Multimedia

Multimedia

Edited by
Kazuki Nishi

In-Tech
intechweb.org

Published by In-Teh

In-Teh
Olajnica 19/2, 32000 Vukovar, Croatia

www.intechweb.org
Additional copies can be obtained from:
publication@intechweb.org

First published February 2010
Printed in India

Technical Editor: Zeljko Debeljuh
Cover designed by Dino Smrekar

Multimedia,
Edited by Kazuki Nishi

p. cm.
ISBN 978-953-7619-87-9

Preface

Multimedia technology will play a dominant role during the 21st century and beyond, continuously changing the world. It has been embedded in every electronic system: PC, TV, audio, mobile phone, internet application, medical electronics, traffic control, building management, financial trading, plant monitoring and other various man-machine interfaces. The usability or user-friendliness is depending on maturity of the multimedia technology. It improves the user satisfaction and the operational safety. Therefore, any electronic systems are no longer able to be realized without the multimedia technology.

The aim of the book is to present the State-of-the-Art research, development, and implementations of multimedia systems, technologies, and applications. Chapters 1 and 2 deal with the cross-correlation and cooperative work in the multimedia environment. Chapters 3 to 7 describe the understanding, recognition and retrieval of image or video data. Chapters 8 to 10 describe the automatic production of graphics or video. Chapters 11 and 12 present new testing systems for camera-shake, image stabilizers and surface inspection. Chapters 13 to 17 deal with wireless secure system, coding, data hiding or digital watermarking technologies of image, video or audio. Chapters 18 and 19 introduce a signal processing technique and a traffic model analysis, respectively. Chapters 20 to 23 deal with computer architecture, file and mail system.

All chapters will represent the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field. The editors would like to thank the authors who devoted so much effort to this publication.

Editor

Kazuki Nishi
UEC Tokyo (University of Electro-Communications), Japan

Contents

Preface V

1. On Cross-Media Correlation and Its Applications 001
   Wei-Ta Chu

2. An Intelligence Fault Tolerance Agent for Multimedia CSCW based on Situation-Awareness 025
   Eung-Nam Ko

3. Study on Data-driven Methods for Image and Video Understanding 037
   Tatsuya Yamazaki

4. Using the Flow of Story for Automatic Video Skimming 051
   Songhao Zhu and Yuncai Liu

5. Image Matching and Recognition Techniques for Mobile Multimedia Applications 077
   Suya You, Ulrich Neumann, Quan Wang and Jonathan Mooser

6. Structured Max Margin Learning on Image Annotation and Multimodal Image Retrieval 105
   Zhen Guo, Zhongfei (Mark) Zhang, Eric P. Xing and Christos Faloutsos

7. Invariant Image Retrieval: An Approach Based On Multiple Representations and Multiple Queries 117
   Noureddine Abbadeni

8. Software applications for visualization territory with Web3D-VRML and graphic libraries 139
   Eduardo Martínez Cámara, Emilio Jiménez Macías, Julio Blanco Fernández, Félix Sanz Adán, Mercedes Pérez de la Parte and Jacinto Santamaría

9. An Interactive Fire Animation on a Mobile Environment 165
   DongGyu Park, SangHyuk Woo, MiRiNa Jo and DoHoon Lee

10. Digital Camera Work for Soccer Video Production with Event Detection and Accurate Ball Tracking by Switching Search Method 173
    Author Name

11. Testing and Evaluation System for Camera Shake and Image Stabilizers (TEVRAIS) 191
Kazuki Nishi

12. Online surface inspection technology of cold rolled strips 205
Guifang Wu

13. Error Resilient Video Coding Techniques Based on the Minimization of End-to-End Distortions 233
Wen-Nung Lie and Zhi-Wei Gao

14. Bit Rate Estimation for Cost Function of H.264/AVC 257
Mohammed Golam Sarwer, Lai Man Po and Q.M. Jonathan Wu

15. Secure Multimedia Streaming over Multipath Wireless Ad-hoc Network: Design and Implementation 281
Binod Vaidya, Joel J. P. C. Rodrigues and Hyuk Lim

16. Complete Video Quality Preserving Data Hiding for Multimedia Indexing 303
KokSheik Wong and Kiyoshi Tanaka

17. Digital Watermarking Techniques for AVS Audio 329
Bai-Ying Lei, Kwok-Tung Lo and Jian Feng

18. Klaman Assisted LMS Methods 353
Nastooh Avesta

19. Delay Difference between the Linear Traffic Model and the Polynomial Traffic Model 375
Woo-Young Ahn, Seon-Ha Lee and Gyeong-Seok Kim

20. Collaborative Negotiation to Resolve Conflicts among Replicas in Data Grids 391
Ghalem Belalem and Belabbes Yagoubi

21. Performance Improvements of Peer-to-Peer File Sharing 411
Dongyu Qiu

22. A study on Garbage Collection Algorithm for Ubiquitous Real-Time System 429
Chang-Duk Jung, PhD; You-Keun Park, PhD

23. Spam Mail Blocking in Mailing Lists 439
Kenichi Takahashi, Akihiro Sakai and Kouichi Sakurai

1

On Cross-Media Correlation and Its Applications

Wei-Ta Chu
National Chung Cheng University
Taiwan

1. Introduction

Multimedia researches integrate and manipulate different media to facilitate media management, browsing, or presentation such that the developed multimedia systems provide promising learning, entertainment, or information access functionalities. The essence of multimedia is that the processed media are often correlated. Cross-media correlation not only defines the uniqueness of a multimedia system, but also provides clues for efficient/effective processing.

Cross-media correlation comes from related properties between different modalities. Maintenance of these properties is often required to facilitate appropriate presentation and to convey correct information. For example, in a news video, the anchorperson's lip motion should temporally matches with the pronounced sound, or it would be very annoying for us to realize meanings of news stories (Chen, 2001). We call this requirement temporal synchronization, and we say that the anchorperson's lip motion has temporal correlation with the pronounced sound. Another common correlation lies on spatial relationship between different modalities. For example, it's very often that a web page contains text, images, animation, or even videos. How these media arranged on a page, i.e., layout, apparently affects whether information is correctly presented. Images should not occlude the text region, for example. This type of spatial correlation is automatically tackled by web browsers and is transparent to users.

As aforementioned, cross-media correlation apparently or subtly exists in various multimedia systems and plays important roles in many aspects. This article investigates four types of cross-correlations: temporal correlation, spatial correlation, content correlation, and social correlation. We will first give a few brief examples to introduce each type of correlation, and then demonstrate a main system that elaborately exploits one or more cross-media correlations to facilitate media management.

Among four types of correlations, temporal correlation and spatial correlation have been studied for years. For example, in audiovisual display and video compression, temporal synchronization guarantees correct presentation and processing, and different visual media should be placed at the right position to avoid misunderstanding. Moreover, some

multimedia standards such as SMIL (Synchronized Multimedia Integration Language) are proposed to describe multimedia documents. In Section 2, we will describe a multimedia lecturing system that integrates tutor's speech, HTML-based lectures, and guided events to assist English learning.

One example for content correlation is the spreadsheet application. We have numerical data and the corresponding charts. If numerical values change, the corresponding chart changes accordingly. Although numerical data and charts present in different modalities, they convey the same meaning and thus have close content correlation. In Section 3, we will describe a video scene detection system that exploits content correlation between photos and videos taken in journeys. On the basis of video scenes, we propose novel photo summarization and video summarization methods, in which photo summarization is influenced by the correlated video segments, and video summarization is influenced by the characteristics of correlated photos.

Finally, we propose a new kind of cross-media correlation, i.e., social correlation, which emerges in recent years and facilitates multimedia content analysis from a new perspective. In Section 4, social relations between people are described as a network structure. We treat a movie as a small society, and explore the relationships between characters to facilitate movie video understanding.

2. Temporal Correlation and Spatial Correlation

2.1 Introduction

Among various corrleations between media, temporal and spatial correlatons have been studied for several decades. A simple example is slide presentation in Powerpoint. If there are animations to display objects, objects should be shown in correct temporal order to convey correct information. Another interesting example is automatic page turning or score following (Arzt et al., 2008). Music and text-based scores are temporally synchronized such that score pages can automatically change when the played music reaches the end of a page.

In addition to temporal order, objects should be located appropriately to prevent confusion. For objects at the same location or overlap, introduction of depth, or the so-called z-index, facilitates spatial description objects. Therefore, even Powerpoint, which is not thought as a complex multimedia application, needs careful consideration of temporal and spatial correlations between objects.

Recently, integration of various media into the same layout has been the most popular presentation. Synchronized multimedia integration language (SMIL) was proposed to enable authoring and interaction in audiovisual presentations. Through description by the markup language, we can specify media types, time information, spatial information, hyperlinks, and interactive functions for a multimedia document. Due to large-scale of applications and potential commercial benefits, many software developers propose multimedia document format to achieve vivid audiovisual presentations and enrich web-based browsing, such as Adobe's Flash and Microsoft's Silverlight.

Temporal and spatial correlations also present in one of the most important applications – video compression by MPEG and related coding standards. In these standards, audio and

video are compressed separately, and correct decoding is achieved only if audio-video synchronization can be guaranteed. Temporal synchronization is especially important in coding schemes. Moreover, in the MPEG-4 standard, object-based coding is introduced and many coding tools such as sprite coding are proposed to describe spatial relationships among different objects.

In this section, we will introduce a multimedia lecturing system, which integrates text, images, audio, and animation to teach English as a second language. Correlations between different modalities would be discovered and manipulated to facilitate effective tutoring.

2.2 Web-based Synchronized Multimedia Lecturing System

The Web-based Synchronized Multimedia Lecturing system (WSML) aims to capture teaching activities and provides web-based lecturing, which is totally developed by dynamic HTML techniques and can be easily browsed via any webpage browser (Chu & Chen, 2005). Figure 1 show the system architecture composed of three main components: the WSML recorder, WSML document servers, and the WSML browser.

Lectures in the WSML system are all stored in the form of web pages. This characteristic facilitates tightly integration of distance learning and web browsing. To produce a WSML document, the WSML recorder retrieves HTML lectures prepared in advance, and records teacher's tutoring activities, such as teacher's speech and guided events on web pages. The guided events include cursor's trajectory, web page scrolling, highlighting, pen stroke, annotation, and hyperlink. These guided events animate still web pages, and provide vivid and practical tutoring activities on web pages.

In a WSML document, teacher's speech plays as foundation of the time axis, and all other guided events are associated to it by relative timestamps. For example, a scrolling event would occur at 1 minute and 23 seconds relative to the start of teacher's speech, and a highlight event would occur at 2 minute and 45 seconds. Therefore, temporal correlation exists between guided events and teacher's speech. In addition, it's obvious that spatial information of guided media is very important. For example, a pen stroke event should be displayed at the right place in a web page only if the coordinates of pen stroke pixels are appropriately stored. Similarly, the WSML recorder captures coordinates relative to the left-top corner of a web page. Cursor trajectory, highlight event, pen stroke, annotation, and scrolling event correlate to web pages with spatial information.

As shown in Figure 1, all guided events associated with temporal and spatial information are captured by the WSML recorder. At the client side, various media and their correlations should be integrated to present correct lecturing information. The WSML browser at the client side is a web browser equipped with dynamic HTML functionalities. It synchronizes various media and implements audiovisual presentations that were invoked by teachers.

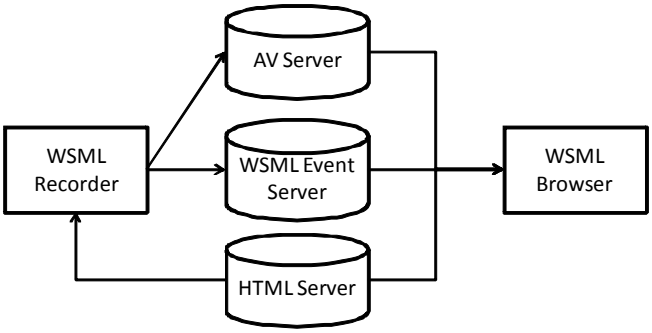


Fig. 1. System architecture of the WSML system.

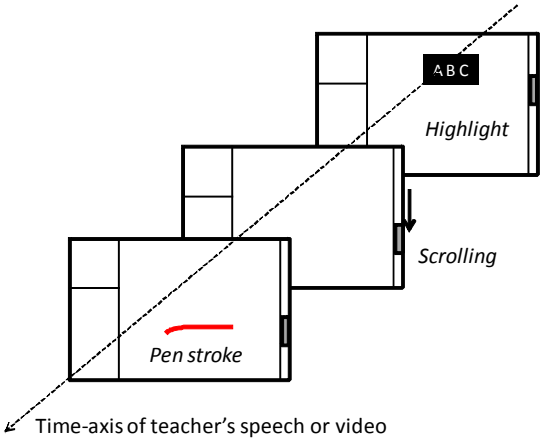


Fig. 2. Presentation scenario of the WSML browser.

Figure 2 shows the presentation scenario of the WSML browser. Along the time-axis of teacher's speech or video, various guided events specially designed for teaching English are displayed in appropriate temporal order and at appropriate positions. In this example, the highlight event is often used to indicate a new vocabulary. The scrolling event is the unique action of web-based lectures. The pen stroke event can be used to highlight a sentence, or to mark corrections on students' articles. In (Chen & Liu, 2009), the authors extend this function by visualized annotations that are commonly used by English teachers to correct students' composition homeworks. Animated visual annotations accompanied with the teacher's speech more accurately point out composition errors and show how to correct.

