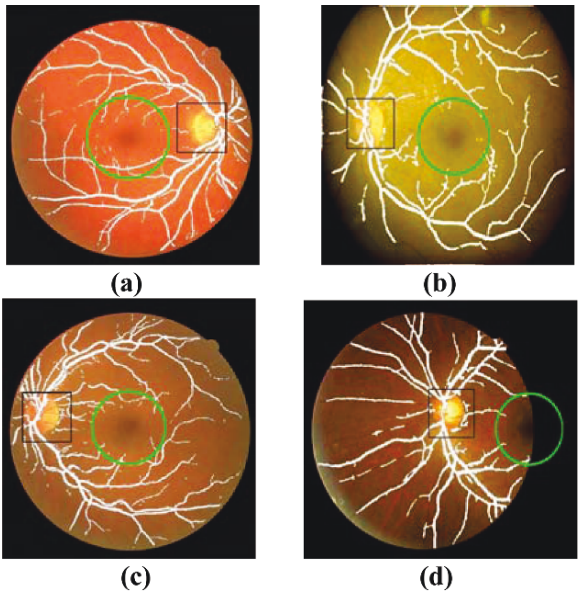


Fig. 37.14 Detection of anatomical structures in the retina. The black square represents the localized OD, the green circle represents the macula region, and the blood vessels are marked white

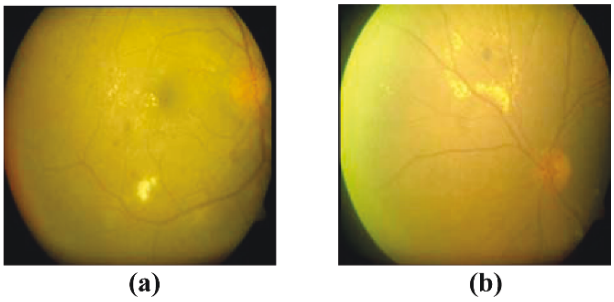


Fig. 37.15 15 Images in which our method failed to detect OD

We show in Fig. 37.15 a couple of cases where our method failed to localize the OD correctly.

The proposed algorithm failed in these images because the OD is not the brightest in these images. The presence of illumination imbalance could pose problems for OD localization, as can be seen in Fig. 37.15(b). But with the advancement of technology and the availability of superior quality fundus cameras, we may assume without any loss in generality that such images may not occur in practical situations.

In conclusion, we have proposed a robust system for automatically detecting anatomical structures in the retina including blood vessels, the OD, and the macula that could be useful in the automatic detection of DR. But there is lot of scope for further research.

An important aspect of DR is the microvascular changes that cause detectable changes in the appearance of the retinal blood vessels. During DR, the fine vessels on the surface of the retina become damaged, a condition that can be treated by laser surgery if detected early. Finding the caliber of the retinal blood vessels can help in identifying microvascular changes. Hence the prediction of DR based on changes in that caliber provides scope for further research in BVD.

In the localization of the OD, we see from the literature that a few attempts were made in the past to locate the OD using genetic algorithms [2] and wavelets [23]. These methods could be improved further by employing the latest developments in the fields of genetic algorithms and wavelets. We found that the quality of the images was not consistent. This brought to light the need for a metric to quantify the quality of an image. A study of image quality metrics in the case of retinal images, and a way of standardizing all the input images to a specified image quality level, would be a worthy contribution.

Acknowledgements We dedicate this work to our chancellor, Bhagawan Sri Sathya Sai Baba. We also acknowledge the use of images provided by [31]. Our thanks are due to SSSIHMS (Puttaparthi, India), Dr. Rema Mohan (MVDSC, Chennai, India), Dr. Rajiv Raman (Shankara Netralaya, Chennai, India), Dr. Puvana Chandra (UK), Prof. Panchanathan (ASU, USA), Dr. Bhujanga Shetty, and Dr. Venkata Subramanian from Narayana Netralaya, Bangalore, India, for providing us images and valuable information on DR.

References

1. Abdel-Ghafar RA, Morris T, Ritchings T, Wood I (2004) Detection and characterisation of the optic disk in glaucoma and diabetic retinopathy. In: Proc of medical image understanding and analysis, London
2. Abraham Chandy D, Vijaya Kumari V (2006) Genetic algorithm based location of optic disc in retinal images. Academic Open Internet Journal 17
3. Mendonca AM, Campilho A (2006) Segmentation of retinal blood vessesl by combining the detection of centerlines and morphological reconstruction. IEEE Tran Med Imag 25: 1200–1213
4. Kirbas C, Quek FKH (2003) Vessel extraction techniques and algorithms: a survey. In: Third IEEE symposium on bioinformatics and bioengineering, pp 238–245
5. Chanwimaluang T, Fan G (2003) An Efficient blood vessel detection algorithm for retinal images using local entropy thresholding. In: Proc IEEE international symposium on circuits and systems, pp V-21–V-24
6. Chaudhuri S, Chatterjee S, Katz N, Nelson M, Goldbaum M (1989) Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Trans Med Imag 8:263–269
7. Dua S, Kandiraju N, Thompson HW (2005) Design and implementation of a unique blood-vessel detection algorithm towards early detection of diabetic retinopathy. In: Proc of the IEEE international conference on information technology, pp 26–31
8. Foracchia M, Grisan E, Ruggeri A (2004) Detection of OD in retinal images by means of a geometrical model of vessel structure. IEEE Trans Med Imag 23:1189–1194
9. Gagnon L, Lalonde M, Beaulieu M, Boucher MC (2001) Procedure to detect anatomical structures in optical fundus images. In: Proc SPIE Med Imag: Image processing, pp 1218–1225
10. Hoover A, Goldbaum M (2003) Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels. IEEE Trans Med Imag 22:951–958

11. Hoover A, Kouznetsova V, Goldbaum M (2000) Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE Trans Med Imag 19:203–211
12. Huiqi L, Opas C (2001) Automatic location of optic disc in retinal images. In: Proc of the International conference on image processing, pp 837–840
13. Kaupp A, Dolemeyer A, Wilzeck R, Schlosser R, Wolf S, Meyer-Ebrecht D (1994) Measuring morphological properties of the human retinal vessel system using a two-stage image processing approach. In: Proc IEEE international conference on image processing, pp 431–435
14. Koozekanani D, Boyer C, Roberts C, Katz S (2001) Tracking the optic nerve in OCT video using dual eigenspaces and an adaptive vascular distribution model. In: Proc IEEE conf computer vision and pattern recognition, pp 1934–1941
15. Lalonde M, Beaulieu M, Gagnon L (2001) Fast and robust optic disk detection using pyramidal decomposition and Hausdorff-based template matching. IEEE Trans Med Imag 20: 1193–1200
16. Li H, Chutatape O (2004) Automated feature extraction in color retinal images by a model based approach. IEEE Biomed Eng 51:246–254
17. Lowell J, Hunter A, Steel D, Basu A, Ryder R, Fletcher E, Kennedy L (2004) Optic nerve head segmentation. IEEE Trans Med Imag 23(2):256–264
18. Mendels F, Heneghan C, Thiran JP (1999) Identification of the optic disc boundary in retinal images using active contours. In: Proc IMVIP Conference, pp 103–115
19. Park M, Jesse SJ, Luo S (2006) Locating the optic disc in retinal images. In: Proc international conference on computer graphics, imaging and visualisation, pp 141–145
20. Neimeijer M (2006) Automatic detection of diabetic retinopathy in digital fundus photographs. PhD thesis, Image Sciences Institute, University Medical Center Utrecht, Utrecht, The Netherlands
21. Niemeijer M, Staal J, Van Ginneken B, Loog M, Abramoff MD (2004) Comparative study of retinal vessel segmentation methods on a new publicly available database. In: Filtzpatrick M, Sonka M (eds) Proc SPIE Med Imag, pp 648–656
22. Osareh A, Mirmehdi M, Thomas B, Markham R (2002) Comparison of colour spaces for optic disc localisation in retinal images. In: Proc 16th international conference on pattern recognition, pp 743–746
23. Pallawala PMDS, Hsu W, Lee ML, Au Eong K-G (2004) Automated optic disc localization and contour detection using ellipse fitting and wavelet transform. Springer Lecture Notes in Computer Science 3022:139–151
24. Sinthanayothin C, Boyce JF, Cook HL, Williamson TH (1999) Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images. British Journal of Ophthalmology 83:902–910
25. Staal J, Abramoff MD, Neimeijer M, Viergever MA, Van Ginneken B (2004) Ridge-based vessel segmentation in colour images of the retina. IEEE Trans Med. Imag 23:501–509
26. Vijaya Saradhi G, Balasubramanian S, Chandrasekaran V (2006) Performance enhancement of optic disc boundary detection using active contours via improved homogenization of optic disc region. In: Proc of the international conference on information and automation, pp 264–269
27. Wareham N (1993) Cost-effectiveness of alternative methods for diabetic retinopathy screening (letter). Diabetic Care 16:844
28. Yulong M, Dingru X (1990) Recognizing the glaucoma from ocular fundus by image analysis. In: Proc annual international conference of the IEEE engineering in medicine and biology society, pp 178–179
29. Zana F, Klein JC (2001) Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. IEEE Trans Med Imag 11:1111–1119
30. Chin ZY, Abidi BR, Page DL, Abidi MA (2006) Gray-level grouping (GLG): an automatic method for optimized image contrast enhancement —part I: the basic method. IEEE Trans Image Process 15:2290–2302
31. *http://cmm.ensmp.fr/∼walter/*
32. *http://www.siggraph.org/education/materials/HyperVis/vision/retina.htm*

33. *http://www.mdsupport.org/anatomy.html*
34. *http://www.stlukeseye.com*
35. *http://www.lasersite.com/Anatomy/index.htm*
36. *http://www.isi.uu.nl/Research/Databases/DRIVE*
37. *http://www.parl.clemson.edu/stare/probing/*

Chapter 38
Denoising in the Wavelet Domain Using Double Filtering

Bob Paul Raj and V. Ramachandran

38.1 Introduction

The modern era is moving towards handheld miniature devices with all kinds of video, audio, imaging, and gaming applications supported. The applications are no longer looking forward to personal computer-based platforms. Handheld devices or mobile devices have limitations, however. The major challenges faced today in designing embedded systems for imaging applications are memory and computational requirements to achieve the same quality as desktop applications. One such application that demands a very high picture quality and memory requirements, is a medical imaging system.