2.3 Explicit and Implicit Correlation

We can see various correlations exist between media in the WSML system. Nonetheless, according to whether the cross-media correlation can be easily captured or not, correlations in this system can be divided into two categories – explicit correlations and implicit correlations. Explicit correlations are able to be directly captured by the WSML recorder, such as position of the cursor, timestamp of a pen stroke, and the extent of scrolling. For example, the WSML recorder periodically stores the position of cursor, which can be

implemented by dynamic HTML programming (Goodman, 2006). Actually, all spatial and temporal correlations of guided events described above can be explicitly captured.

On the other hand, some correlations are hidden between media and cannot be directly captured by the WSML recorder. In studying English, listening to teacher's reading and comparing it with text in lecture is an important process. Teacher's speech is related to text in web pages, but this relationship cannot be explicitly recorded. Therefore, we would like to develop a speech-text alignment module to discover correlations between two modalities.

Figure 3 shows the system flowchart of the speech-text alignment module. First, we apply a speech recognition module to recognize teacher's reading into text. Note that the recognition module not only transforms speech into text, but also stores timestamp of each recognized word. Through this process, aligning speech and text has been transformed into aligning two text sequences. However, recognized text is imperfect, and many words pronounced similarly would be erroneously recognized. Therefore, we further transform recognized words and text in lectures into phonetic sequences by the CMU pronunciation dictionary (CMU, 2009). Through this process, words with similar pronunciation are encoded as the same string. For example, both *through* and *threw* are converted to "*TH R UW*", in which each term indicates a phoneme symbol.

Based on phonetic strings, we find the longest common subsequence (LCS) to determine the correspondence between recognized text and text in lectures. Due to variations of tense or plurality, words having the same meaning are not necessarily converted into exactly the same phonetic string. For example, the word *student* is converted to "*S T UW D AH N T*", while the word *students* are converted to "*S T UW D AH N T S*". Therefore, we calculate the edit distance between two phonetic strings, and claim them as matched strings if their distance is less than a threshold.

We apply a dynamic programming strategy to find the optimal LCS, which represents the best correspondence between two text sequences. The timestamps of the words matched with recognized words are then determined. For those words that are not aligned, we estimate their timestamps by interpolation or extrapolation. Finally, a fully timestamped sequence is obtained.

The proposed process discovers implicit temporal correlation between teacher's reading and text in lectures. With this correlation, a sentence can be especially highlighted when the teacher speaks it. It is very helpful for students with bad English listening comprehension.

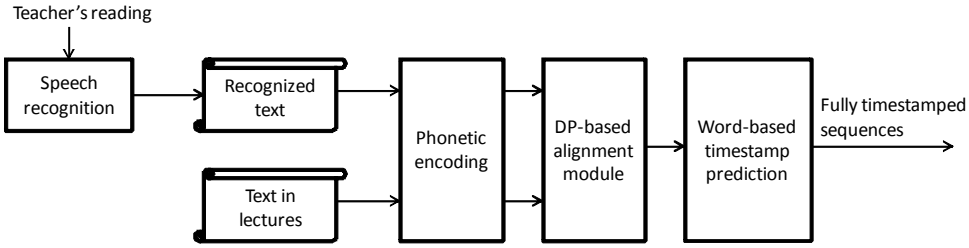


Fig. 3. System flowchart of the speech-text alignment module.

2.4 Evaluation

To evaluate the propose speech-text alignment process, one subjective experiment and one objective experiment are designed. In the subjective experiment that is similar to the investigation in (Steinmetz, 1996), we study how synchronization skews between highlighted sentences and teacher's reading affect human perception. We invited several males and females to evaluate sentence-based sync skews by giving scores from 1 to 5. Smaller score means higher degree of perceiving sync skew. The assessors were asked to see highlighted sentences and listen to corresponding speech, and then evaluate whether a sync error is perceived. In this work, the sentences with scores less than three are claimed to be out-of-sync.

Figure 4 that is by courtesy of (Chu & Chen, 2005) shows the relationships between human perception and sentence-based sync skews. The horizontal axis denotes the sync skews in milliseconds (ms), in which positive values mean sentences show after speech, and negative values mean sentences show before speech. The vertical axis denotes the percentage of assessors who are able to perceive sync errors. This figure can be divided into three regions: 1) The hardly-detected region ranges from -750 ms to +300 ms, in which few assessors can perceive sync errors. 2) The transient region ranges from +300 ms to +1000 ms and from -750 ms to -1350 ms. Synchronization skews in this region can be perceived by more than half of assessors, but they don't lead to misunderstanding. 3) The out-of-sync region spans beyond -1300 ms and +1000 ms. All assessors perceive sync errors, and this presentation largely annoys viewers and may cause misunderstanding.

From Figure 4, human perception is not symmetric when sentences are highlighted before or after the speech. Basically, humans have higher tolerance of speech-text sync skews when sentences are highlighted before start of the corresponding speech. This again confirms the non-uniformity nature of human perception.

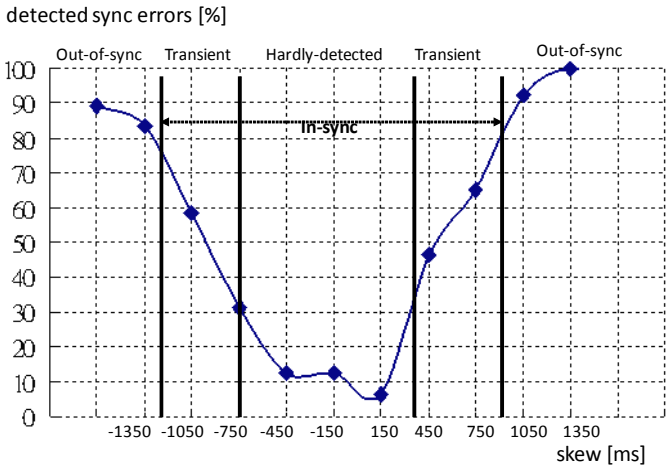


Fig. 4. Relationships between human perception and sync skews.

In the objective experiment, we measure the percentage of in-sync sentences under different speech recognition accuracy. From Figure 4, we define that sentences with sync skews in

"hardly-detected" region or in "transient" region are in-sync. We adopt SPHINX speech recognition engine (Huang, 1993) in real implementation, and achieve different recognition accuracy in different teachers' speeches and in different documents. Because the recognition engine is not specially trained by any specific teacher, the recognition accuracy is generally not high. But fortunately, the proposed speech-text alignment process is still able to determine most of the correspondence based on partially correct text. In this experiment, 80% of sentences are in-sync when the recognition accuracy is 25%, i.e., only quarter of words are correctly recognized. All sentences can be correctly synchronized with speech when the recognition accuracy is higher than 40%. In summary, we effectively discover implicit temporal correlation between speech and text and facilitate teaching English as a second language.

3. Content Correlation

3.1 Introduction

Content correlation between different media or different data is also an important area of multimedia research. A straightforward example is content-based image retrieval. Given an image containing a specific scene or an object, we would like to find images with the same scene or object from image database or from the web. Generally these images have similar appearance or convey similar semantics. Extending from this study, multimedia indexing (Snoek & Worring, 2005), video concept detection (Snoek et al, 2007; Jiang et al, 2008), and video copy detection (Law-To et al., 2007) have been widely investigated in recent years.

Another perceptive of content correlation comes from spreadsheet applications. If we change numerical values, the corresponding chart changes dynamically. Numbers and charts present the same meanings but are in different presentations. In the WSML system described in the previous section, guided events are attached to the corresponding HTML pages. If the HTML lectures are removed, all corresponding guided events should be removed, too. Tight content correlations exist between different media.

In this section, we focus on media captured in journeys and find content correlations between images and videos to facilitate efficient media management, including video scene detection and video/photo summarization. We demonstrate the assist of cross-media correlation on multimedia content analysis.

3.2 Travel Media Analysis

There are at least two challenges in travel media analysis. First, there is no clear structure in travel media. Unlike scripted videos such as news and movies, videos captured in journeys just follow the travel schedule, and content in video may consist of anything people willing or unwilling to capture. Second, because amateur photographers don't have professional skills, the captured photos and videos often suffer from bad quality. The same objects in different photos or video segments may have significant appearance. Due to these characteristics, conventional image/video analysis techniques cannot be directly applied to travel media.

People often take both digital cameras and digital camcorders in journeys. Even with only one of these devices, digital cameras have been equipped with video capturing functions,

and on the other hand, digital camcorders have the "photo mode" to facilitate taking high-resolution photos. Therefore, photos and videos in the same journey often have similar content, and the content correlation can be utilized to analyze travel media.

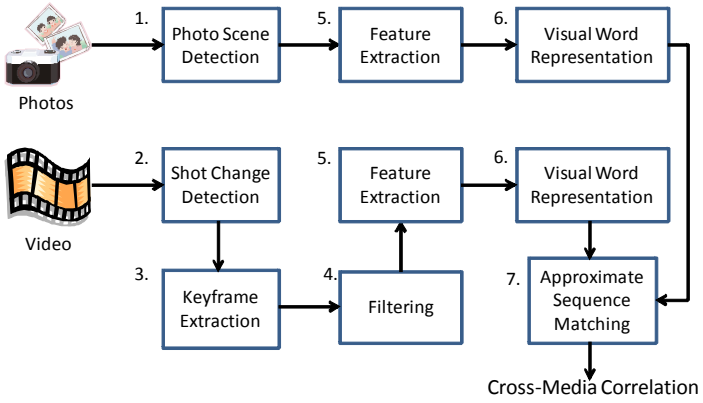


Fig. 5. Flowchart for finding cross-media correlation between photo and video.

3.3 Content Correlation Between Photos and Videos

We assume that travelers alternately take photos and videos when they visit each scenic spot. Along the travel schedule, photos and videos are captured in the same temporal order. Figure 5 shows the flowchart of finding cross-media correlation between a photo set and a video. Note that all video segments captured in the same journey are concatenated as a single video stream according to the temporal order.

- Photo Scene Detection

There are large time gaps between photos in different scenic spots because of transportation. We check time gaps between temporally adjacent photos, and claim a scene boundary exists between two photos if their time gap exceeds a dynamic threshold (Platt et al., 2003). The method proposed in (Platt et al., 2003) has been widely applied in photo clustering, and has been proven very effective. After this time-based clustering, photos taken at the same scenic spot (scene) are clustered together.

- Keyframe Extraction

For the video, we segment it into shots based on difference of HSV color histograms in consecutive video frames (Hanjalic, 2002). To efficiently represent each video shot, one or more keyframes are extracted. We adopt the method proposed in (Chasanis et al., 2007), which automatically determines the most appropriate number of keyframes based on an unsupervised global k-means algorithm (Likas et al, 2003). The global k-means algorithm is an incremental deterministic clustering algorithm that iteratively performs k-means clustering while increasing k by one at each step. The clustering process ends until the clustering results converge.

- Keyframe Filtering

Video shots with blurred content often convey less information, and would largely degrade the performance of correlation determination. To detect blurred keyframes, we check edge information in different resolutions (Tong et al, 2004). The video shots with blurred keyframes are then put aside from the following processes. After video shot filtering, fewer video shots (keyframes) are needed to be examined in the matching process. This filtering reduces influence of blurred content, which may cause false matching between keyframes and photos.

- Visual Word Representation

After the processes above, correlation between photos and videos is determined by matching photos and keyframes. Image matching is an age-old problem, and is widely conducted based on color and texture features. However, especially in travel media, the same place may have significantly different appearances, which may be caused by viewing angles, large camera motion, and overexposure/underexposure. On the other hand, landmarks or buildings with apparent structure are often important clues for image matching. Therefore, we need features that resist to luminance and viewpoint changes, and are able to effectively represent local structure.

We extract SIFT (Scale-Invariant Feature Transform) features (Lowe, 2004) from photos and keyframes. SIFT features from a set of training photos and keyframes are clustered by the k-means algorithm. Feature points belong to the same cluster are claimed to belong to the same *visual word* (Sivic & Zisserman, 2003). Before matching photos with keyframes, visual words in photos and keyframes are collected as visual word histograms. Based on this representation, the problem of matching two image sequences has been transformed into matching two sequences of visual word histograms.

Conceptually, each SIFT feature point represents texture information around a small image patch. After clustering, a visual word presents a concept, which may correspond to corner of building, tip of leaves, and so on. A visual word histogram presents what concepts compose an image. To discover cross-media correlation, we would like to find photos and keyframes that have similar concepts.

- Approximate Sequence Matching

To find the optimal matching between two sequences, we exploit the dynamic programming strategy to find the longest common subsequence (LCS) between them. Given two visual word histogram sequences, $X = \langle \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m \rangle$ and $Y = \langle \boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n \rangle$, which correspond to photos and keyframes, respectively. Each item in these sequences is a visual word histogram, i.e., $\boldsymbol{x}_i = h[j]$, $0 \le j \le N-1$, where N is the number of visual words. The LCS between two subsequences X_m and Y_n is described as follows.

$$LCS(X_m, Y_n) = \begin{cases} LCS(X_{m-1}, Y_{n-1}) + 1, & \text{if } x_m = y_n, \\ \max(LCS(X_{m-1}, Y_n), LCS(X_m, Y_{n-1})), & \text{otherwise,} \end{cases} \quad (1)$$

where X_i denotes the *i*th prefix of X, i.e., $X_i = \langle \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_i \rangle$, and $LCS(X_i, Y_j)$ denotes the length of the LCS between X_i and Y_j. This recursive structure facilitates usage of the dynamic programming approach.

Based on visual word histograms, the equality in eqn. (1) occurs when the following criterion is met:

$$x_i = y_j \quad \text{if } \sum_{k=0}^{N-1} |(h_i(k) - h_j(k))| < \delta, \tag{2}$$

where h_i and h_j are the visual word histograms corresponding to the images x_i and y_j. According to this measurement, if visual word distributions are similar between a keyframe and a photo, we claim that they are conceptually "common" and contain similar content.

3.4 Video Scene Detection

Figure 6 shows an illustrated example of video scene detection based on cross-media correlation. The double arrows indicate the determined correlations between photos and keyframes. If a video shot's keyframe matches the photo in the *i*th photo scene, this video shot is assigned as in *i*th video scene as well. For those video shots without any keyframe matched with photos, we apply interpolation and nearest neighbor processing to assign them (Chu et al., 2009).

Because photo scene detection is much easier than video scene detection, this method first solves an easier problem, finds the correlation between two problems, and then solves the harder problem by consulting with cross-media correlation. This idea brings a new perspective to conduct travel media analysis.

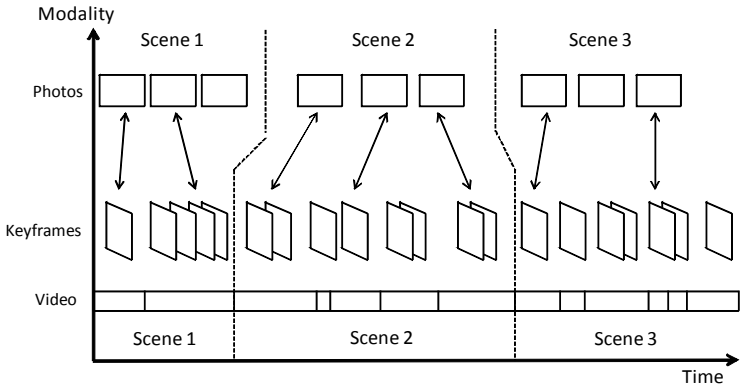


Fig. 6. Illustration of video scene detection using cross-media correlation.

3.5 Photo Summarization and Video Summarization

Correlation determined by the previous process suffices for scene boundary detection. However, to generate summaries, finer cross-media correlation is needed to define importance value of each photo and each keyframe. We call the correlation described above *global cross-media correlation*, which describes matching in terms of visual concepts. In this subsection, we need further analyze *local cross-media correlation* to find matching in terms of objects.

For photos and keyframes in the same scene, we perform finer matching between them by the SIFT matching algorithm (Lowe, 2004). For the photo p_m and the keyframe k_n, we claim

they contain the same object if the number of matched feature points exceeds a predefined threshold τ. This threshold can be adjusted dynamically according to the requirements of users. Note that local cross-media correlation is determined based on SIFT features rather than visual word histograms. Visual word histograms describe global distribution of concepts (visual words), while feature points conveying local characteristics more appropriately describe whether two images have the same building or objects.

- Photo Summarization

The idea of defining each photo's importance comes from two perspectives. First, when a view or an object is both captured in photos and videos, the captured content must attract people more and is likely to be selected into summaries. The second factor is involved with considering characteristics of correlated videos to define photo's importance. When people take a closeup shot on an object, this object must attract people more. Therefore, a photo's importance is set higher if it matches with a keyframe that is between a zoom in action and a zoom out action, or is between a zoom in action and camera turning off. Two factors defining importance values can be mathematically expressed as follows.

Factor 1:
The first importance value of the photo p_m is defined as

$$PT_{1,m} = \frac{I_{1,m}}{\max_{i=1,2,\ldots,M} I_{1,i}}, \tag{3}$$

where *M* is the number of photos in this dataset. The value $I_{1,m}$ is calculated as

$$I_{1,m} = \begin{cases} L_1(h_{p_m}, h_{k_n}), & \text{if the photo } p_m \text{ matches with the keyframe } k_n, \\ 0, & \text{otherwise,} \end{cases}$$

where h_{p_m} and h_{k_n} are visual word histograms of the photo p_m and the keyframe k_n, respectively. The value $L_1(\cdot,\cdot)$ denotes L_1-distance between two histograms.

Factor 2:
The second importance value of the photo p_m is defined as

$$PT_{2,m} = \frac{I_{1,m} \times ZoomIn(p_m)}{\max_{i=1,2,\ldots,M} I_{1,i} \times ZoomIn(p_i)}, \tag{4}$$

where $ZoomIn(p_m) = 1$ if the keyframe k_n that matches with p_m locates between a zoom in action and a zoom out action, or between a zoom in action and camera turning off. The value $ZoomIn(p_m) = 0$, otherwise.

Two importance values are normalized and integrated to form the final importance value:

$$PT_m = \alpha \times PT_{1,m} + \beta \times PT_{2,m}. \tag{5}$$

Currently, the values α and β are set as 1.

Users can set the desired number of photos in summaries. To ensure that the generated summary contains photos of all scenes (scenic spots), we first pick the most important photo in each scene to the summary. After that, we sort photos according to their corresponding importance values in descending order, and pick photos sequentially until the desired number is achieved.

According to the definitions above, only photos that are matched with keyframes have importance values larger than zero. If all photos with importance values larger than zero are picked but the desired number hasn't achieved, we define the importance value of a photo p_i not picked yet by calculating the similarity between p_i and its temporally closest photo p_j that has nonzero importance value, i.e.,

$$T_i = L_1(h_{p_i}, h_{p_j}). \tag{6}$$

We sort the remaining photos according to the alternative importance values in descending order, and pick photos sequentially until the desired number is achieved.

- Video Summarization

Similar to photo summarization, we advocate that photo taking characteristics in a scene affect selection of important video segments. Two factors are also designed. The first factor is the same as that in photo summarization, i.e., video shots whose content also appears in photos are more important. Moreover, a video shot in which more keyframes match with photos is relatively more important. Two factors can be mathematically expressed as follows.

Factor 1:
The first importance value of a keyframe k_m is defined as

$$KT_{1,m} = \frac{I_{1,m}}{\max_{i=1,2,\ldots,M} I_{1,i}}, \tag{7}$$

where *M* is the number of keyframes in this dataset. The value $I_{1,m}$ is calculated as

$$I_{1,m} = \begin{cases} L_1(h_{k_m}, h_{p_n}), & \text{if the keyframe } k_m \text{ matches with the photo } p_n, \\ 0, & \text{otherwise,} \end{cases}$$

where h_{k_m} and h_{p_n} are visual word histograms of keyframe k_m and the photo p_n, respectively.

Factor 2:
The second importance value of the keyframe k_m is defined as

$$KT_{2,m} = \frac{I_{2,m}}{\max_{i=1,2,\ldots,M} I_{2,i}}, \tag{8}$$

where the value $I_{2,m}$ is defined as

$$I_{2,m} = \sum_{j=1}^{J} L_1(h_{k_j}, h_{p_{j^*}}). \tag{9}$$

This expression means there are J keyframes in the shot containing k_m, and the notation p_{j^*} denotes the photo matched with the keyframe k_j.

These two importance values are integrated to form the final importance value of k_m:

$$KT_m = \alpha \times KT_{1,m} + \beta \times KT_{2,m}. \tag{10}$$

Users can set the desired length of video summaries. To ensure that the generated summary contains video segments of all scenes (scenic spots), we first pick the most important keyframe of each scene. Assume that the keyframe k_i is selected, we determine length and location of the video segment S_i corresponding to k_i as

$$S_i = \left(\frac{t(k_{i-1}) + t(k_i)}{2}, \frac{t(k_i) + t(k_{i+1})}{2} \right), \quad (11)$$

where $t(k_i)$ denotes the timestamp of the keyframe k_i, and k_{i-1} and k_{i+1} are two nearest keyframes that are before and after k_i, and with nonzero importance values. Two values in the parentheses respectively denote start time and end time of the segment S_i.

We pick keyframes and their corresponding video segments according to keyframe's importance values until the desired length is achieved. If all keyframes with nonzero importance values are picked but the desired length hasn't achieved, we utilize a method similar to that in eqn. (6) to define remaining keyframes' importance values, and pick appropriate number of keyframe accordingly.

3.6 Evaluation

We evaluate the proposed methods based on seven data sets. Each dataset includes a video clip and a set of photos, and there are totally 85 minutes of videos with 1084 keyframes and 423 photos. Photos are rescaled to smaller sizes due to the efficiency of feature points processing and visual word construction. Videos and photos are captured by different amateur photographers, with different capturing devices.

To measure video scene detection, we calculate purity value (Vinciarelli & Favre, 2007) by comparing detected video scenes and the ground truths. A purity value ranges from 0 to 1, and a larger purity value means that the detection result is closer to the ground truth. We averagely obtain a purity value of 0.95, which is very promising in scene detection, especially in the uncontrolled travel media.

To further show that the proposed method is more appropriate to travel videos, we compare it with the method proposed in (Chasanis et al, 2007). One of the major challenges in scene detection is the over-segmentation problem. We measure this effect in two methods and list the results in Table 1. In each cell of this table, the value (*m*, *n*) denotes that a scene is segmented into *m* and *n* scenes, by the method in (Chasanis et al, 2007) and our method, respectively. For example, in the S2 column for Video 1, (4,1) means that the second scene in Video 1 is segmented into four scenes and one scene by the method in (Chasanis et al, 2007) and our approach, respectively. There are totally 6 scenes in Video 1, but it is segmented into 27 and 8 scenes by two methods. We see the effect of over-segmentation is severe in Chasanis's approach, and our approach works much better from this perspective.

To objectively demonstrate summarization results, we ask content owners to manually select subset of keyframes and photos as summarization ground truth. In generating video summaries or photo summaries, we set the number of keyframes or photos in manual summaries as the targeted number to be achieved. Summarization results are measured by precision values, i.e.,

$$\text{Precision} = \frac{\#\text{ correctly selected keyframes}}{\#\text{ selected keyframes}} \text{ and } \text{Precision} = \frac{\#\text{ correctly selected photos}}{\#\text{ selected photos}}. \quad (12)$$

Note that precision and recall rates are the same due to the selection policy.

	S1	S2	S3	S4	S5	S6	Overall
Video 1	(1,1)	(4,1)	(7,2)	(3,1)	(9,2)	(3,1)	(27,8)
Video 2	(6,1)	(3,1)	(1,1)	(1,1)			(11,4)
Video 3	(1,1)	(1,1)	(1,1)	(3,1)	(2,1)		(8,5)
Video 4	(1,1)	(2,2)	(1,1)	(5,2)	(1,1)		(10,7)
Video 5	(4,2)	(2,2)	(3,1)				(9,5)
Video 6	(3,1)	(2,1)					(5,2)
Video 7	(5,1)	(2,2)	(2,2)	(1,1)	(3,2)	(1,1)	(14,9)

Table 1. Over-segmentation situations in different videos.

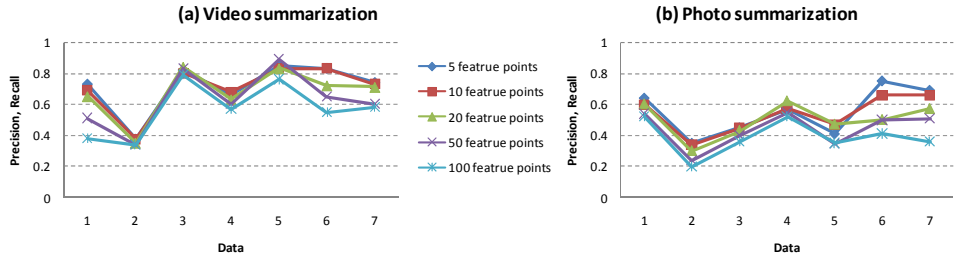


Fig. 7. Performance of (a) video summarization and (b) photo summarization.

Figure 7(a) shows precision/recall rates of video summarization under different matching thresholds τ, while Figure 7(b) shows precision/recall rates of photo summarization. Generally, we see that using five or ten matched points as the threshold we can obtain better summarization results, i.e., looser thresholds draw slightly better performance. Summarization performance of the second dataset is especially worse for both video and photos. This reason is that photos and video in this dataset have less similar content, thus content correlation between them is weak. Except for the second dataset, the proposed methods overall achieves 80% of accuracy in video summarization and 60% of accuracy in photo summarization.

4. Social correlation

4.1 Introduction

In addition to the aforementioned correlations, we introduce a new type of cross-media correlation that hasn't been studied widely before. We can view objects or humans as instances of a modality. The appearance of this modality would be presented by sound, such as anchorperson and guest in radio broadcast programs, or would be presented by visual content, such as actors' faces in TV programs. How objects or humans interact embeds semantics, which facilitates multimedia content management and retrieval.

One of the earliest works about investigating social relationships between humans in multimedia content analysis was proposed in (Vinciarelli, 2007). For radio broadcastings, this work segments audio recordings into segments of different speakers, and utilizes characteristics of duration and interaction between speakers to recognize speaker's role as

an anchorperson or a guest. Correlation between roles is conveyed by speaking duration and order. Another interesting work was proposed in (Rienks et al., 2006). By using features of speech behaviour, interaction, and topics in group meeting videos, they analyze the influence of each participant. With audiovisual features such as speaking length and motion activity, the work in (Hung et al., 2007) estimates the most dominant person in a group meeting. Also for meeting recordings, the work in (Garg et al., 2008) performs role recognition based on lexical information and social network analysis.