The state-of-the-art compression models always demand a nearly loss-less compression, which can be transmitted over any medium. Medical imaging standards insist on lossless compression for more accurate diagnosis. The JPEG 2000 standard defines the usage of wavelet transform for lossless and efficient compression practices for medical images. The probability of noise polluting the image data is high during transmission or acquisition. The noise can be additive, multiplicative, or impulsive in nature. To eliminate the noise, a suitable enhancement technique has to be employed that retains most of the features, and the reconstructed image should not be a hindrance for diagnosis. The preprocessing algorithm being used for enhancement of the images should not have high computational complexity and minimum memory requirements because the software implementation has to go into a handheld device. In general, wavelet coefficients of a function are large in regions where there is an irregularity and small in smooth regions. If a function is corrupted with additive noise, the noise dominates the coefficients at small scales and hence most of the coefficients contain the noisy part [1]. The noise has to be eliminated

Bob Paul Raj,
Member, IAENG, Sasken Communication Technologies Ltd., Domlur, Bangalore-29, India

V. Ramachandran
Department of Computer Science and Engineering, Anna University, Chennai-25, India

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

in the wavelet domain. There are many methods that are present to denoise in the wavelet domain, like hard and soft thresholding [2] and Bayesian denoising [3]. Hard thresholding eliminates the less significant coefficients. Donoho and Johnson have determined the optimal threshold for eliminating noise. Soft thresholding shrinks less-significant coefficients more than significant coefficients, and thresholding is also done through Bayesian estimation. Bayesian estimation requires that the mean square error be determined, which can be done by determining the probability density function of the noise and the original image. The assumption in all these approaches is that the distribution of the noise is Gaussian. An attempt is made to study the effect of linear and nonlinear approaches for filtering Gaussian noise [4] and comparing the performance with adaptive filtering using a least mean square (LMS) algorithm, both in terms of quality and the performance on an embedded digital signal processor (DSP). The proposed technique considers real time aspects and makes the software implementation easy. The computational complexity is minimal in comparison to the other existing algorithms. Using this technique, a particular region of interest can also be chosen and texture smoothing can be performed. The algorithm utilizes linear estimation and values of various neighborhood pixels, which is a combination of Wiener estimation and a median filter.

38.2 Wavelet Decomposition and Basis Functions

The concept of multi-resolution underlies the theory of wavelets [1]. The process of decomposition—the separation of the information to be analyzed into its principal parts—should be thought of primarily of low pass and the residual part should be thought of primarily as high pass. The process of decomposing the image function f using Haar wavelets is given by the following equation

$$f = \sum \left\langle f, \psi^j_{j,k} \right\rangle \psi^i_{j,k} \tag{38.1}$$

where

$$\psi^j_{j,k} = 2^j \psi^i_{j,k}(2^j - k), \text{ is the Haar basis function}$$
$$j \in Z,\ k \in Z^2,\ i = 1,2,3$$

The superposition of the wavelet function over an image function can be represented as the inner product of f and Ψ that is defined as

$$\left\langle f, \psi^j_{j,k} \right\rangle = \int f(t)\psi(t)dt \tag{38.2}$$

The Haar wavelet matrices of rank m can be shown as

$$H(m;F) = WM(m,l;F) \tag{38.3}$$

where $H(m;F)$ is the Haar matrix and *WM(m,1;F)* is the wavelet matrix.

Moreover, there is a mapping from wavelet matrices to the Haar matrices of the same rank. The Haar scaling function is defined by

$$\psi(x) = \begin{cases} 1 \; if \; 0 \leq x < 1 \\ 0 \quad otherwise \end{cases} \tag{38.4}$$

The coefficient of expansion is an approximation at any level in terms of the wavelet basis function described for Haar wavelets shown by the scaling function in Eq. 38.4.

38.3 Estimation and Smoothing in the Wavelet Domain

38.3.1 Wiener Estimation

The proposed double-filtering technique for the image enhancement is a combination of linear estimation and a nonlinear filter. Linear estimation on the image is done using the Wiener method. To apply the Wiener estimation technique the covariance matrix has formulated, a block covariance structure can be modeled [5]. A block with strong edges is highly correlated and those with light edges are weakly correlated [6]. A substantial amount of correlation exists for pixels separated by scales [3]. The correlation of the pixels in the strong edges and weak edges is utilized in Weiner estimation after the image has been decomposed using wavelets. A generalized block diagram for linear estimation [7] is shown in Fig. 38.1, where the image restoration suppresses the Gaussian noise.

The technique emphasizes filtering and prediction. The transfer function $G(\omega)$ of the Wiener filter in the frequency domain is given by Eq. 38.5

$$G(\omega) = S_{uu}/(S_{uu} + S_{\eta\eta}) \tag{38.5}$$

where s_{uu} and $s_{\eta\eta}$ are the estimated power spectra of image and noise, respectively. Many applications vary slowly with time and an adaptive system that can track these changes uses short-time windowed estimates of correlation functions that are near optimal at all times. Better pixel estimates can be computed for small windows because larger windows are slower to adapt. A Wiener filter is an optimal mean square error (MSE) stationary linear filter and can be applied in the frequency domain.

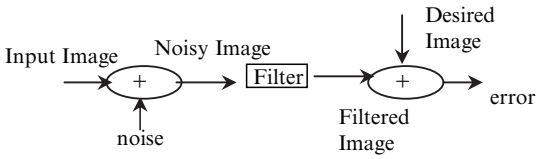


Fig. 38.1 Block diagram illustrating optimum linear estimation

To build an optimal estimate, knowledge of the original image is known to perform restoration effectively. Wavelet coefficients are better modeled using a stochastic model, and the power spectrum is better estimated. Hence, both inverse filtering and noise smoothing in the wavelet domain can be performed. The assumption made here is that the image and noise are considered to be wide sense stationary with constant mean.

38.3.2 Median Filters

The median filter uses the method where the gray level of the pixel is replaced by the median of the neighboring pixels [8]. The technique described here preserves the edge sharpness and is a nonlinear filtering approach [9]. For two sequences x(m) and y(m)

$$Median\{x(m)+y(m)\} \neq Median\{x(m)\}+Median\{y(m)\} \tag{38.6}$$

Few other characteristics of a median filter are that the filter can remove isolated lines or pixels while preserving the spatial resolution.

38.3.3 Adaptive Filtering using LMS Algorithms

An LMS algorithm is an important member of the family of stochastic gradients [10]. An LMS algorithm is a linear adaptive filtering algorithm that, in general, consists of two basic processes:

- A filtering process that involves
  1. Computing the output of a linear filter in response to an input signal and
  2. Generating an estimation error e(n) as shown in Eq. 38.7 by comparing this output y(n) with a desired response d(n).

$$e(n) = y(n) - d(n) \tag{38.7}$$

- An adaptive process that involves the automatic adjustment of the parameters in accordance with the estimation error.

An LMS algorithm is obtained by substituting instantaneous estimates in the steepest decent algorithm [11]. The LMS algorithm is recursive in nature for convergence, which depends on the step size and the number of iterations. Due to the recursive nature of the algorithm, these estimates are averaged during adaptation. Direct averaging is also initiated assuming small step size [12]. The error estimate e(n), step size μ, and the input x(n) is a function of the updated estimate $y(n+1)$. The tap-weight vector adaptation for generating an updated estimate is given by Eq. 38.8

$$y(n+1) = y(n) + \mu e(n).x(n) \tag{38.8}$$

The performance of the adaptive LMS adaptive filter depends on the speed of convergence. The speed of convergence is equal to the number of iterations it takes to reach a finite solution. The higher number of iterations has more computations within the loop. The speed of convergence of linear predictors is reduced under these circumstances [13].

- The Eigen value is usually large due to a high correlation of the input signal
- Long filters require a small step size that, in turn, slows down the adaptation process.

38.4 Double-filtering Description

A noisy image can be modeled as shown in Eq. 38.9

$$x(m,n) = s(m,n) + w(m,n) \tag{38.9}$$

where $x(m,n)$ is the resultant noisy image

$s(m,n)$ is the input image
$w(m,n)$ is the additive white gaussian noise.

Gaussian noise is additive and random in nature. All the types of noise are specified with mean, variance, or noise density. It is required to obtain an estimate of a noise-free image from the noisy version $x(m,n)$. The proposed image enhancement algorithm is basically two steps, as shown in Fig. 38.2.

The enhancement operation is carried out in the wavelet domain. For analysis, the image is polluted with noise characteristics of mean zero and variance of 0.01, 0.02, and 0.03. The noisy image is further decomposed using Haar wavelets. The wavelet decomposition function is given by Eq. 38.1 and the Haar scaling function is given by Eq. 38.4. The Haar decomposition on the image matrix involves the differencing and averaging operations on the rows followed by the same operations on the columns of the resultant matrix. The Haar wavelet determination illustrates the ease of implementation at an embedded level because most of the embedded DSP processors have strong arithmetic logic unit (ALU) functionalities and separate video units are also present that perform four 8-bit adds in a single cycle. The Blackfin processor supports single instruction multiple data (SIMD) operation. The resultant image

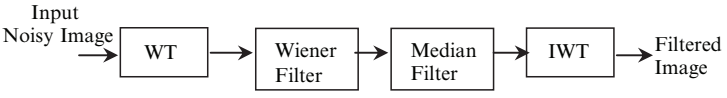


Fig. 38.2 Illustration of the technique using block diagrams

will have four sub-bands consisting of approximation and detailed coefficients. The Wiener filter is applied to each of the sub-bands, resulting in the Wiener filtered image. The correlation between the original image and corrupted image can be determined. The covariance matrix is formulated and Wiener filter coefficients can be estimated.