In this section, we exploit co-occurrence information of actors in movies to approach movie understanding. We treat a movie as a small society, and model social correlation between actors by a network called *RoleNet* (Weng et al., 2009). Based on this network, the leading roles are automatically determined and community structure of actors is discovered as well. With this information, we are able to develop an accurate story segmentation system.

4.2 RoleNet

A RoleNet is a weighted graph expressed by $G = \langle V, E, W \rangle$, where $V = \{v_1, v_2, ..., v_n\}$ represents a set of roles in a movie, $E = \{e_{ij} | \text{ if } v_i \text{ and } v_j \text{ have relationship}\}$, and the element w_{ij} in W represents strength of the relationship between v_i and v_j. Relationship between roles is developed when they interact with each other. More often two roles appear in the same scenes, more chances they can interact, and closer relationship is built.

For a movie that consists of m scenes and n different roles, we can express the status of occurrence by a matrix $A = [a_{ij}]_{m \times n}$, where the element

$$a_{ij} = \begin{cases} 1, & \text{if the } j\text{th role appears in the } i\text{th scene,} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The jth column vector, $\boldsymbol{a}_j = (a_{1j}, a_{2j}, ..., a_{mj})$, of A denotes the scenes where the jth role appeared. The co-occurrence of the ith role and the jth role is identified by

$$w_{ij} = \sum_{k=1}^{m} a_{ki} a_{kj} = \boldsymbol{a}_i^T \boldsymbol{a}_j, \text{ for } i \neq j. \tag{14}$$

The value of w_{ij} is actually the inner product of $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$. This measurement can be generalized to the whole matrix. The co-occurrence status between roles can be expressed by $W_{n \times n} = A^T A$, in which w_{ij} is especially set as 0 when $i = j$.

The left side of Figure 8 shows the RoleNet of the movie "You've Got Mail," in which thickness of an edge represents weights on it. Relationships between roles seem to be complicated, and the goal of this work is to find the leading roles and associated communities.

4.3 Social-Based Analysis

- Leading Role Determination

There is a large gap between the impact of leading roles and that of supporting roles. Based on this observation, the problem of leading role determination can be mathematically expressed as follows.

$$\Gamma^* = \arg\max_{\Gamma}(\min \Theta_1 - \max \Theta_0), \tag{15}$$

where $\Theta_1 = \{c_i | \ell_i = 1\}$ and $\Theta_0 = \{c_i | \ell_i = 0\}$.

$\Gamma = \{\ell_i, i = 1, 2, ..., n\}$

$$\begin{cases} \ell_i = 1 & \text{if the } i\text{th role is assigned as a leading role,} \\ \ell_i = 0 & \text{otherwise,} \end{cases}$$

where n is the number of roles, Γ is a set of binary labels representing which roles are assigned as leading roles. The value c_i is weighted degree centrality, which defines the importance value of the node i, and is calculated by

$$c_i = \sum_{j \neq i} w_{ij}. \tag{16}$$

The set Θ_1 represents centrality values of the roles assigned to leading roles. The physical meaning of $(\min \Theta_1 - \max \Theta_0)$ is the difference of centrality values between the least important leading role and the most important supporting role. The result Γ^* we want is the labels that cause the largest centrality difference.

Taking "You've Got Mail" as an example, we calculate the centrality value of each role, and then sort the centrality values in descending order, as shown in Figure 8(a) and Figure 8(b). Next, the differences of centrality values between two adjacent roles in Figure 8(b) are calculated. Figure 8(c) shows the centrality difference distribution, in which each point represents a boundary between two roles. Finally, we find the maximum point in the difference distribution, which represents the largest gap in centrality. In this example, the difference between the role no. 1 and the role no. 7 is maximal, and therefore the role no. 2 and role no. 1 are claimed as leading roles.

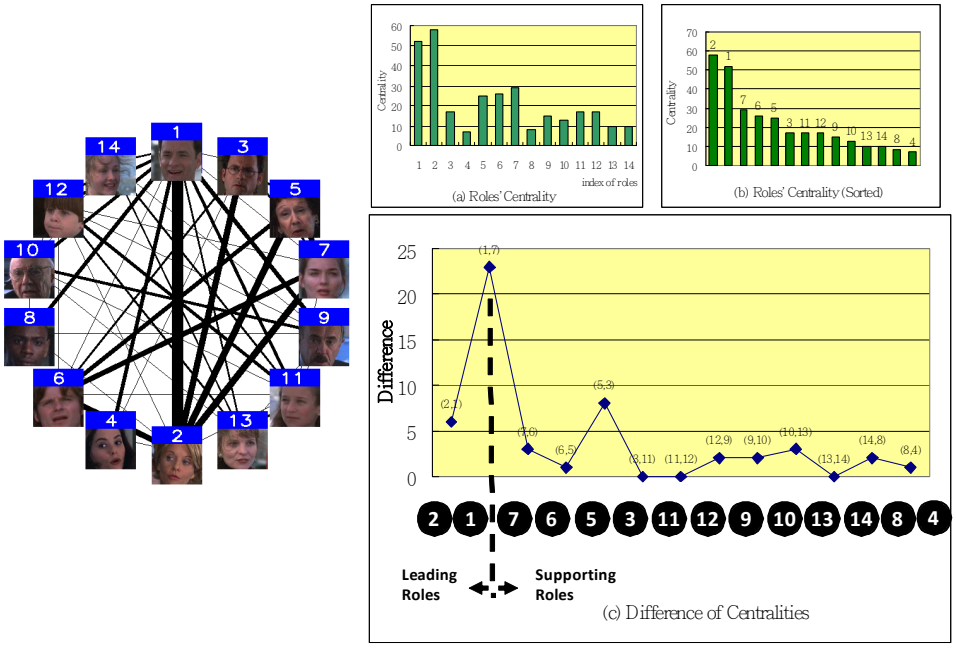


Fig. 8. Left: the RoleNet of "You've Got Mail"; Right: The process of leading role determination.

- Community Identification

After determining the leading roles, we devise a method to appropriately group certain roles into communities. Because leading roles may pass through several communities, it's not reasonable to assign them into any community. Therefore, we first remove leading roles and the edges linked to them from the RoleNet. Then, the following iterative algorithm is applied to the modified RoleNet. We use the value t to index the community's evolution situation. The value t is initialized as 0 in the beginning and increases by one when the community situation changes.

Algorithm 1: Community Identification

1. Initialize every individual node as a community. The set of community is denoted as $\Pi_t = \{T_1^t, T_2^t, ..., T_n^t\}$, t = 0, if there are initially n individual nodes. The size of the pth community in Π_t is denotes as $|T_p^t|$, which is the number of nodes included in this community.
2. From the modified RoleNet, find the edge that has the largest weight, say the edge e_{ij} between the node v_i and the node v_j, $v_i \in T_p^t$ and $v_j \in T_q^t$, then
   - If $|T_p^t| \geq 1$ and $|T_q^t| = 1$, then $T_p^{t+1} = T_p^t \cup T_q^t$, $\Pi_{t+1} = \Pi_t - \{T_q^t\}$, and t = t+1.
   - If $|T_p^t| > 1$ and $|T_q^t| > 1$, then keep current community situation.
3. Remove the edge e_{ij} from the modified RoleNet and go to Step 2 until all edges have been removed.

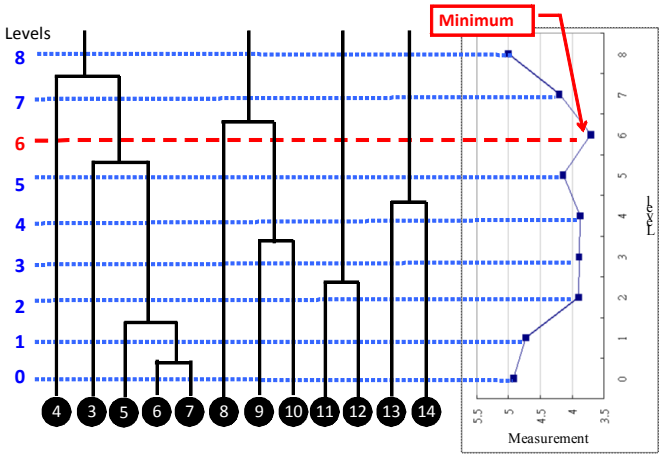


Fig. 9. Dendrogram of the clustering process.

The progress of this algorithm can be illustrated as a dendrogram, which describes how we cluster communities at each step. As shown in Figure 9, the roles no. 6 and no. 7 are first categorized together (t=1), then the role no. 5 is merged into this community at the second level (t=2). (We say "level" but not "iteration" because the community situation may not change at each iteration.) The same process can be iteratively applied until all nodes have been examined.

Each level in the dendrogram represents a case of community situation. Now the problem is to determine which level in the dendrogram is the best. We design a measurement to evaluate communities at different levels. For the level *t*, the measurement is defined as

$$AvgW_t = \frac{\sum w_{ij}}{||\Pi_t||}, \forall v_i \in T_p^t, v_j \in T_q^t, p \neq q, \quad (17)$$

where Π_t denotes the community situation at the level *t*, and $||\Pi_t||$ denotes the number of communities in this case. The value $AvgW_t$ represents the average weight between different communities at level *t*. The right part of Figure 9 shows the measures at different levels.

In community identification, we prefer that roles in different communities are least related. Therefore, we pick the community case that causes the minimal $AvgW$. In Figure 9, the minimal $AvgW$ value occurs at the sixth level, in which six communities are found to include roles {4}, {3, 5, 6, 7}, {8}, {9, 10}, {11, 12}, and {13, 14}, respectively.

To our knowledge, there is no prior study to conduct movie understanding from social perspective. How characters interact affect us to understand a movie. By analyzing social correlation between roles, we not only can determine characters with high impacts, but also determine construct community structure. In the next subsection, we further demonstrate using social correlation to facilitate story segmentation.

4.4 Story Segmentation

- Scene Representation

We first define the representation of scenes. The major difference between the proposed representation and conventional ones is that we describe scenes by "the context of roles" rather than audiovisual features. Story segmentation is achieved by comparing the role's context in successive scenes.

Let $r(k)$ denote the identification of the kth character in a specific scene. The relationship between this role and others can be expressed by a "profile vector" $\boldsymbol{w}_{r(k)} = (w_{1r(k)}, w_{2r(k)}, ..., w_{nr(k)})$, which is the $r(k)$-th column vector of the matrix W. The vector $\boldsymbol{w}_{r(k)}$ denotes the closeness between the role no. $r(k)$ and others. It is normalized into a unit vector. For the *i*th scene, we collect the profile vectors of the roles appearing in this scene to form a matrix $CM_i = [\boldsymbol{w}_{r(1)}\boldsymbol{w}_{r(2)}...\boldsymbol{w}_{r(p)}]$. It is an *n* by *p* matrix if there are *p* roles in this scene and there are totally *n* roles in the movie. Similarly, the matrix CM_j for the *j*th scene is denoted by $CM_j = [\boldsymbol{w}_{r(1)}\boldsymbol{w}_{r(2)}...\boldsymbol{w}_{r(q)}]$, if there are *q* roles in this scene. Note that the vector $\boldsymbol{w}_{r(k)}$ in the *i*th scene and the vector $\boldsymbol{w}_{r(k)}$ in the *j*th scene would be different, i.e., the identifications of the *k*th characters in these two scenes are different. Using the notations of $\boldsymbol{w}^i_{r(k)}$ and $\boldsymbol{w}^j_{r(k)}$ is more precise but dazzles readers. Therefore, we use the simplified notation in the following description.

The context-based similarity between "the *s*th role in the *i*th scene" and "the *t*th role in the *j*th scene" is defined as the inner product of two corresponding profile vectors: $\boldsymbol{w}_{r(s)} \cdot \boldsymbol{w}_{r(t)} = \boldsymbol{w}^T_{r(s)}\boldsymbol{w}_{r(t)}$. By calculating the context-based similarities between every two roles in two successive scenes, the similarity between the *i*th scene and the *j*th scene can be

expressed in a matrix form: $CM_{ij} = CM_i^T CM_j$. Finally, the context-based difference between the *i*th scene and the *j*th scene is defined as

$$d_{ij} = 1 - \frac{1}{pq} \sum_{s=1}^{p} \sum_{t=1}^{q} CM_{ij}(s,t), \tag{18}$$

which represents the average difference between pairs of roles in two different scenes. The value is between 0 and 1.

- Story Segmentation

Going through the whole movie, we can plot a "difference curve" that represents difference between adjacent scenes. Based on this curve, the goal of story segmentation is to find appropriate scene boundaries that represent changes of stories. We propose a story segmentation method called "storyshed," which is motivated by the watershed algorithm for image segmentation but is modified to meet the need of this task.

Denote the set of scene boundaries as $B = \{b_1, b_2, ..., b_{N-1}\}$, in which the element b_i denotes the boundary between the *i*th and the (*i*+1)-th scenes, and N is the total number of scenes. The proposed storyshed algorithm is to determine whether a scene boundary is a story boundary, based on the context-based difference between adjacent scenes. The context-based difference corresponding to b_i is denoted by d_i. In the first step of the segmentation process, we first find the valleys and peaks from the difference curve by checking d_i. That is,

$$\begin{cases} b_i \in Y, & \text{if } d_i < d_{i-\alpha_1} \text{ and } d_i < d_{i+\alpha_2}, \\ b_i \in P, & \text{if } d_i > d_{i-\alpha_1} \text{ and } d_i > d_{i+\alpha_2}, \\ b_i \in OT, & \text{otherwise}, \end{cases} \tag{19}$$

$\alpha_1 = \min\{j | j \in A_1\}, \quad A_1 = \{k | (d_i - d_{i-k}) \neq 0, 1 \leq k \leq i-1\}$,
$\alpha_2 = \min\{j | j \in A_2\}, \quad A_2 = \{k | (d_i - d_{i+k}) \neq 0, 1 \leq k \leq (N-1-i)\}$,
where $2 < i < N-2$, Y denotes the set of scene boundaries in valleys, P denotes the set of boundaries in peaks, and the set OT includes all other boundaries. Thus, $Y \cup P \cup OT = B$.

Initialize an empty set SB that will store the story boundaries. For each valley y_j in Y, find the nearest peaks to it. Let's denote the left peak of y_j as p_1 and the right peak of y_j as p_2, $p_1 \in P$ and $p_2 \in P$. Fill water into this valley until the height of the horizontal just floods p_1 or p_2. Without loss of generality, assume that p_1 is flooded first and the height of p_1 is H. Pick the scene boundaries b_k between the peaks p_1 and p_2, for which the corresponding context-based difference d_k is no less than H. Therefore, the set of story boundaries would be $SB = SB \cup \{b_k\}$.

The results reveal the boundaries that scenes around them have significantly different context information. However, the storyshed algorithm only considers local characteristics and may miss the ones that are indeed story boundaries. To consider the global characteristics, we take average of the context-based difference values of all peaks as a global threshold. If the difference value corresponding to a scene boundary is larger than this threshold, it's viewed as a story boundary. The global threshold is adaptively calculated according to the social relationships between roles in different movies.

Figure 10 shows an example of storyshed segmentation with the global threshold. Two valleys (solid black circles) in this example are at b_{t+2} and b_{t+6}. The nearest peaks to b_{t+2} are at b_{t+1} and b_{t+4}, and that to b_{t+6} are at b_{t+4} and b_{t+7}. With the global threshold, the boundary b_{t+5} is

detected as a story boundary as well. Processing with this global threshold is the same as applying a constraint so that height of water is limited to be lower than the threshold.

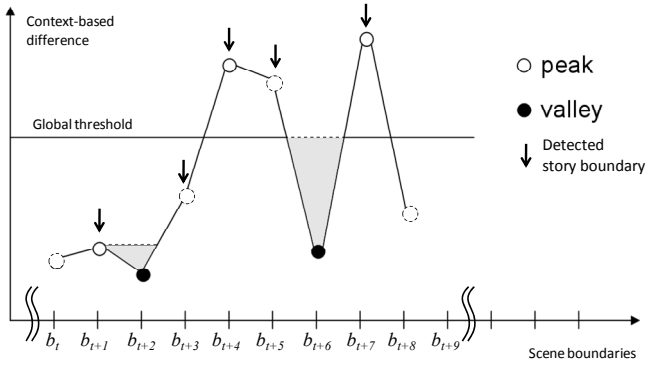


Fig. 10. An example of the storyshed segmentation method with a global threshold.

4.5 Evaluation

We use ten Hollywood movies and three TV shows to evaluate the proposed methods. Total length of evaluation data is over 21 hours, and 428 story segments are included. For each movie, we asked three persons to manually label leading roles. The ones that were labeled as the leading ones by all the three persons are treated as the ground truth. Table 2 shows performance of leading role determination. The numbers in each cell denote the indices of roles. Because the trend of appearance of leading roles is often apparent, the proposed method for leading role determination achieves very promising performance.

To evaluate story segmentation, the ground truths of story boundaries were also decided manually. We invited several subjects who were not familiar the goal and process of our work and knew nothing about the chapter information in advance. They were asked to examine every scene change boundary and decide whether it's a story boundary. We measure story segmentation performance in terms of purity value again (Vinciarelli & Favre, 2007). In addition, to show the superiority of utilizing social correlation in story segmentation, we compare our approach with a conventional tempo-based approach (Chen et al., 2004).

MovieID	M1	M2	M3	M4	M5	M6	M7
GT	1	1, 2	1, 2, 6	1, 2	1	1	1
Det.	1	1, 2	1, 2, 6	1, 2	1	1	1
MovieID	M8	M9	M10	S1	S2	S3	
GT	1	1	1	1	1, 2, 4, 5, 6, 7	1, 2, 3, 4	
Det.	1	1	1	1	1, 2, 3, 4, 5, 6, 7, 9	1, 2, 3	

Table 2. Performance of leading role determination. GT: Ground Truth; Det.: Determined resuls.

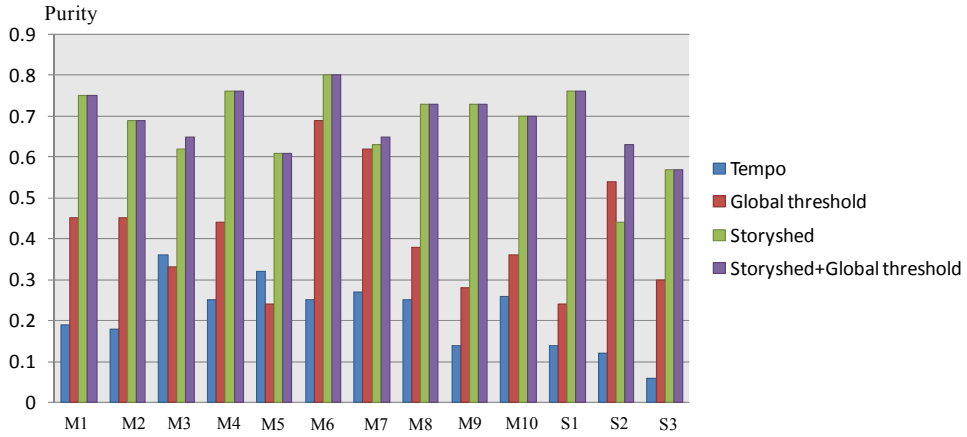


Fig. 11. Performance of story segmentation.

Figure 11 shows that the social-based approach works much better than the tempo-based one. Although the performance improvement varies in different movies, the storyshed algorithm has significantly better performance than thresholding. After combining storyshed with thresholding, the result is even slightly better than the storyshed algorithm, especially in M3, M7, and S2.

5. Conclusion

This article presents cross-media correlation and its applications in various multimedia systems. Four perspectives of correlations are investigated: temporal correlation, spatial correlation, content correlation, and social correlation. Regarding temporal and spatial correlations, we introduce a web-based multimedia lecturing system, in which various guided events have tight temporal correlation to teacher's speech, and have spatial correlation to HTML-based lectures. In addition, we design a speech-text alignment module to discover hidden correlation between teacher's reading and text in lectures. Elaborate usage of temporal and spatial correlation facilitates vivid multimedia lecturing for teaching English as a second language.

Regarding content correlation, a travel media analysis system is introduced to find correlation between photos and videos. The essential idea of this work is to solve a harder problem (video scene detection) by first solving an easier problem (photo scene detection) and then consulting with the correlation between two modalities. Besides, we argue that characteristics of photo taking can be exploited to conduct video summarization, and vice versa. We show that appropriately utilizing cross-media correlation facilitates effective multimedia content analysis.

Regarding social correlation, we analyze relationships between characters to determine leading roles and community structure. Community information makes us approach movie understanding and efficiently movie video management/retrieval. Evolution of social relationships is also studied to conduct story segmentation. We have demonstrated this

approach more matches human's cognition and works better than conventional content-based segmentation methods.

Benefits of employing cross-media correlation in multimedia content analysis are evident. Although this article describes three practical applications, correlations are often specially extracted according to characteristics of different environments or applications. Systematic description and unified framework for cross-media correlation are needed to extend its applicable domain. Finally, this article introduces the new social correlation in multimedia content analysis. We wish to draw more attention on this issue; especially community websites and online video sharing have explosively grown in recent years.

Acknowledgement
This work was partially supported by the National Science Council of the Republic of China under grants NSC 98-2221-E-194-056 and NSC 97-2221-E-194-050.

6. References

Arzt, A.; Widmer, G. & Dixon, S. (2008). Automatic Page Turning for Musicians via Real-Time Machine Listening. *Proceedings of European Conference on Artificial Intelligence*.

Chasanis, V.; Likas, A. & Galatsanos, N. (2007). Scene Detection in Videos Using Shot Clustering and Symbolic Sequence Segmentation. *Proceedings of IEEE International Conference on Multimedia Signal Processing*, pp. 187-190.

Chen, T. (2001). Audiovisual Speech Processing. *IEEE Signal Processing Magazine*, Vol. 18, No. 1, pp. 9-21.

Chen, H.-W.; Kuo, J.-H.; Chu, W.-T. & Wu, J.-L. Action Movies Segmentation and Summarization Based on Tempo Analysis. *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 251-258.

Chen, H.-Y. & Liu, K.-Y. (2009). WMA : A Marking-Based Synchronized Multimedia Tutoring System for English Composition Studies. *IEEE Transactions on Multimedia*, Vol. 11, No. 2, pp. 324-332.

Chu, W.-T. & Chen, H.-Y. (2005). Towards Better Retrieval and Presentation by Exploring Cross-Media Correlations. *ACM Multimedia Systems Journal*, Vol. 10, No. 3, pp. 183-198.

Chu, W.-T.; Lin, C.-C. & Yu, J.-Y. (2009). Using Cross-Media Correlation for Scene Detection in Travel Videos. *Proceedings of ACM International Conference on Image and Video Retrieval*.

Carnegie Mellon University. (2009). CMU Pronouncing Dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Garg, N.P.; Favre, S.; Salamin, H.; Hakkani Tur, D. & Vinciarelli, A. (2008). Role Recognition for Meeting Participants: An Approach Based on Lexical Information and Social Network Analysis. *Proceedings of ACM Multimedia Conference*, pp. 693-696.

Goodman, D. (2006). *Dynamic HTML : The Definitive Reference*. O'Reilly Media, Inc.

Hanjalic, A. (2002). Shot-Boundary Detection : Unraveled or Resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 2, pp. 90-105.

Huang, X.; Alleva, F.; Hon, H.W.; Hwang, M.Y. & Rosenfeld, R. (1993). The SPHINX II Speech Recognition System : An Overview. *Computer Speech and Language*, Vol. 2, No. 7, pp. 137-148.

Hung, H.; Jayagopi, D.; Yeo, C., Friendland, G., Ba, S., Ramchandran, J.; Mirghafori, N. & Gatica-Perez, D. (2007). Using Audio and Video Features to Classify The Most Dominant Person in A Group Meeting. *Proceedings of ACM Multimedia Conference*, pp. 835–838.

Jiang, Y.-G.; Yanagawa, A.; Chang, S.-F. & Ngo, C.-W. (2008). CU-VIREO374: Fusing Columbia374 and VIREO374 for Large Scale Semantic Concept Detection. *Columbia University ADVENT Technical Report* #223-2008-1.

Law-To, J.; Chen, L .; Joly, A.; Laptev, I.; Buisson, O.; Gouet-Brunet, V.; Boujemaa, N. & Stentiford, F. (2007). Video Copy Detection : A Comparative Study. *Proceedings of ACM International Conference on Image and Video Retrieval*, pp. 371-378.

Likas, A.; Vlassis, N. & Verbeek, J.J. (2003). The Global K-means Clustering Algorithm. *Pattern Recognition*, Vol. 36, pp. 451-461.

Lowe, D. (2004) Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110.

Platt, J.C.; Czerwinski, M. & Field, B.A. (2003). PhotoTOC: Automating Clustering for Browsing Personal Photographs. *Proceedings of IEEE Pacific Rim Conference on Multimedia*, pp. 6-10.

Rienks, R.; Zhang, D. & Post, W. (2006). Detection and Application of Influence Rankings in Small Group Meetings. *Proceedings of International Conference on Multimodal Interfaces*, pp. 257-264.

Sivic, J. & Zisserman, A. (2003). Video Google: A Text Retrieval Approach to Object Matching in Videos. *Proceedings of the International Conference on Computer Vision*, Vol. 2, pp. 1470-1477.

Snoek, C.G.M. & Worring, M. (2005). Multimodal Video Indexing: A Review of the State-of-the-art. *Multimedia Tools and Applications*, Vol. 25, No. 1, pp. 5-35.

Snoek, C.G.M.; Huurnink, B.; Hollink, L.; de Rijke, M.; Schreiber, G. & Worring, M. (2007). Adding Semantics to Detectors for Video Retrieval. *IEEE Transactions on Multimedia*, Vol. 9, No. 5, pp. 975-986.

Steinmetz, R. (1996). Human Perception of Jitter and Media Synchronization. *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 1, pp. 61-72.

Tong, H.; Li, M.; Zhang, H.-J. & Zhang, C. (2004). Blur Detection for Digital Images Using Wavelet Transform. *Proceedings of IEEE International Conference on Multimedia & Expo*, pp. 17-20.

Vinciarelli, A. (2007). Spearkers Role Recognition in Multiparty Audio Recordings Using Social Network Analysis and Duration Distribution Modeling. *IEEE Transactions on Multimedia*, Vol. 9, No. 6, pp. 1215-1226.

Vinciarelli, A. & Favre, S. (2007). Broadcast News Story Segmentation Using Social Network Analysis and Hidden Markov Models. *Proceedings of ACM Multimedia*, pp. 261-264.

Weng, C.-Y.; Chu, W.-T. & Wu, J.-L. (2009). RoleNet: Movie Analysis from the Perspective of Social Networks. *IEEE Transactions on Multimedia*, Vol. 11, No. 2, pp. 256-271.