The coefficients were determined for each sub-band separately by selecting 3×3 blocks or 5×5 blocks. The wavelet transform (WT) defines various sub-bands that demarcate the low frequency components from higher frequency components, wherein the pixel estimation becomes easier unlike the discrete cosine transform (DCT) where the resultant matrix has to be regrouped to obtain higher frequency components and lower frequency components from the left-hand uppermost corner. The blocks of various sizes are chosen to determine the coefficients, which are filtering by linear estimation. Linear estimation will effectively suppress additive or multiplicative noise depending upon the size of the block chosen. The size of the block is chosen such that the edges are also preserved. The second pass involves nonlinear filtering.

The nonlinear filter used here is the median filter. The median filter also uses 3×3 and 5×5 masks for replacing the pixel with the median of the neighborhood pixels in the chosen blocks. The median filter is applied to each sub-band separately, and the filter preserves the edges if the suitable block size is chosen. The nonlinear filter enhances the image even for nonuniform noise, like the salt and pepper noise inverse wavelet transform (IWT) is performed to get the reconstructed and enhanced image.

Choosing a constant step size, and fixing a constant number of iterations, is the key in implementing the LMS algorithm at an embedded level. The number of iterations for convergence has to be constant in an embedded system because for certain cases the convergence might go into an infinite loop and the entire mobile system might stall, taking all the CPU time.

38.5 Results and Discussion

The technique has been tested on CT scan images of the brain of a patient. The test set included ten image slices, each of 512×512 size. All the simulations were carried out in MATLAB and a pseudo-software implementation was done for determining the computational overhead. The ten image slices chosen had different contours and histogram. Each test image was polluted with Gaussian, salt, and pepper, with a mean of zero and variance of 0.01. Figure 38.3 shows a sample CT scan slice with optimal contours. The image in Fig. 38.4 shows a noisy image polluted with Gaussian noise with a variance of 0.01. The image in Fig. 38.5 shows the noisy image decomposed using Haar wavelets. The objective is to determine an optimal window for the Wiener filter and the median filter. We experiment between smaller size windows ranging from 3×3 to 5×5. The reason for choosing smaller windows is to get a better pixel estimate as described in the previous section on Wiener estimation and optimal usage of memory in an embedded system. The set of results

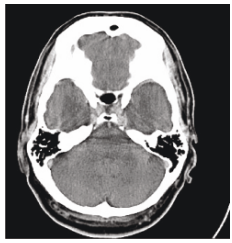


Fig. 38.3 Original image slice obtained from CT scan

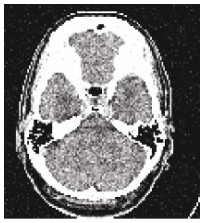


Fig. 38.4 Image slice polluted with Gaussian noise

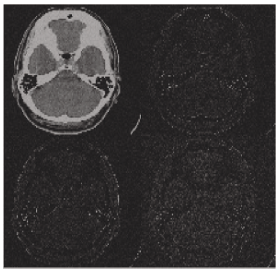


Fig. 38.5 Noisy decomposition of image

Table 38.1 Variations of the PSNR reconstructed from Gaussian noise

	PSNR Variations for windows of Gaussian noise (dB)			
S. No	$3\times3/3\times3$	$5\times5/3\times3$	$3\times3/5\times5$	$5\times5/5\times5$
1	24.798	24.896	28.394	28.46
2	28.229	28.046	29.317	29.211
3	20.893	21.159	28.049	28.755
4	24.118	24.804	28.588	28.252
5	20.62	22.365	28.264	28.194
6	21.889	21.718	27.52	27.38
7	27.77	27.97	29.986	29.644
8	25.698	25.499	28.181	28.457
9	26.989	30.168	31.104	30.401
10	30.376	31.824	34.027	33.762

tabulated in Table 38.1 and Table 38.2 is used to determine an optimal window for the median filter and Wiener filter. The smaller window size chosen for both the Wiener and median filter shows nonuniform variations as shown in Figs. 38.6 and 38.7. As windows get larger, the variations in the PSNR were almost within a finite range of 1–2dB. It is also observed that the larger the size of the median window

Table 38.2 Variations of the PSNR reconstructed from Salt and pepper noise

S. No	PSNR Variations for windows of Salt and Pepper noise (dB)			
	$3\times3/3\times3$	$5\times5/3\times3$	$3\times3/5\times5$	$5\times5/5\times5$
1	21.1138	21.354	34.587	34.405
2	29.2086	29.403	34.301	33.959
3	21.4113	22.125	31.568	30.987
4	25.2629	21.911	30.914	30.744
5	20.8559	21.01	31.249	30.591
6	22.2405	21.479	30.72	31.193
7	23.4305	22.555	33.201	34.667
8	25.5123	21.786	31.736	31.304
9	31.457	27.516	38.781	39.6
10	26.419	37.402	46.092	46.15

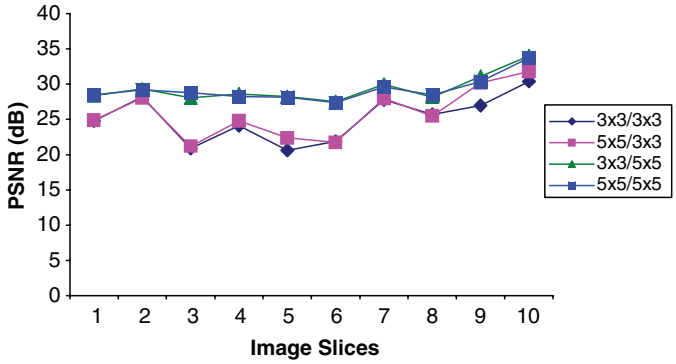


Fig. 38.6 Variations in PSNR of the image reconstructed from Gaussian noise

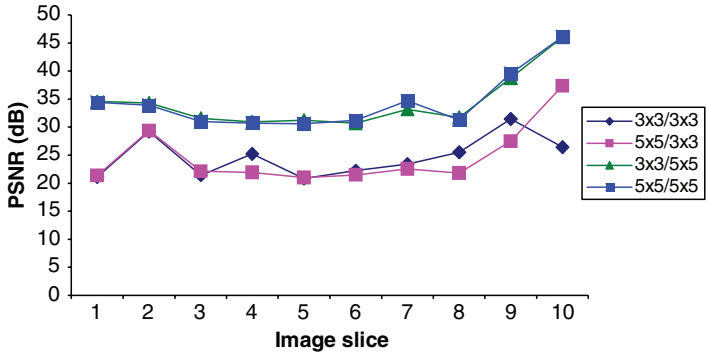


Fig. 38.7 Variations in PSNR of the image reconstructed from salt and pepper noise

becomes, the more the sharpness of the edges and the contours of the resultant image are reduced. The smaller the size of the Wiener window gives a better estimate of the pixel. The optimum window for the filter would be a 3×3 for a Wiener filter

and 5×5 for a median filter to have a high and steady PSNR, which also preserves the edges and contours. In the case of an image polluted with salt and pepper noise, the median filter plays an important role. The variations are very small and not very large, which is not going to be detrimental to diagnosis. The reconstructed image will have very good rating on the goodness scale of the psycho visual system. The effect of estimation can be measured using the peak signal-to-noise ratio (PSNR) for the various enhanced images. The PSNR is calculated in dB as $10^* \log_{10}(255^2/\text{mse})$ and has been tabulated for the two types of noise additives and salt and pepper noise.

The typical performance of the double-filtering technique for enhancing medical images on a RISC and DSP architecture is shown in Table 38.3 for a single block. The performance on the RISC machine is almost half that of the DSP, basically because it can multiply and add a 32×32 in two cycles when compared to a Blackfin, which consumes four cycles. The ARM9E can perform and store multiple loads in comparison to a single load on Blackfin. The register set on the RISC is also larger than the DSP.

Double filtering was tested for Gaussian noise. Each test image was polluted with Gaussian noise with a mean of zero and variances of 0.01, 0.02, and 0.03. Adaptive estimation with median filtering is performed for each sub-band, and an analysis is carried out by performing a study on the PSNR of the images under various conditions.