An Intelligence Fault Tolerance Agent for Multimedia CSCW based on Situation-Awareness

Eung-Nam Ko
Division of Information & Communication, Baekseok University
Korea

1. Introduction

CSCW (Computer Supported Cooperative Work) is the study of how people use technology, with relation to hardware and software, to work together in shared time and space. CSCW began as an effort by technologists to learn from anyone whom could help them better understand group activity and how one could use technology to support people in their work (Grudin J, 1994). There are two dimensions to CSCW-space and time. Figure 1 shows examples relevant to this study in the usual quadrants (J Lama et al, 2006). In this study, we describe video conferencing in the 3rd quadrant.

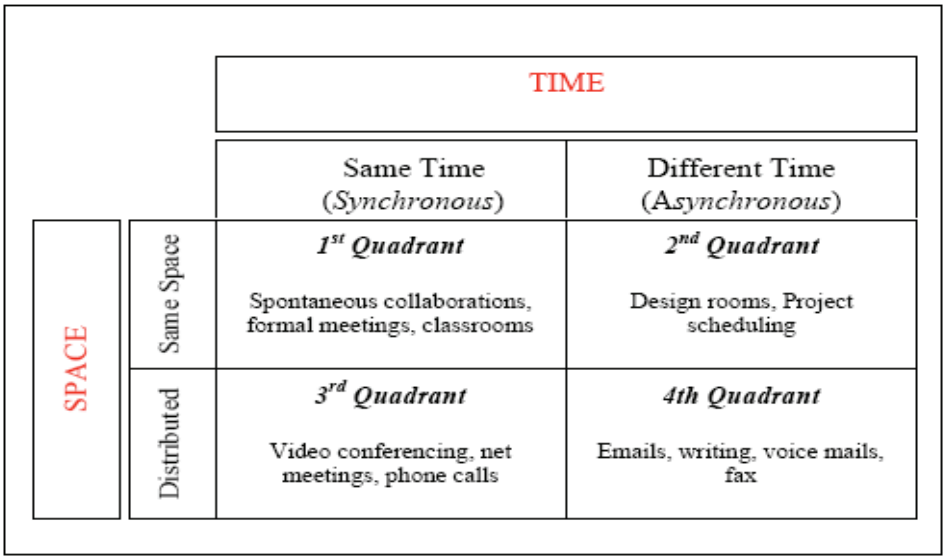


Fig. 1. CSCW quadrants

Distance education is planned learning that normally occurs in a difference place from teaching and as a result requires special techniques of course design, special instructional techniques, special methods of communication by electronic and other technology, as well as special organizational and administrative arrangements. It presents a general systems model that describes the main component processes and elements of a distance education institution, program, unit, consortium, or course (Michael G.Moore et al, 1996).
A system includes the subsystems of knowledge sources, design, delivery, interaction, learning, and management. The more integrated these are in practice, the greater will be the effectiveness of the distance education organization (Jae Young Ahn et al, 1996).
The implementation of interactive multimedia distance education system can be recognized as a diversification of videoconferencing system which first appeared in the 1980's. Early implementations of videoconferencing systems were circuit-based systems relying on dedicated video devices, telephone networks or leased lines. After the early 1990's, the major basis of videoconferencing system moved to packet-based systems which operate on computer network (Francois Fluckiger, 1995; C.W.Loftus et al, 1995).
However, since this new education system must be developed in a way that combines various field of technologies, including group communication and distributed multimedia processing which are the basis of packet based videoconferencing systems, integrated service functions such as middle ware are required to support it (ITU-T Recommendation, 1995).
The development of middleware is closely related to the evolution of ubiquitous computing began in the mid of 1970s, when the PC first brought computers closer people. With the advent of networking, personal computing evolved into distributed computing. With seamless access and World Wide Web, distributed computing marked a next step toward pervasive computing, and mobile computing emerged from the integration of cellular technology with the Web. The "anytime anywhere" goal of mobile computing is essentially a reactive approach to information access, and it prepare the way for pervasive computing's proactive "all the time everywhere" goal (Satyanarayanan, M , 2001; Saha, D, Mukherjee, A. 2003).
The requirement of distributed multimedia applications is the need for sophisticated quality of service(QoS) management. In most traditional computing environments, requests for a particular service are either met or ignored. In a multimedia system, however, the quality of the service achieved is central to the acceptability of the application. In terms of distributed multimedia systems, the most important categories for quality of service are a timeless, volume, and reliability (Gordon Blair, Jean-Bernard Stefani, 1997).
In this chapter of book, we propose a method for increasing reliability. The rest of this chapter of book is organized as follows. Section 2 describes context awareness. Section 3 denotes our approach. Section 4 evaluate and concludes the chapter.

2. Related Works

Distance education systems must be able to support real-time interaction, temporal/spatial synchronization and floor control for smooth interaction (Jae Y. Ahn et al, 1996; Gil C. Park et al, 1995).
In ubiquitous computing environment, application should be able to properly adapt itself according to its own context information coming from ubiquitous sensors (Younglok Lee et

al, 2006). The recently many interest comes to collect in about the sensor network and it has environmental monitoring, a weather measurement, a military field and etc. specific goal and actively from many field, the researches are advanced (Man Seok Yang et al, 2009).

Context awareness (or context sensitivity) is an application software system's ability to sense and analyze context from various sources; it lets application software take different actions adaptively in different contexts (S. Yau, F. Karim et al, 2002). In a ubiquitous computing environment, computing anytime, anywhere, any devices, the concept of situation-aware middleware has played very important roles in matching user needs with available computing resources in transparent manner in dynamic environments (S. Yau, F. Karim et al, 2002). Although the situation-aware middleware provides powerful analysis of dynamically changing situations in the ubiquitous computing environment by synthesizing multiple contexts and users' actions, which need to be analyzed over a period of time, it is difficult to analyze Quality of Service (QoS) of situation-aware applications because the relationship between changes of situations and resources required to support the desired level of QoS is not clear.

The field of fault-tolerant computing has evolved over the past twenty-five years (H. Zhang A et al, 1995). In spite of this current trend, however, study on fault- tolerance of application software has not actually been enough. Generally, fault-tolerance system can be classified as software techniques, hardware techniques and composite techniques (Victor P. Nelson and Bill D. Carroll). The tolerance of software faults is in most cases more difficult than dealing with hardware faults since most software-fault mechanisms are not well understood and do not lend themselves readily to "nice" techniques such as error coding (Dhiraj K. Pradhan, 1996).

Since the application needs of middleware services and computing environment (resources) keep changing as the application change, it is difficult to analyze: whether it is possible that Quality of Service (QoS) of requirements for error-detection recovery are met, and what QoS requirements for error-detection recovery have tradeoff relationships.

Thus, there is a great need for fault tolerance algorithm in situation-aware middleware to provide dependable services in ubiquitous computing in case of errors occurrence. Context awareness (or context sensitivity) is an application software system's ability to sense and analyze context from various sources; it lets application software take different actions adaptively in different contexts. With the rapid development of multimedia and network technology, more and more digital media is generated (T. Zhang et al, 1999). Although the situation-aware middleware provides powerful analysis of dynamically changing situations in the ubiquitous computing environment by synthesizing multiple contexts and users' actions, which need to be analyzed over a period of time, access control in using multimedia shared object causes a problem of the seam in the ubiquitous computing environment. It is difficult to avoid a problem of the seam in the ubiquitous computing environment for seamless services. There is a great need for situation-aware middleware to be able to predict whether all QoS requirements of the applications are satisfied and analyze tradeoff relationships among the QoS requirements, if all QoS requirements cannot be satisfied to determine a higher priority of QoS requirements.

This paper proposes a new model of intelligence fault tolerance agent for multimedia CSCW based on situation-aware ubiquitous computing. It is analyzing the window and attributes of the object, and based on this, a mechanism that offers a seamless multimedia view without interfering with access control is also suggested. Our FTA model is to present the

relationship of missions, actions, QoS and resources. FTA model is used to detection and recover the QoS resource errors among actions.

3. Our Approach

3.1 A Situation-Awareness Middleware: RCSM

In this paper, we propose a new access control mechanism in situation-aware middleware. A conceptual architecture of situation-aware middleware based on Reconfigurable Context-Sensitive Middleware (RCSM) is proposed in (Saha, D, Mukherjee, A., 2003). Figure 2 shows Architecture of RCSM.

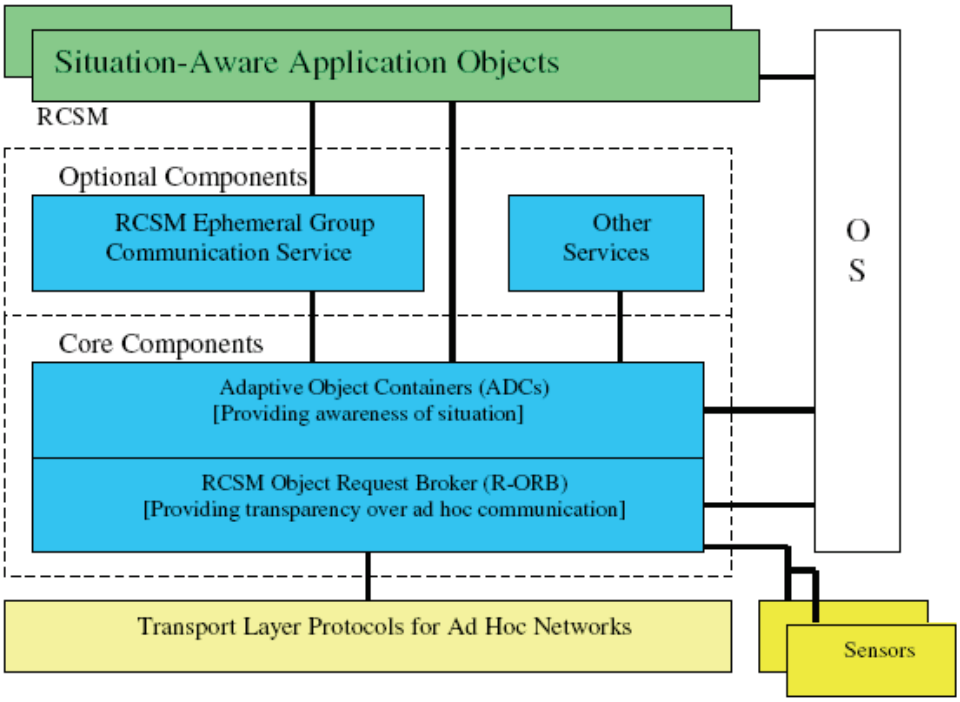


Fig. 2. Architecture of RCSM

All of RCSM's components are layered inside a device. The Object Request Broker of RCSM (R-ORB) assumes the availability of reliable transport protocols; one R-ORB per device is sufficient. The number of ADaptive object Containers (ADC) s depends on the number of context-sensitive objects in the device. ADCs periodically collect the necessary "raw context data" through the R-ORB, which in turn collects the data from sensors and the operating system. Initially, each ADC registers with the R-ORB to express its needs for contexts and to publish the corresponding context-sensitive interface. RCSM is called reconfigurable because it allows addition or deletion of individual ADCs during runtime (to manage new

or existing context-sensitive application objects) without affecting other runtime operations inside RCSM (Saha, D, Mukherjee, A., 2003).
Ubiquitous applications require use of various contexts to adaptively communicate with each other across multiple network environments, such as mobile ad hoc networks, Internet, and mobile phone networks. However, existing context-aware techniques often become inadequate in these applications where combinations of multiple contexts and users' actions need to be analyzed over a period of time. Situation-awareness in application software is considered as a desirable property to overcome this limitation. In addition to being context-sensitive, situation-aware applications can respond to both current and historical relationships of specific contexts and device-actions. An example of situation-aware applications is a multimedia distance education system. The development of multimedia computers and communication techniques has made it possible for a mind to be transmitted from a teacher to a student in distance environment. However, it did not include access control support in the architecture of situation-aware middleware.
However, it did not include QoS support in the architecture. In this paper, we focus on how to represent QoS requirements in situation-aware middleware. In the next subsection, we will present a conceptual model for QoS requirements representation in situation-aware middleware.

3.2 RCSM Optional Components: DOORAE

Ubiquitous applications require use of various contexts to adaptively communicate with DOORAE agent layer includes many agents. They are AMA (Application Management Agent), IA (Intelligent Agent), SMA (Session Management Agent), ACA (Access Control Agent), MCA (Media Control Agent) , and FTA (Fault Tolerance Agent). The organization of DOORAE agent layer running on RCSM is shown in Figure 3.
AMA (Application Management Agent) consists of various subclass modules. These subclass modules provide the basic agent, while AMA supports a mixture of various basic services. AMA includes creation/deletion of shared video window and creation/deletion of shared window. For providing heterogeneous platforms with interoperability, it is necessary to share media data and to furnish awareness to the remote users involved in collaborative work. To solve the problem, we set the IA (Intelligent Agent) that modifies the transmitting packets by using TCP/ IP or UDP. Event messages including information about shared objects are bypassed among the homogeneous. SMA (Session Management Agent) controls the access to the whole session. This agent can be used in meeting, distance learning, playing games and development of any software. Session control also facilitates access and limits it to the whole session. ACA (Access Control Agent) controls the person who can talk, and the one who can change the information. The mechanism of floor control consists of brainstorming, priority, mediated, token-passing and time-out. MCA (Media Control Agent) support convenient application using DOORAE based on RCSM environment. Supplied services are the creation and deletion of the service object for media use, and media share between the remote users. This agent limits the service by hardware constraint.

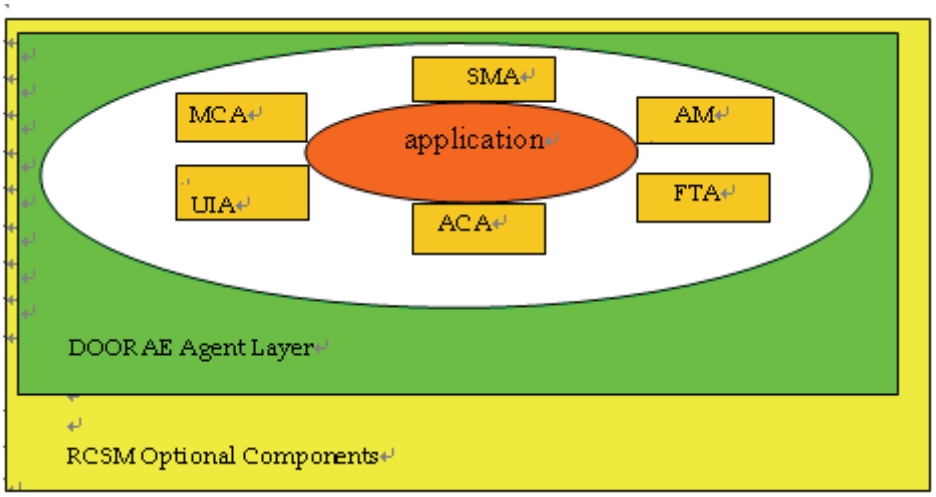


Fig. 3. The Organization of DOORAE

3.3 FTA based on RCSM Optional Components

FTA consists of EDA(Error Detection Agent) and ERA(Error Recovery Agent). EDA consists of ED(Error Detector), EC(Error Classifier), and EL(Error Learner). FTA is an agent which plays a role in detecting, classifying, and recovering errors. As shown in Figure 4, you can see the message flows in the organization of FTA.

ED is an agent which plays a role as an interface to interact among an application, EC, and EL. ED has functions which detect an error. ED informs EC of the results of detected errors. ED inspects applications by using process database function periodically to find an error. EC and EL deal with learning in reactive multi-agent systems. Generally, learning rules may be classified as supervised or unsupervised. In this paper, it uses a perception training. Hence, the training set consists of a set of input vectors, each with its desired target vector. Input vector components take on a continuous range of values; target vector components are binary valued (either zero or one). After training, the network accepts a set of continuous inputs and produces the desired binary valued outputs. KB has a registration information of creation of service handle and session manager handle by Daemon and GSM. EC can decide whether it is hardware error or software error based on learning rules by EL. In case of hardware error, it cannot be recoverable. In case of software error, it can be recoverable. This approach is based on the idea of comparing the expected error type which is generated by an EL with the actual error occurred from sites.

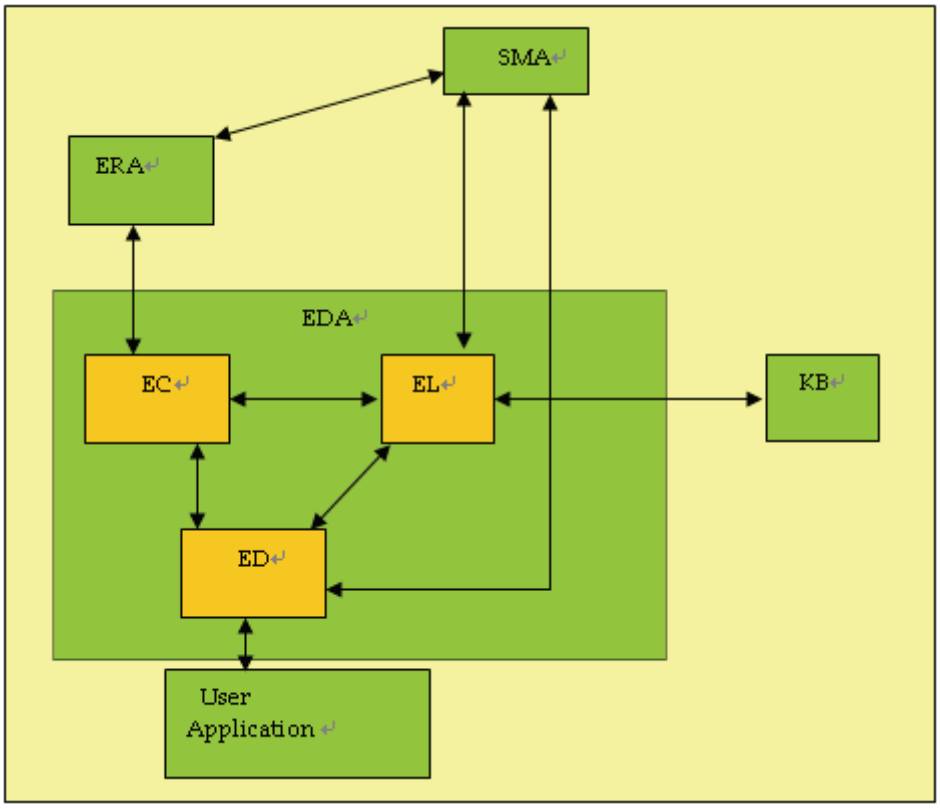


Fig. 4. The Organization of FTA based on RCSM Optional Components

Second, EC is an agent that plays a role as an interface to interact between ED for detection and ER for recovery such as Figure 5. EC and EL have functions which classify the type of errors by using learning rules. EC and EL deal with learning in reactive multi-agent systems. Generally learning rules may be classified as supervised or unsupervised or reinforcement learning. Reinforcement learning is similar to supervised learning, except it, instead of being provided with the concept output for each network input, the algorithm is only given a grade. The grade (or score) is a measure of the network performance over some sequence of inputs. This paper deals with Q-learning that is one of the reinforcement learning. Because EC and EL have not knowledge of error classification, it receives an acknowledge information which is necessary for fault diagnosis from PDB (Process Data Base). Hence the training set consists of a set of input vectors, each with its desired target vector. Input vector components take on a continuous range of values. Target vector components valued. After training, the network accepts a set of continuous inputs and produces EC and EL can decide whether it is hardware error or software error based on learning rules.

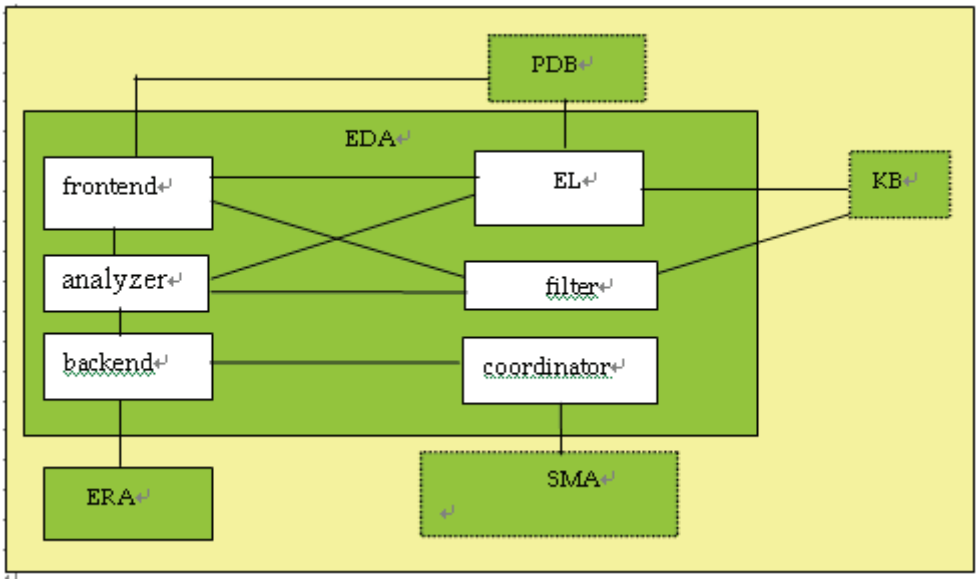


Fig. 5. The Organization of EC and EL

The scheme of classification mode is as follows. It is an ordered set. EC consists of frontend, backend, analyzer, coordinator, filter. Frontend has a function of playing a role in receiving error detection information from ED. Backend has a function of playing a role in receiving error recovery information from ERA. Coordinator informs SMA of the result. Analyzer has a function of classifying error's information that is received from frontend. Filter has a function of storing an error's history information in KB from error information that is classified by EL. EL has a function of classifying the type of errors by using learning rules with consideration of information from analyzer.

The sequence of Recovery's message flow can be shown in Figure 6. If an error is to be recovered, you can create sequences below. It creates a session with initial configuration information. It requests port ids for audio/video servers to build-up a Local Session Manager. It assigns port ids for audio/video servers of an application. It invites to the session and build-up a session instance monitor. It sends invited messages to start build-up of session instance monitor. It builds up Session Instance Monitor using the configuration information from LSM. It sends joint message to the Local Session Manager. It sends session information to Global Session Manager for set-up of GSM table. It begins a session. It exchanges message or command between LSM and PSM and media data between media server based on interpretation of message handler.

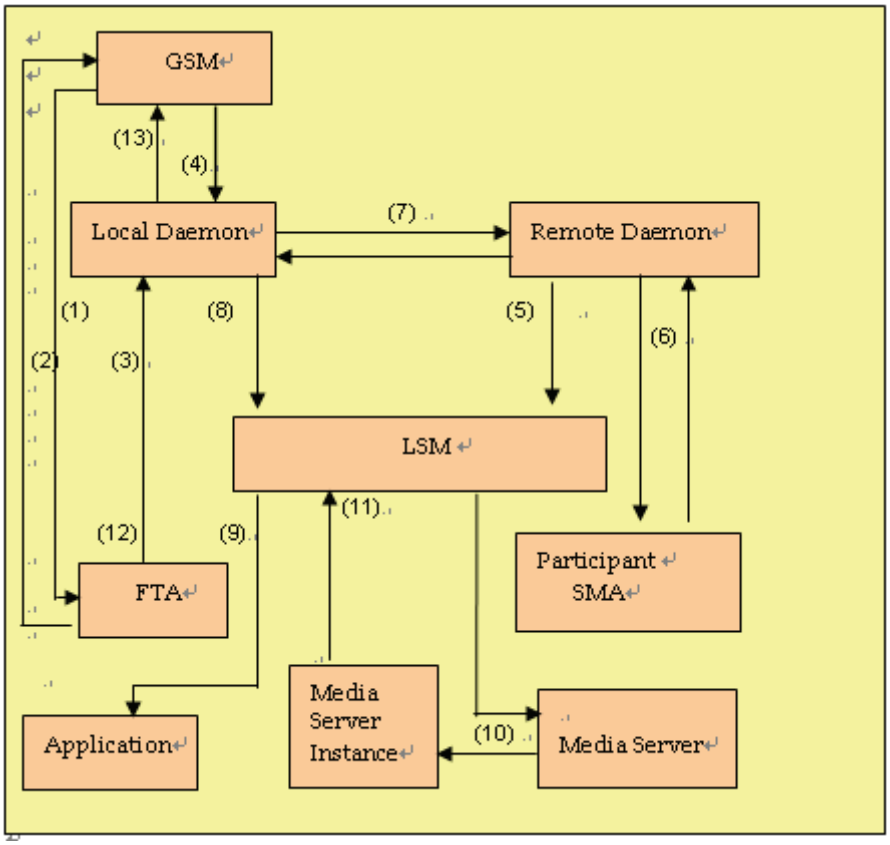


Fig. 6. Message Flow for Recovery

First it is decided whether it is hardware error or software error. In case of software error, it can be recoverable. If an error is to be recoverable, you can create sequences below. FTA requests to GSM session information. -GSM give response FTA session information. FTA requests to Daemon for recovery. Daemon announce to Remote-Daemon for recovery. Remote-Daemon announce to Participant Session Manager for recovery. Remote-Daemon receives an acknowledgement for recovery packet. Daemon receives an acknowledgement for recovery packet. Daemon creates Local Session Manager. Local Session Manager create Media server. Media server create Media server Instance. Media server Instance makes an acknowledgement to Local Session Manager. LSM creates application. Daemon informs SA-GSM of an information for recovery. The strong point of this system is to detect and recovered autonomously in case that the session's process come to an end from software error.

Our proposed FTA model aims at supporting adaptive QoS requirements defined in application-level Missions described by a set of Actions of objects by reserving, allocating, and reallocating necessary Resources given dynamically changing situations. A high-level FTA conceptual architecture to support adaptive QoS requirements is shown in Figure 7.