Table 38.4 shows the variations of Gaussian noise and the observations are made by choosing an optimal window. The optimum window for the filter would be 3×3 for Wiener and 5×5 for a median filter [14] as determined above. Table 38.4 shows the PSNR variations for the images polluted with Gaussian noise polluted with

Table 38.3 Cycle count of the technique

Blackfin-533	ARM 9E-S
577 Cycles	215 Cycles

Table 38.4 Tabulated results for the variations in PSNR for various reconstructed image slices for different variances of Gaussian noise

	PSNR Variations for variance of Gaussian noise (dB)		
S. No	Var 0.01	Var 0.02	Var 0.03
1	28.3935	25.938	24.7806
2	29.3172	27.413	26.1253
3	28.0491	26.476	25.028
4	28.5876	26.29	25.2445
5	28.2639	25.185	24.5483
6	27.5195	25.002	24.3159
7	29.9862	27.563	26.027
8	28.1811	27.482	26.5732
9	33.1041	27.482	26.5732
10	34.0267	30.647	28.4944

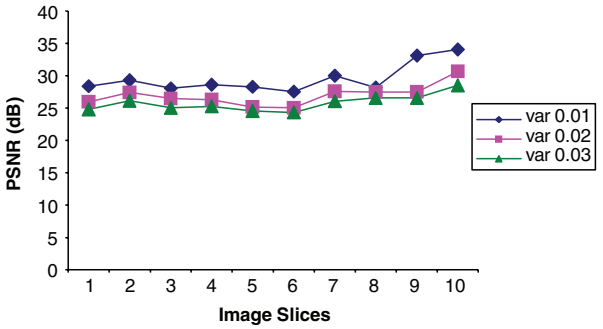
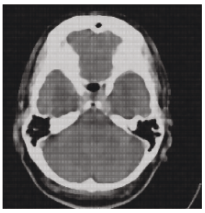


Fig. 38.8 Variations in PSNR for various reconstructed image slices for different variances of Gaussian noise

Fig. 38.9 Reconstructed image from Gaussian noise



different noise densities. The table shows the highest PSNR measured is 34 dB. Figure 38.8 shows the variation of the PSNR for various variance of Gaussian noise. The variations show that PSNR for the image filtered from noise with variance of 0.01 is high when compared to the PSNR of the filtered image polluted with higher variance. There is almost a variation of 4 dB between 0.01 and 0.03. The variation in the PSNR is also dependent upon the contours of the image. The image with greater contours has a PSNR that varies between 25–28 dB, whereas the image with lesser contours has a PSNR as high as 34 dB.

The graphical variations show a lateral shift with respect to the variance of the noise and the variations are constant with a higher PSNR for image slices with minimal contours. The image in Fig. 38.9 shows the enhanced image after the double filter has been applied to eliminate the Gaussian noise. The resultant image has the edges preserved even after double-filtering enhancement technique has been applied. Most of the diagnosis depends on the psycho visual system of the consultant.

The double-filtering technique was compared with the LMS algorithm. The experimentation for LMS algorithms involved an image obtained through magnetic resonance imaging (MRI) for analysis, and the image was polluted with additive white Gaussian noise with a mean of zero and 0.1, 0.01, and 0.02 variances. All the analysis is carried out in the wavelet domain. Table 38.5 shows variations in PSNR for different variances and iterations. It is generally observed that for larger number of iterations and a smaller step size, the PSNR is larger. The PSNR variations can also be seen graphically in Fig. 38.13. It can be noticed that for the larger number of iterations, the PSNR variations almost become constant. The variation is limited to 29–30 dB above 45 iterations for the chosen step size. This is an indication that

Table 38.5 PSNR variations for different iterations and constant step size

S. No	Iterations	PSNR variations in dB		
		Var 0.1	Var 0.01	Var 0.02
1	10	26.1342	34.6341	32.8325
2	15	26.9238	35.5591	33.68
3	20	27.4958	36.2346	34.3661
4	25	28.1154	36.564	34.9336
5	30	29.1703	37.3555	35.1349
6	35	28.937	38.013	35.3871
7	40	29.2472	37.9213	35.3042
8	45	29.8863	38.4695	36.2639
9	50	30.474	38.776	36.4236
10	60	31.1085	39.2478	37.0539

Fig. 38.10 MRI image polluted with Gaussian noise

Fig. 38.11 Block of image reconstructed with lesser number of iterations

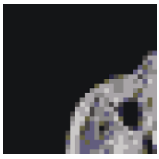


Fig. 38.12 Block of the image reconstructed after 60 iterations

the number of iterations for an embedded system can be limited, which reduces the system memory requirements, in turn. The implementation of the LMS algorithm in hardware or software for a pipeline processor will not be difficult. The parallelism can be exploited by accessing the inputs for the algorithm from simultaneous memory locations followed by arithmetic operations. The structure that follows similar computational methodology is a transversal structure, which reduces the complexity.

Figure 38.10 shows the resultant image after it has been polluted with Gaussian noise. Figure 38.11 shows a block of the image reconstructed with lesser number of iterations. The same block if image with a higher PSNR is seen in Fig. 38.12 when the number of iterations for the LMS algorithm to converge is made higher and by adapting a smaller step size. An embedded solution for using the LMS adaptive filter

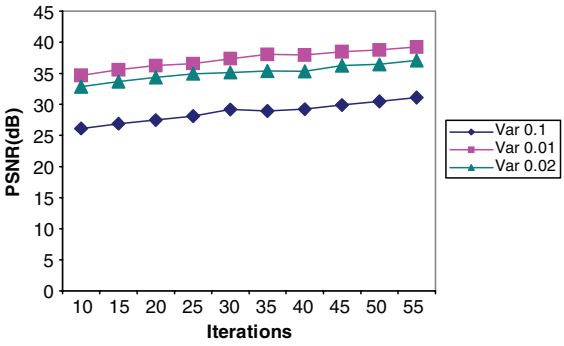


Fig. 38.13 Variations in the PSNR for different iterations and constant step size

Table 38.6 Cycle count of the enhancement techniques on Blackfin

Double filtering	Adaptive Filtering using LMS Algorithm
2503 Cycles	6125 Cycles

would be to choose smaller blocks of the image and with larger iterations and moderately chosen step size for faster processing and minimum storage requirements.

Table 38.6 shows the computational cycles involved on a Blackfin DSP processor. The Blackfin is a 16-bit, fixed-point DSP processor that supports only 16×16 multiplications. The LMS algorithm has floating point operations that have to be converted to the Q.15 format, and then multiplication has to be performed. The step size and error values e(n) can be fractional. A floating-point precision cannot be expected and the operation is carried out on every individual pixel. The computation at the pixel level is not expensive, but when the operation is performed on the block of the image, it becomes expensive for a 5×5 block of image as can be seen in Table 38.6. The double-filtering technique does not employ any floating-point operations, so the issue of accuracy does not arise, and manipulations are done using the pixel values and at the block level. The computational overhead for a 5×5 block of image is almost one-third the cycle count of the LMS algorithm. The cycle count was estimated only for the computations involved, and the corresponding load of data and the overhead of stacks and top-layer applications were not considered.

38.6 Conclusion

This chapter has shown some experimentation on a two-level filtering method for effective estimation and smoothing in the wavelet domain. Here, the images were tested using Haar wavelets. They can be tested for other types of wavelets and the results should prove encouraging. A formulated Wiener filter exploiting the correlation between the coefficient and noise can be removed from the poorly correlated

coefficient. The features of the image are not much distorted after smoothing in the spatial domain. Wiener filtering, along with median filtering, can be applied very effectively to avoid any loss of information during transmission. The double-filtering technique is compared with the LMS algorithm and shows that the optimum software implementation is less cycle-expensive on a DSP processor than the LMS algorithm. The ease of implementation of the suitable hardware can easily make the process faster and simpler when applied in a very big distributed environment. The hardware can be embedded in any major telemedicine facility with additional features, which tends to increase the efficiency of the system.

References

1. Resnikoff HL andWells RO (1998) Wavelet Analysis. Springer
2. Donoho DL (1993) Denoising via soft thresholding. IEEE Trans. Info. Theory 43: 933–936
3. Muller P and Vidakovic B (1999) Bayesian Inference in Wavelet Based Models Chapter 18, Lecture Notes in Statistics, vol 141 Springer-Verlag, New York, pp 291–308
4. Jain AK (1989) Fundamentals of digital image processing. Prentice-Hall. USA
5. Strela V (2000) Denoising via block wiener filtering in the wavelet domain. 3rd European Congress of Mathematics, Barcelona. Verlag, Birkhauser
6. Donoho DL and Johnstone IM (1995) Adapting to unknown smoothness via wavelet shrinkage. Journal of American Statistical Association, vol 90, no 432 pp 1200–1224
7. Proakis JG and Monalakis DG (2001) Digital Signal Processing. 3rd Ed
8. Gonzalez RC and Woods RE (2002) Digital Image Processing. 2nd Ed
9. Pratt WK (2001) Digital Image Processing. 3rd Ed
10. Haykins S (2002) Adaptive Filter Theory. 4th Ed
11. Hooli K (2001) Adaptive filters/LMS algorithm. University of Oulu.
12. Treicher JR, Johnson CR and Larimore MJ (2001) Theory and design of adaptive filters. Eastern Economy Edition. Wiley-Interscience. USA
13. Prandoni P and Vitterli M (1998) An FIR cascade structure for Adaptive Linear Prediction. IEEE transactions on signal processing. vol 46, no 9
14. Raj BP and Ramachandran V (2006) Performance analysis of adaptive estimation and non-linear filtering on the enhancement of medical Images. International Conference on Intelligent Systems and Control, India. August 2006

Chapter 39
A New Linear Appearance-based Method in Face Recognition

M. Hajiarbabi, J. Askari, S. Sadri, and M. Saraee

39.1 Introduction

Human identification recognition has attracted scientists for many years. During these years, and due to increases in terrorism, the need for such systems has increased much more. The most important biometric systems that have been used during these years are fingerprint recognition, speech recognition, iris, retina, and hand geometry, and face recognition. For comparing biometric systems, four features have been considered: intrusiveness, accuracy, cost, and effort. The investigation has shown that among the other biometric systems, face recognition is the best one [1].

A face recognition system has three parts—face localization, feature extraction, and classification. In face localization, part of the background and other parts of the image that may influence the recognition process is removed from the image. For this reason, the face is found in the image and the system just works on this part of the image. For simplicity, we ignore this part of the system. In the feature extraction part, the unique patterns of the face will be extracted from the image, and in the classification part these patterns will be placed in the class in which they belong. Each class shows a person's identity. The process of extracting the most discriminating features is very important in every face recognition system. In this chapter, we introduce fractional multiple exemplar discriminant analysis, which is a variation of a linear discriminant analysis algorithm. The results show that the proposed method, combined with RBF neural networks, has better results in comparison to other methods.

The rest of the chapter is organized as follows: In Sect. 39.2, linear discriminant analysis is reviewed. In Sect. 39.3, fractional multiple exemplar discriminant analysis is introduced. In Sect. 39.4, the RBF classifier is introduced, and finally in Sect. 39.5 the algorithm results are compared with other algorithms on UMIST [2] and ORL [3] databases.

M. Hajiarbabi, J. Askari, S. Sadri, and M. Saraee
Electrical and Computer Engineering Department, Isfahan University of Technology, Iran
e-mail: m_arbabi@ec.iut.ac.ir, j-askari@cc.iut.ac.ir, sadri@cc.iut.ac.ir

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

39.2 Linear Discriminant Analysis

LDA is used for projecting a set of training data. In face recognition, these training data are the images that belong to a person. In face space, which is a $m \times n$ dimension (and m,n are the image dimensions), consider that $X = (X_1, X_2, \ldots, X_n) \subset \Re^{m \times n}$ is a matrix containing the images in the training set. X_i is an image that has converted to a column vector. In LDA, two matrixes within a class scatter matrix and between class scatter matrixes, are defined. This method finds an optimal subspace in which the between class scatter matrix to the within class scatter matrix will be maximized [4]. The between class scatter matrix is computed by

$$S_B = \sum_{i=1}^{c} n^i \left(\overline{X}^i - \overline{X}\right)\left(\overline{X}^i - \overline{X}\right)^T \tag{39.1}$$

where $\overline{X} = \left(1/n\right) \sum_{j=1}^{n} X_j$ is the mean of the images in the training set and $\overline{X}^i = \left(1/n^i\right) \sum_{j=1}^{n^i} X_j^i$ is the mean of class i, and c is the number of the classes. In face recognition, each class is the total of images that belong to one person. The between class scatter matrix defines the average scattering of one class across the average of the total classes. The within class scatter matrix is computed by

$$S_W = \sum_{i=1}^{c} \sum_{X_i \in n^i} \left(X_i - \overline{X}^i\right)\left(X_i - \overline{X}^i\right)^T \tag{39.2}$$

The within class scatter matrix defines the data of one class across the average of the class. The optimal subspace is computed by

$$E_{optimal} = \arg\max_{E} \frac{\left\|E^T S_B E\right\|}{\left\|E^T S_W E\right\|} = [c_1, c_2, \ldots, c_{c-1}] \tag{39.3}$$

where $[c_1, c_2, \ldots, c_{c-1}]$ is the set of Eigen vectors of S_B and S_W corresponding to $c-1$ greatest generalized Eigen value λ_i and $i = 1, 2, \ldots, c-1$.

$$S_B E_i = \lambda_i S_W E_i \; i = 1, 2, \ldots, c-1 \tag{39.4}$$

$E_{optimal}$ is an optimal matrix that maximizes the proportion of the between class scatter matrix to the within class scatter matrix. This means that it maximizes the scattering of the data that belongs to different classes and minimizes the scattering of the data belonging to the same class.

Thus, the most discriminant answer for face images X would be [4]:

$$P = E_{optimal}^T \cdot X \tag{39.5}$$

One of the drawbacks that LDA encounters in face recognition is the small sample size problem. To avoid singularity of the S_w matrix, the number of the training images must be much more than the dimension of the subspace, a situation that

rarely occurs in face recognition problems. To avoid the singularity problem, we first have to reduce the dimension of the problem and then apply LDA. Principal component analysis (PCA) [5] is the most popular method that has been used for dimension reduction. In this chapter, we apply PCA on the images before every method that we mention. In addition to PCA, there are other effective methods that can be used for dimension reduction before LDA, such as discrete cosine transform (DCT) [6].

Some researchers have noticed that applying PCA to reducing the dimension of the space can cause another problem, and that's eliminating some useful information from the null space. Xiang [7] introduced 2FLD to avoid this problem, as well as the computational problems that applying PCA produces. But the 2FLD algorithm produces other problems that have not been mentioned here. The output of 2FLD method is a matrix and its dimension for an $m \times n$ image could be $n \times n$. This high dimension will make trouble when we want to use a neural network for classification. A two-dimensional (2D) matrix can not be applied to a neural network. If we change the matrix into a vector, we will have a vector with size n^2, and because of low sample tests of each face, the network cannot be trained well. Yu [8] proposed a direct LDA method that did not need to use the PCA method before applying LDA, but this method is time inefficient.

39.3 Fractional Multiple Exemplar Discriminant Analysis

The problem of face recognition differs from other pattern recognition problems, and therefore needs different discriminant methods rather than LDA. In LDA, the classification of each class is based on just one sample, and that's the mean of each class. Because of a shortage of samples in face recognition problems, it is better to use all the samples instead of the mean of each class for classification. Rather than minimizing the within class distance while maximizing the between class distance, multiple exemplar discriminant analysis (MEDA) finds the projection directions along which the within class exemplar distance (i.e., the distances between exemplars belonging to the same class) is minimized while the between class exemplar distance (i.e., the distances between exemplars belonging to different classes) is maximized [9].

In MEDA, the within class scatter matrix is computed by

$$S_W = \sum_{i=1}^{C} \frac{1}{n^{i^2}} \sum_{j=1}^{n^i} \sum_{k=1}^{n^i} \left(X_j^i - X_k^i\right) \left(X_j^i - X_k^i\right)^T \tag{39.6}$$

where X_j^i is the jth image of ith class. By comparing it with the within class scatter matrix of LDA, we see that in this method all the images in a class have participated in making the within class scatter matrix instead of using just the mean of the class as in the LDA method. The between class scatter matrix is computed by

Table 39.1 Fractional algorithm

Set $W = I_{n\times n}$ (the identity matrix)
for $k = n$ to $(m+1)$ step (-1)
    for $\ell = 0$ to $(r-1)$ to step 1
        Project the data using W as $y = W^T x$
    Apply the scaling transformation (defined below) to obtain $z = \varphi\left(y, \alpha^{\ell}\right)$
    For the z patterns, compute the $k \times k$ between class scatter matrix S_b
    Compute the ordered Eigen values $\lambda_1, \lambda_2, \ldots, \lambda_k$ and corresponding Eigen vectors $\phi_1, \phi_2, \ldots \phi_k$ of S_b
Set $W = W\tilde{\Phi}$, where $\tilde{\Phi} = [\phi_1, \phi_2, \ldots, \phi_k]$
    end for
    Discard the last (kth) column of W.
end for

$$S_B = \sum_{i=1}^{C} \sum_{j=1, j\neq i}^{C} \frac{1}{n^i n^j} \sum_{k=1}^{n^i} \sum_{l=1}^{n^j} \left(X_k^i - X_l^j\right)\left(X_k^i - X_l^j\right)^T \tag{39.7}$$

Contrary to LDA—in which the means of each class and means of all samples make the between class scatter matrix—in MEDA, all the samples in one class will be compared to all samples of the other class. The computation of $E_{optimal}$ is the same as LDA.

There is a drawback, however, that is common in both LDA and MEDA. In the between class scatter matrix (S_B) there will be no difference if the samples are closer or far from each other. It is clear, however, that for the classes that are closer to each other, the probability of collision is more than in the other classes.

To better discriminate the vectors, we used the idea in Lotlikar and Kothari [10]. The idea was used for LDA and has not been experienced for face recognition, and also has not yet been applied to MEDA. This algorithm suggests reducing the dimension of the problem step by step, and in each iteration the samples that are closer are forced farther apart. For this purpose, a weight function has been introduced

$$w\left(d_{X_1X_2}\right) = \left(d_{X_1X_2}\right)^{-p},\ p = 3,4,\ldots \tag{39.8}$$

where $d_{X_1X_2}$ was introduced as the distance of the center of each class from each other. In this chapter, however, we used it as the distance of two samples. The between class scatter matrix in fractional MEDA is:

$$S_B = \sum_{i=1}^{C} \sum_{j=1, j\neq i} \frac{1}{n^i n^j} \sum_{k=1}^{n^i} \sum_{l=1}^{n^i} w\left(d_{X_k^i X_l^j}\right) \times \left(X_k^i - X_l^j\right)\left(X_k^i - X_l^j\right)^T \tag{39.9}$$

The within class scatter matrix is the same as MEDA. The fractional algorithm is shown in Table 39.1 [10]. In the code, r is the number of fractional steps used to reduce the dimensionality by one.

Table 39.2 FMEDA algorithm

1. Applying PCA on the training set.
2. Computing within class scatter matrix using Eq. 6.
3. Computing between class scatter matrix using Eq. 9.
4. Applying fractional step dimensionally reduction algorithm.
5. Computing optimal subspace using Eq. 3.
6. Computing most discriminant vectors using Eq. 5.

The scaling transformation compresses the last component of y by a factor α^{ℓ} with $\alpha < 1$, *i.e*, $\Psi\left(y;\alpha^{\ell}\right) : y \in \Re^k \rightarrow z \in \Re^k$ such that

$$z_i = \begin{cases} \alpha^{\ell} y_i, & i = k \\ y_i, & i = 1,2,\ldots,(k-1) \end{cases} \tag{39.10}$$

Some comments about this algorithm are as follows:

- In the *rth* step, the reduction factor is α^{r-1}. It means that a dimension is removed by $1, \alpha,\ \alpha^2, \ldots \alpha^{r-1}$ scales.
- When the number of steps is smaller, α should be chosen larger and vice versa.
- Weighting functions should be chosen d^{-3}, d^{-4}, and so on [10].

The FMEDA algorithm is shown in Table 39.2.

39.4 Radial Basis Function Neural Network

For classification, we use a distance measure and also RBF neural network to compare their classification power. As a distance measure, we select Euclidean distance. RBF neural network is a powerful classification method for pattern recognition problems. It doesn't have the drawbacks of multi-layer perceptron neural networks and trains much faster. Figure 39.1 shows an RBF neural network.

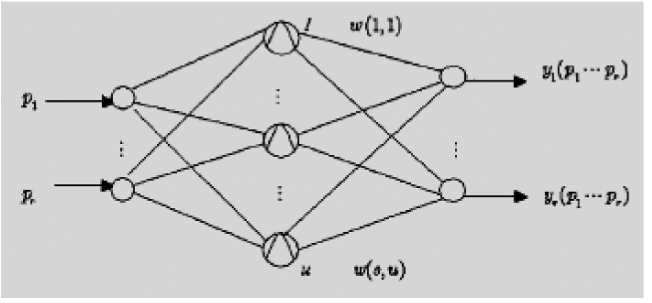


Fig. 39.1 RBF neural network

Let $P \in \Re^r$ be the input vector and $C_i \in \Re^r$ $(1 \leq i \leq u)$ be the prototype of the input vectors. The output of each RBF unit is as follows:

$$R_i(P) = \exp \frac{-\|P - C_i\|^2}{\sigma_i^2} \tag{39.11}$$

where σ_i is the width of the ith RBF unit. The jth output $y_j(P)$ of an RBF neural network is

$$y_i(p) = \sum_{i=1}^{u} R_i(P) * w(j,i) + w(j,0) \tag{39.12}$$

where $R_0 = 1$, $w(j,i)$ is the weight of the ith receptive field to the jth output. The weights of the first layer are all equal to one. The number of nodes in the second layer at first equals the number of classes. Whenever two classes have an intersection with each other, a node is added to the second layer and a class is split into two subclasses. For further knowledge about the RBF neural network, the reader can refer to neural network references.

39.5 Implementation and Results

To test the algorithms mentioned above, we used the UMIST [2] and ORL [3] databases, which are free. The UMIST database contains 575 images that belong to 20 people with a variety of head poses from front view to profile. Ten images from each person were used as a training set, and the rest were used as test sets. Figure 39.2 shows a sample of this database.

The ORL database contains 400 images that belong to 40 people with variety in scale and pose of the head. Five images from every person were used as a training set and the rest were used as test sets. Figure 39.3 shows a sample of this database.

For FMEDA, we used p = 4 because it showed better results than other p's, and for a fractional step, we chose r = 35. The results are shown in Figs. 39.4 through 39.7. The output of the RBF neural network has 40 nodes for the ORL database and 20 for the UMIST.

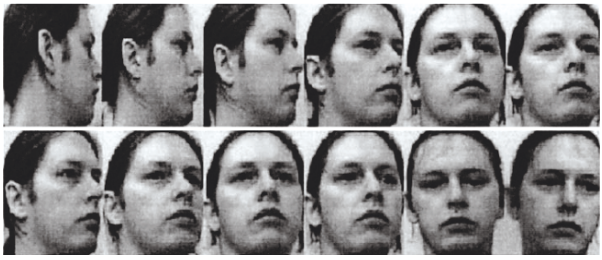


Fig. 39.2 UMIST database

Fig. 39.3 ORL database

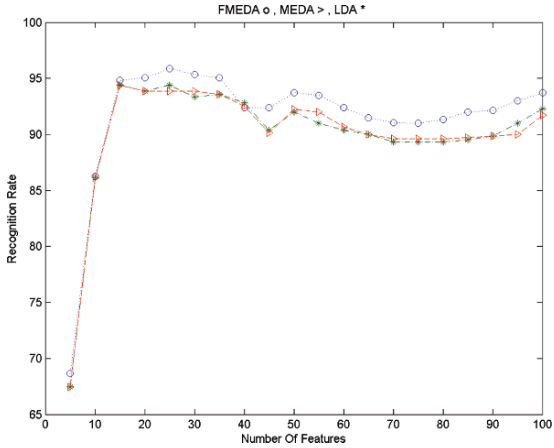


Fig. 39.4 Results for the UMIST database using Euclidean distance

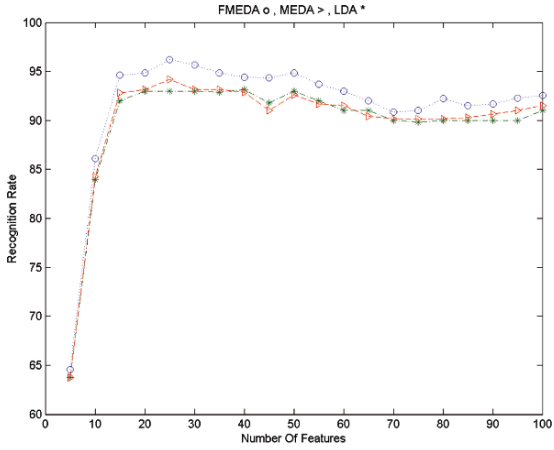


Fig. 39.5 Results for the UMIST database using RBF

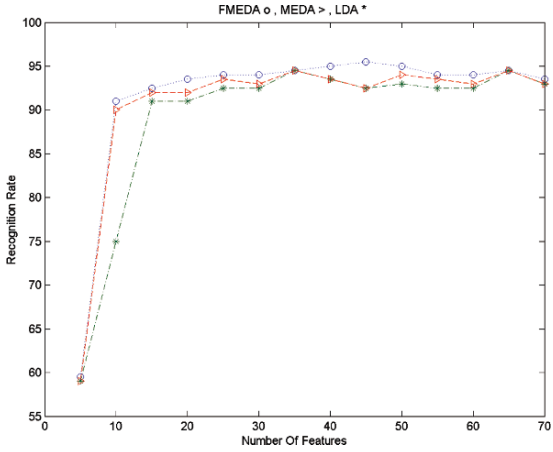


Fig. 39.6 Results for the ORL database using Euclidean distance

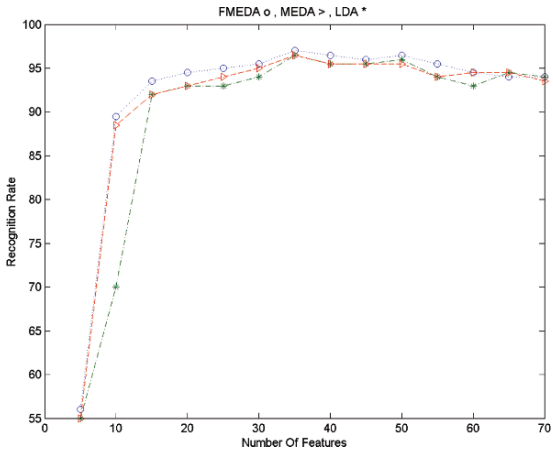


Fig. 39.7 Results for the ORL database using RBF

39.6 Conclusion

In this chapter, we introduced fractional multiple exemplar discriminant analysis, which was a variation of LDA algorithm. We used UMIST and ORL databases to evaluate the performance of FMEDA algorithms, and Euclidean distance and RBF neural networks as classifiers. The results showed that FMEDA extracts more discriminant features than MEDA and LDA, and also showed that the RBF classifier is much stronger than distance-based classifiers such as Euclidean distance.

References

1. Hietmeyer R (2000) Biometric identification promises fast and secure processing of airline passengers. The Int'l Civil Aviation Organization Journal. vol 55. no 9. p 10–11
2. UMIST database: http://images.ee.umist.ac.uk/danny/dadabase.html
3. ORL database: http://www.camorl.co.uk
4. Fukunaga K (1990) Introduction to statistical pattern recognition, 2nd ed. Academic Press, San Diego p 445–450
5. Turk M and Pentland A (1991) Eigenfaces for recognition. J. Cognitiv Neurosci. (3):71–86
6. Hajiarbabi M, Askari J, Sadri S and Saraee M (2007) Face Recognition Using Discrete Cosine Transform plus Linear Discriminant Analysis. Accepted in WCE
7. Xiong H, Swamy M, and Ahmad M (2005) Two-dimensional FLD for face recognition. Elsevier Pattern Recognition, vol 38. p 1121–1124
8. Yu H and Yang J (2001) A direct LDA algorithm for high dimensional data with application to face recognition. Pattern Recognition, vol 34. p 2067–2070
9. Zhou SK and Chellappa R (2003) Multiple-Exemplar Discriminant Analysis for Face Recognition. Center for Automation Research and Department of Electrical and Computer Engineering University of Maryland, College Park, MD
10. Lotlikar R and Kothari R (2000) Fractional-step dimensionality reduction. IEEE Trans. Pattern Anal. Machine Intell., vol 22. p 623–627

Chapter 40
A Meta-search Engine with Hits Redundancies Filtering

Suet Peng Yong and Wan Nordiana Wan Abd Kadir

40.1 Introduction

Searching information on the Web can be viewed as extremely easy, or it can be extremely difficult. This is because the WWW is not indexed like many library catalogs or journal-article indexes. When we search on the Web, we are not searching it directly but are actually searching the Web pages collected and indexed by a search tool from computers all over the world that contains the actual Web pages. Examples of search tools are Yahoo! Search, Google, Alta Vista, and so on.

The different types of search tools have their own strengths and weaknesses. Depending on your information needs, one may work better for you than another. Search directories are hierarchical databases with references to websites. The websites that are included are classified according to the rules of that particular search service. Search engines, on the other hand, use software to *crawl* the Internet in search of what you would like using of terms or keywords. Specialized databases are the hidden parts of the WWW that are normally not found by regular search engines [1].

This paper presents the extension work from Integrated Filtered Web-Search Engine (IFWSE) [2] by looking at the enhancement of the program structure to retrieve the search hits. Further experiments were also carried out to compare its performance with with a well-established, meta-search engine—the MetaCrawler [3] [4].

40.2 Related Work

Meta-search engines send the search terms to the databases maintained by search engine companies. *Smarter* meta-search technology includes clustering and linguistic analysis that attempts to show the themes within results, and some fancy textual

Suet Peng Yong and Wan Nordiana Wan Abd Kadir
Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia
e-mail: yongsuetpeng@petronas.com.my

X. Huang et al. (eds.), *Advances in Communication Systems and Electrical Engineering.*

analysis and display that can help with looking deeply into a set of results. However, neither of these technologies is any better than the quality of the search engine databases from which they obtained the results [5].

There are some intelligent search agents in the market that act similarly as meta-search where it can search several search engines simultaneously. Examples of such search agents include Copernic Agents by Copernic Technologies [6] and MetaCrawler developed by the University of Washington in 1994 and operated by InfoSpace after 2000 [3] [4]. Copernic features the ability of a search wizard to search using a question or keywords, keyword highlighting in results and Web pages, a detailed search history, automatic software updating, and many useful search management functions [3]. By combining robustness and scalability, its technology retrieves and indexes data wherever it is found: on corporate intranets, company servers, and public websites. It makes use of advanced language and linguistic analysis technologies, resulting in unparalleled indexing precision [3]. However, it is an agent that needs to be installed in a Web browser.

MetaCrawler works by querying a number of existing, free search engines, organizing the results into a uniform format, and displaying them. A so-called fast search method produces results the quickest. After a few seconds, this search method will bring up a new page filled with links that link to information related to the keywords as *hits*. The MetaCrawler operates in two general modes: *normal mode* and *verification mode*. In normal mode, the MetaCrawler reports results immediately after retrieval from remote search engines. In verification mode, the MetaCrawler loads and verifies each reference to ensure the validity of the data [5]. However, it was found that the retrieved results are usually around 100 hits.

40.3 Project Work

The prototype system discussed in this chapter (IFWSE) has a similar aim as MetaCrawler, which is to use a single, central interface for WWW document searching. However, it is designed without using any internal database and works by hitting the selected remote search engines when requested by a user so that the results obtained are the latest update from the related search engines.

Figure 40.1 illustrates the components in the architecture of the system and Fig. 40.2 illustrates the flow of the processes involved. In a typical session using IFWSE, a user submits a query to the IFWSE through a user-friendly interface. The IFWSE then sends the user query to a number of remote search engines based on user preferences. Different remote search engines may accept queries in different formats. The user query may thus need to be translated to an appropriate format for each remote system. The search engine library stores the format of remote search engines invocation and used by *query core* components to do the translation process. Hence, the number of remote search engine systems can be expanded by extending the search engine library.

In this prototype, only the top 50 results (but, of course, not limited to that) from each of the search engines will be retrieved. After the retrieved results from

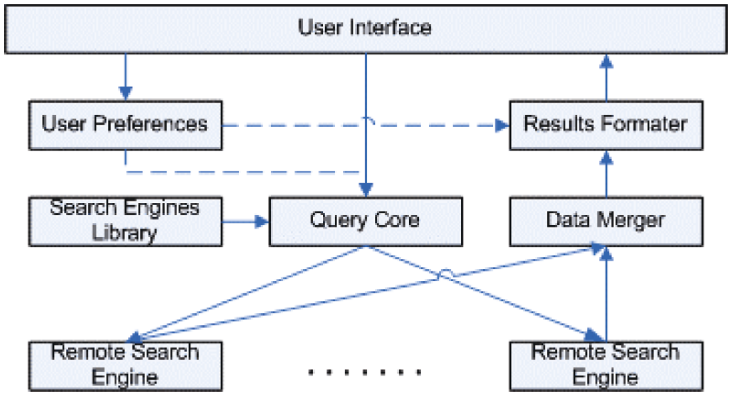


Fig. 40.1 System components

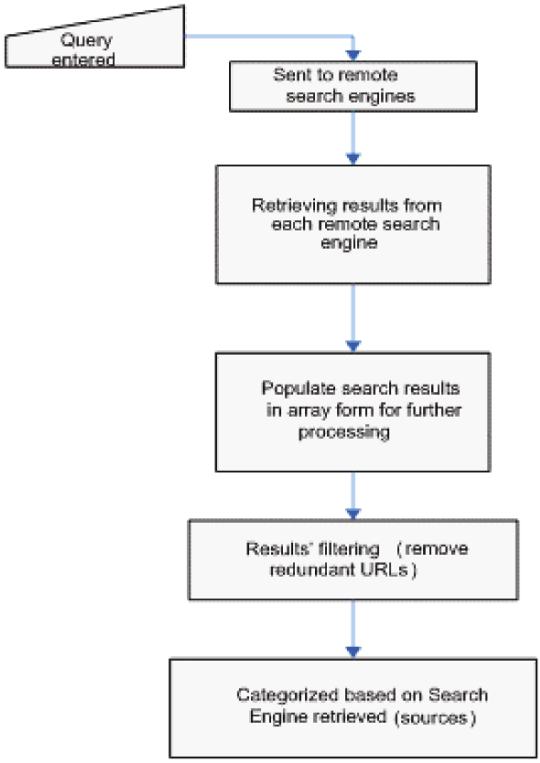


Fig. 40.2 System process flow diagram

the remote search engines are received, the system shall filter the results for any irrelevant and duplicate links. The final process is to rank the results according to the user preferences or according to the most relevant hits according to the user's interests. The results will then be displayed and presented to the user.

40.3.1 Hits Redundancies Filtering Processes

The filtering process is done after the array of the results have been populated and arranged in the form where it is easier for filtering and sorting processes. The results from each of the search engines are being filtered step by step using one particular search engine among the platform of search engines as the base search results, and then comparing the URLs to check for any redundancies. For the start, the base search engine (S1) results will be compared with the second search engine (S2) results, and the comparison for S1 and the third search engine (S3) will follow afterwards.

The process will be recursive while looping until the maximum number of the total results is retrieved. If a similar URL is found while doing the comparing, the loop will break and continue to the next results. One of the results that have similar URLs will be omitted. For example, if the URL for S1 result and S2 result were identical, the particular S2 result will be omitted so that later on it will not be displayed. Otherwise, the result retrieved from S2 will be added to the final result list. A similar process will be carried out on S2 and S3, in which the S2 list will be added up by the results from S3 that were not identical to S2. The algorithm can be illustrated as in Fig. 40.3.

40.3.2 Categorizing and Displaying Results

Results from the filtering process will be the nonredundant hits. For display purposes, the filtered hits will be populated in a new array in a form that can be

```
Call the Query Function
Query Function invoke selected remote search engines
based on user preferences
Retrieve the results
Converting results from strings to arrays
Filtering and Populating Processes:
      For each S1 compare to each S2
            If found same URL then
                  Mark current S1 ∩ S2 hits
   (Process will be between S1 ∩ S2, S1 ∩ S3, and S2
   ∩ S3 and also based on user preferences)
Final Results = (S1 ∩ S2 ∩ S3) + (S1 ∩ S2) + (S1 ∩
S3) + (S2 ∩ S3) + (S1 - S2 - S3) + (S2 - S1 - S3) +
(S3 - S1 - S2)
Output is displayed based on the categorized
filtered hits
Display the results
```

Fig. 40.3 Pseudocode for major retrieval, parsing and filtering process

categorized by the source of hits retrieved. Let say that if the hits were retrieved from S1 ∩ S2 ∩ S3, they will be displayed on the top list followed by the hits group retrieved from S1 ∩ S2, S1 ∩ S3, and S2 ∩ S3, respectively. Te remaining will be hits that appeared in S1 only, S2 only, and—last but not least—S3 only.

40.4 Results and Discussion

40.4.1 Test on Redundancy

Experiments were conducted by feeding 50 keywords and phrases to S1, S2, S3, and IFWSE, respectively. The key words were randomly picked, and the same key words were keyed in 10 times in 10 different days. This was to see the behavior of IFWSE as it retrieved the results in real time from the platform of various search engines. The mean hits achieved after performing a number of experiments with IFWSE to acquire the total results retrieved from each search engine were recorded to compare with the filtered hits from IFWSE. Table 40.1 shows 25 of the 50 key words' results retrieved as compared with the results from remote search engines.

IFWSE, for the time being, limits the hits retrieved from each search engine to only the search hits by IFWSE is reduced by about 25 percent.

Let's take the first searched key word 'Petronas' as an example. The total hits produced from the three search engines were 152, but IFWSE produced only 116 hits. In detail, the overlapped hits for all three search engines (S1 ∩ S2 ∩ S3) were 10, S1 ∩ S2 were 9, S1 ∩ S3 were 8, S2 ∩ S3 got only one overlapped, and the rest of the results were unique and nonredundant. From the observation, results that overlapped were normally high-ranked results that were retrieved from the top 10 and 20 hits of every search engine. The remaining hits seldom had overlapped results.

Table 40.1 Comparison of number of hits retrieved

KEYWORDS SEARCHED / RETRIEVE FROM	TOTAL HITS RETRIEVED																								
	Petronas	Data Mining	Information System	E-Learning Technology	Open Source Software	Nano Technology	University Technology Petronas	Ontology Based Application	Oil and Gas Research	Meta Search Engine Application	Computer	Malaysia	Algorithm and Data Structure	Relational Database Management System	International Conference on Information and Technology	Problem Solving Agent	Knowledge and Reasoning	Mobile Content Challenge	Natural Language Processing	Multi Agent System	Wireless Innovation Contest	measuring search effectiveness	Internet search strategies	Study and analysis of user queries on the web
Google	50	50	50	51	50	50	50	51	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Yahoo!	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Windows Live	52	50	50	50	51	50	50	50	51	50	51	50	50	50	50	50	50	50	51	50	51	50	50	50	50
Total Results	152	150	150	151	151	150	150	151	151	150	151	150	150	150	150	150	150	150	151	150	151	150	150	150	150
IFWSE Results	114	102	138	136	125	128	133	132	128	137	132	122	137	123	125	141	136	132	110	133	131	124	120	135	

Obviously, as shown in Table 40.1, IFWSE could filter the redundant hits out of each of the search engines. This filtering process allows each of the hits that have identical URLs to be displayed only once to the user. With reference to Table 40.1, it can be seen that for only the top 50 hits of the search engines, 75 percent of the hits retrieved are unique. Indirectly, IFWSE helps to save reading similar hits about 25 percent of the time. Search engines' hits overlap far less than we would have expected. That's the reason why users constantly have the habit of opening more than one browser for another search engine.

The objectives of IFWSE include filtering hits to eliminate any redundancy, and giving users more top-ranked hits taken from several well-known search engines. The first mentioned objective was evaluated by using precision theory where it stresses how high the precision of the filtering process was. The second objective was evaluated by the comparison of total hits returned by IFWSE, and the total hits returned by a well-established integrated search engine—in this case, the MetaCrawler. MetaCrawler is a free integrated search tool that mainly consists of Google, Yahoo! Search, MSN Search, and Ask.com.

40.4.2 Precision

Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is basically used to show the percentage of desired hits and those successfully retrieved. It is difficult to measure recall because the WWW is such a huge warehouse and we don't know how many relevant hits there are in it [7] [8]. Hence, precision was used in this case for the retrieval performance measurement by looking at the redundant hits. In this project, the precision formula was as follows:

$$\text{Precision} = (\text{A}/\text{B}) \times 100\% \tag{40.1}$$

where

$$\text{A} = \text{total hits returned} - \text{number of redundant hits}$$
$$\text{B} = \text{total hits returned}$$

In this case, redundant hits mean repeated hits that IFWSE failed to filter. Figure 40.4 shows the preciseness of the IFWSE's filtering performance as compared to MetaCrawler. A number of redundant hits were obtained from a very close inspection during the experiments. The precision was calculated for each of the search key words. Basically, IFWSE can approximately filter hits with an average precision rate of 99.5 percent (compare to MetaCrawler with its 99.85 percent average precision rate). The redundant trends are actually those results from $\text{S2} \cap \text{S3}$. It is believed that the performance of the filtering process for the second round becomes less effective due to the structure of the algorithm.

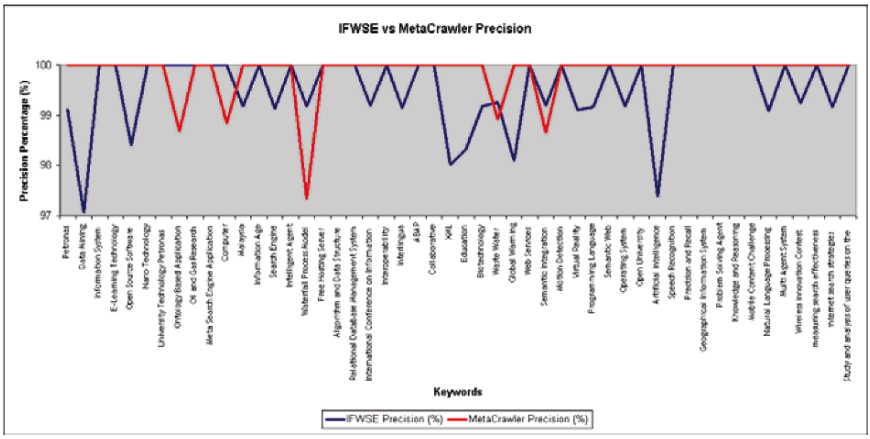


Fig. 40.4 Precision evaluation

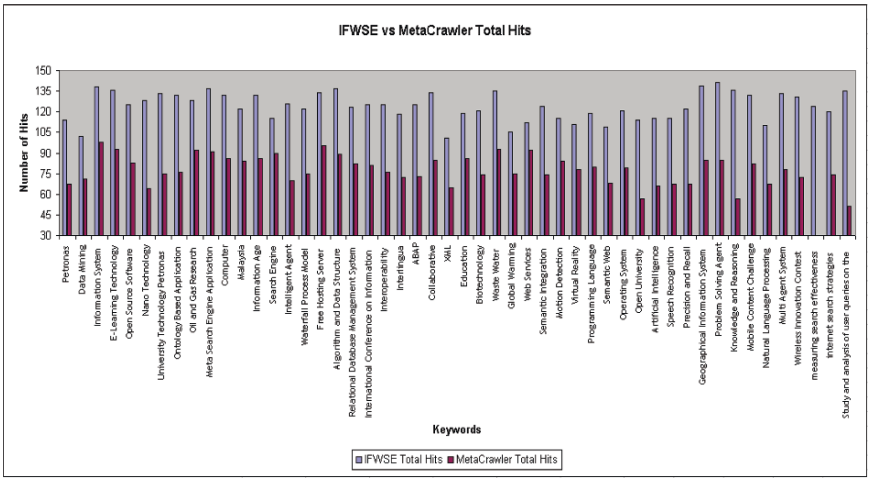


Fig. 40.5 Comparison of total results retrieved from IFWSE and MetaCrawler

40.4.3 IFWSE vs MetaCrawler

The comparison of total hits based on the previous key words retrieved by IFWSE (after omitting the duplicate hits) and MetaCrawler was conducted and shown in Fig. 40.5.

It is worth noting here that the number of hits retrieved by MetaCrawler is smaller than IFWSE, although it integrates four search engines whereas IFWSE uses only three. The rationale behind this is that there were more overlapping hits retrieved by MetaCrawler in comparison to IFWSE. Why did this happen? Based on close observation, MetaCrawler retrieved up to the top 30 hits of the results in each search engine, whereas IFWSE retrieved up to top 50 hits from each search engine. On top

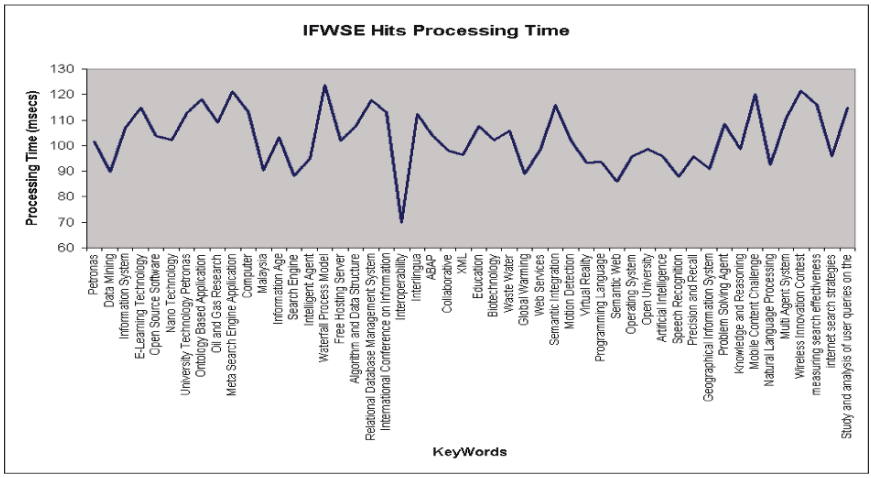


Fig. 40.6 IFWSE hits processing time

of that, most of the overlapped results were highly-ranked. The lower the ranking, the more unique the hits returned are.

Hence, the hits retrieved from MetaCrawler were mostly the top 30 hits of each search engine, and most of the time these results are overlapped or redundant. On the other hand, IFWSE retrieved the top 50 hits and was able to get less overlapped results. Basically, IFWSE gives the advantage of variety hits returned to the users.

40.4.4 IFWSE Processing Time

Implementation of the filtering algorithm drawn in Fig. 40.3 takes around 100 milliseconds on the average for IFWSE to process the hits retrieved from all the search engines. Figure 40.6 shows the average processing time of each searched key word. Note that the processing time is the duration needed for IFWSE to process the retrieved results (from the search engines), filter out the redundancies, and display to the user.

40.5 Conclusion

Database selection, document selection, and result-merging that required more knowledge on the databases of component search engines were pointed out as the major problems of meta-search engines [9]. IFWSE is designed without using any internal database and works by integrating the selected remote search engines upon user request. Hence, the results obtained are the latest updates from the component

search engines. Because it does not host any database, it reduces the problems of database maintenance and integration. However, it was found that the processing time was a little short in comparison to the meta-search engine that hosts an internal database. Nevertheless, this time constraint does not pose a significant disadvantage to IFWSE because IFWSE is able to provide more unique results. The real implication of time constraints can be further looked into and that will lead to some future research by further enhancing the program structure.

References

1. Chris S (2005) Internet Search Strategies: Search Tools. Minnesota West Home, Minnesota West Community and Technical College Available: http://www.mwctc.cc.mn.us/libraries/strategies/tools.htm
2. Yong SP and Wan Nordiana WAK (2007) Search Engines Integration with Hits Redundancy Filtering. In Proceedings of International Multi Conference of Engineers and Computer Scientists, Hong Kong, International Association of Engineers (IAENG)
3. Selberg E and Etzioni O (1997) The MetaCrawler Architecture for Resource Aggregation on the Web. IEEE Expert, 12(1): 8–14
4. Computer Science and Engineering (2001) MetaCrawler/Husky/Grouper Available: http://www.cs.washington.edu/research/projects/WebWarel/www/metacrawler/
5. Joe B (2005) Meta-Search Engines. University of California. Available: http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/MetaSearch.html
6. Copernic (2005) About us – Technologies, Indexing. Available: http://www.copernic.com/en/company/technologies.html
7. Jensen B, Spink A and Saracevic T (2000) Real life, real users and real needs: a study and analysis of user queries on the Web. Information Processing and Management. 36(2): 207–2267
8. Richard J (2000) Measuring Search Effectiveness. Available: Creighton University Health Sciences Library and Learning Resources Center
9. Meng W, Yu C and Liu KL (2002) Building Effective and Efficient Metasearch Engine, ACM Computing Surveys, vol 34 No 1, March p 48–89

Index