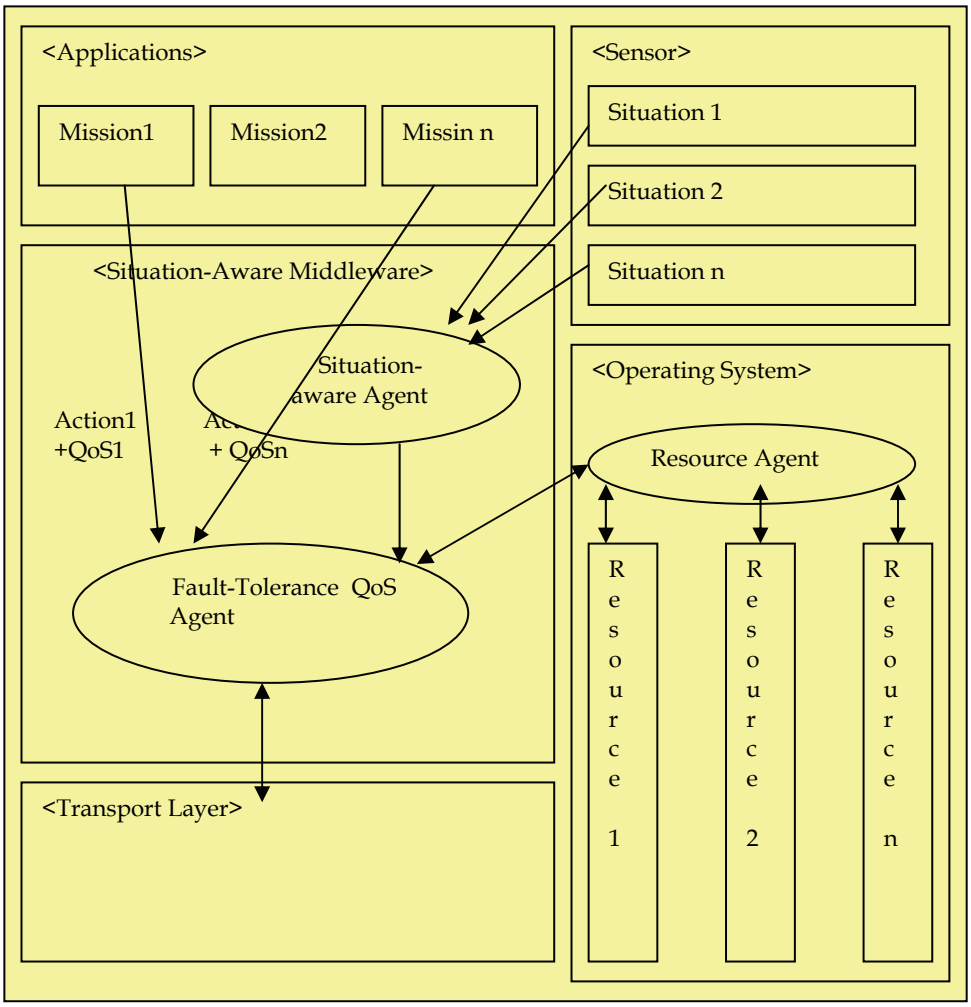


Fig. 7. Message Flow for QoS Architecture of RCSM

Situation-aware Manager, Resource Manager, and QoS Management Agent are the main components shown in Situation-Aware Middleware. Applications request to execute a set of missions to Situation-aware Middleware with various QoS requirements. A Situation-aware Manager analyzes and synthesizes context information (e.g., location, time, devices, temperature, pressure, etc.) captured by sensors over a period of time, and drives a situation. A Resource Manager simultaneously analyzes resource availability by dividing requested resources from missions (i.e., a set of object methods, called actions) by available resources. It is also responsible for monitoring, reserving, allocating and deallocating each resource. Given the driven situations, A QoS Management Agent controls resources when it met errors through the Resource Manager to guarantee requested QoS requirements. If there

is some error resource due to low resource availability, it performs QoS resource error detection-recovery. This system resolves the errors by recovering resources for supporting high priority missions. To effectively identify and resolve QoS conflicts, we need to capture the relationships between mission, actions, its related QoS requirements, and resources. In this paper, the FTA access requires situation-aware fault-tolerance QoS, in which the different fault-tolerance can be automatically enforced according to different situations such as wired or wireless network environment.

4. Conclusion

In this paper, we proposed a QoS resource error detection-recovery model called "FTA" for situation-aware middleware. An adaptive distance education system is used as an illustrative example of the FTA model and its resource error detection-recovery. The focus of situation-aware ubiquitous computing has increased lately. An example of situation-aware applications is a multimedia education system. The development of multimedia computers and communication techniques has made it possible for a mind to be transmitted from a teacher to a student in distance environment. This paper proposed an Intelligence Fault Tolerance Agent in situation-aware middleware framework model. FTA provided several functions and features capable of developing multimedia distant education system among students and teachers during lecture. FTA is a system that is suitable for detecting and recovering software error based on distributed multimedia education environment as FTA by using software techniques. This method detects an error by using process database. The purpose of this research is to return to a healthy state or at least an acceptable state for FTA session. It is to recover application software running on situation-aware ubiquitous computing automatically. When an error occurs, FTA inspects it by using API function for process database. If an error is found, FTA decides whether it is hardware error or software error. In case of software error, it can be recoverable. FTA informs Daemon and Session Manager of the fact. As they receive the information from the FTA, Daemon and Session Manager recovers from the error. The purpose of FTA system is to maintain and recover for FTA session automatically.
In the future work, fault-tolerance system will be generalized to be used in any environment, and we will progress the study of domino effect for distributed multimedia environment as an example of situation-aware applications.

5. References

C. W. Loftus, E. M. Sherratt, R. J. Gautier, A. M. Grandi, D. E. Price and M. D. Tedd (1995). Distributed Software Engineering-The Practitioner Series, *Prentice Hall Inc.*, Herfordshire, UK

Dhiraj K. Pradhan (1996). Fault-Tolerant Computer System Design, *Prentice Hall Ptr*

Francois Fluckiger, (1995). Understanding Networked Multimedia-Application and Technology Technology, *Prentice Hall Inc.*, Herfordshire, UK

Gil C. Park, Dae J. Hwang (1995). Design of a multimedia distance learning system: MIDAS, *Proceedings of the ISATED international conference*, Pittsburgh, USA

Gordon Blair, Jean-Bernard Stefani (1997). Open Distributed Processing and Multimedia, *Addison- Wesley*

Grudin, J. (1994). CSCW: History and focus, IEEE Computer 27(5), pp. 19-26

ITU-T Recommendation (1995). T.122 Multipoint Communication Service for Audiographics and Audiovisual Conferencing Service Definition, *ITU-T SG8*

Jae Y. Ahn, Gil C. Park and Dae J. Hwang (1996). A Dynamic Window Binding Mechanism for Seamless View Sharing in Multimedia Collaboration, *Proceedings of 14th ISATED International Conference*, Innsbruck, Austria

Jae Young Ahn, Gyu mahn Lee, Gil Chul Park, Dae Joon Hwang (1996). An implementation of Multimedia Distance Education System Based on Advanced Multi-point Communication Service Infrastructure: DOORAE, *In proceedings of the IASTED International Conference Parallel and Distributed Computing and Systems,* October 16-19, 1996, Chicago, Illinois, USA.

Jiten Rama, Judith Bishop (2006). Survey and Comparison of CSCW Groupware applications, *Proceedings of SAICSIT 2006,* pp. 507-512, South Africa

H. Zhang A. Kankanhalli, S. Smoliar (1995). Automatic Partitioning of Full-motion Video, A Guided Tour of Multimedia Systems and Applications, *IEEE Computing Society Press*

Man Seok Yang, Kim Dae Hyun, Lee Kyung Oh, Yoon Young Park (2009). A LEACH-based Clustering protocol for the Connection Persistent Improvement in Sensor Networks, *Proceedings of the 8th APIS*, pp. 507-512, ISSN 1976-7587, University of the Ryukyus, Jan. 11-12, 2009, APIS, Okinawa, Japan

Michael G.Moore, Greg Kearsley (1996). DISTANCE EDUCATION A System View, *An International Thomson Publishing Company*

Saha, D, Mukherjee, A. (2003). Pervasive computing: a paradigm for the 21st century, IEEE Computer, pp. 25-31, Volume: 36, Issue: 3, March 2003,

Satyanarayanan, M. (2001). Pervasive computing: vision and challenges, *IEEE Personal Communications*, pp. 10-17, IEEE, Volume: 8 Issue: 4, Aug. 2001

S. Yau, F. Karim, Y. Wang, B. Wang, and S. Gupta (2002). Reconfigurable Context-Sensitive Middleware for Pervasive Computing, *IEEE Pervasive Computing*, 1(3), pp. 33-40, July-September 2002

S. Yau and F. Karim (2002). Adaptive Middleware for Ubiquitous Computing Environments, Design and Analysis of Distributed Embedded Systems, *Proc. IFIP 17th WCC*, pp. 131-140, August 2002, Vol. 219.

T. Zhang and C.-C, J. Kuo (1999). Hierarchical Classification of Audio Data For Archiving and Retrieval, *ICASSP'1999*, Vol. 6, pp.3001-3004, Phoenix

Victor P. Nelson and Bill D. Carroll, Fault-Tolerant Computing, *IEEE Computer Society*, Order Number 677, Library of Congress Number 86-46205, IEEE Catalog Number EH0254-3, ISBN 0-8186-0677-0

Younglok Lee, Seungyoung Lee, Hyunghyo Lee (2006). Development of Secure Event Service for Ubiquitous Computing, Proceedings of ICIC 2006, pp. 83-94, ISBN 3-540-37255-5, August, 8, 2006, LNCIS 344, Springer, Kunming, China

3

Study on Data-driven Methods for Image and Video Understanding

Tatsuya Yamazaki
National Institute of Information and Communications Technology
Japan

1. Introduction

Owing to progress of broadband Internet communication infrastructure, people can enjoy multimedia services. Among the services, images and videos are making an appeal and people desire to find out what they want to see. But they do not always make a success of searching and it sometimes takes plenty of time to reach their targets. A solution is to attach information tags according to the image or video content. Since image and video submission into Internet is increasing day by day, manual tag attachment is almost impossible. Development of automatic tag attachment is an urgent theme for future Internet service.
Another imaging technology in the real world is environmental cameras. The environmental cameras mean the cameras set in the environment such as a ceiling or a street corner. One can easily imagine surveillance cameras as an example of the environmental cameras and sometimes they are considered to be identical. In this paper, I want to segregate them because the purpose of the environment cameras is not only to keep a watch on accidents or crimes but to understand peoples' situations and behaviours. Namely the surveillance cameras are a subset of the environmental cameras.
For both of the above cases, the key point is understanding image and video. It is one of the issues that have been discussed for a long time and related with the artificial intelligence technology. Therein image clustering and object extraction are most essential technologies.
Image clustering means to divide an image into several segmented regions in a way of unsupervised, which was used for object extraction (Meier & Ngan, 1998), image compression (Kunt, 1988), or image categorization. Since it is hard to say how many regions are included in an image in general, there are a few studies that estimated the number of regions. Usually the number of regions has been assumed to be known. As a previous research in which the number of region was estimated, Won and Derin (Won & Derin, 1992) proposed to use the Akaike Information Criterion to determine the number of regions. But these are model-based approaches, that is, how to select a suitable model is important.
Regarding with the environmental cameras, there have been previous works to extract moving object in order to capture human behaviours. Satake and Shakunaga (Satake & Shakunaga, 2004) proposed an appearance-based condensation tracker, which is composed of a condensation tracker and a sparse template matching method to detect the movement of people with a camera. The template-based condensation tracker is stabilized for tracking even in the case of object occlusion. Thonnat and Rota (Thonnat & Rota, 1999) used low-

level image processing techniques to detect and track mobile objects. Their aim was rather to understand images, namely, to generate alarms automatically for operators when interesting scenarios had been recognized by the system. In both of the above works, they used only one camera. On the contrary, Matsuyama (Matsuyama, 1999) proposed a protocol for negotiation among multiple environmental cameras. Because of the protocol, they could synchronize several cameras in real time and grab the human behaviours more smoothly and more seamlessly.
As I described in the above paragraphs, automatic processing of image and video media should be deployed more in future Internet services for the information explosion era. Technically adaptation of processing based on the observed data, which is called data-driven, is necessary. Therefore, in this paper, data-driven image clustering and object detection methods are proposed.

2. Data-driven Image Clustering

In this section, a data-driven clustering algorithm for colour images is proposed based on a multi-dimensional histogram. A statistical model is introduced and the image data is assumed to be derived from a mixture of multi-variant distributions.

2.1 Multi-dimensional histogram

Although the R-G-B colour space is assumed to make a discussion simple, the proposed method can be applied to other colour spaces. The R-G-B colour image data is represented as $\boldsymbol{y}=(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{Np})$, where $\boldsymbol{y}_i=(y_i^R, y_i^G, y_i^B)$ (i=1, … , N_p) is the observed element at the i-th pixel. N_p is the total number of pixels, y_i^X is a scalar value observed on the X plane at the i-th pixel, and y_i^X is assumed to range from G_{min} to G_{max}, where X is R, G, or B.
The multi-dimensional histogram is formed easily. First, distribute the observed data into the R-G-B color space. Second, construct a complete set of nonoverlapping intervals, called bins, by dividing the cube ($[G_{min}, G_{max}]^3$) equally with a width h. Finally, count the number of elements in each bin. Fig. 1 shows a construction of a multi-dimensional histogram with G_{min}=0. The histogram width h is an important parameter, and relates to the data distribution. When the data comes from a single density, several rules have been proposed to determine h. When the data are obtained a mixture of densities and the number of densities, N_c, is unknown, it is difficult to determine h. The multi-dimensional histogram of the colour image data deal with in this study corresponds to the latter case.

2.2 Data-driven clustering algorithm

It is assumed that there are a set of candidates for the histogram width, that is, $\{H=h_1, h_2, \ldots, h_C\}$. Here, a novel algorithm is proposed to determine h and N_c.

Step 1) Select a candidate h_j (j=1, … , C) from H. Construct a histogram with a width h_j. Select the bins that have at least one element in the histogram, and sort them in the order of the number of elements in each bin. The sorted bins are numbered in the order of the number of elements as $b_1^j, b_2^j, \ldots, b_{nz}^j$, where b_{nz}^j is the number of bins having at least one element. The k-th bin, b_k^j, has n_k^j elements (Fig. 2).

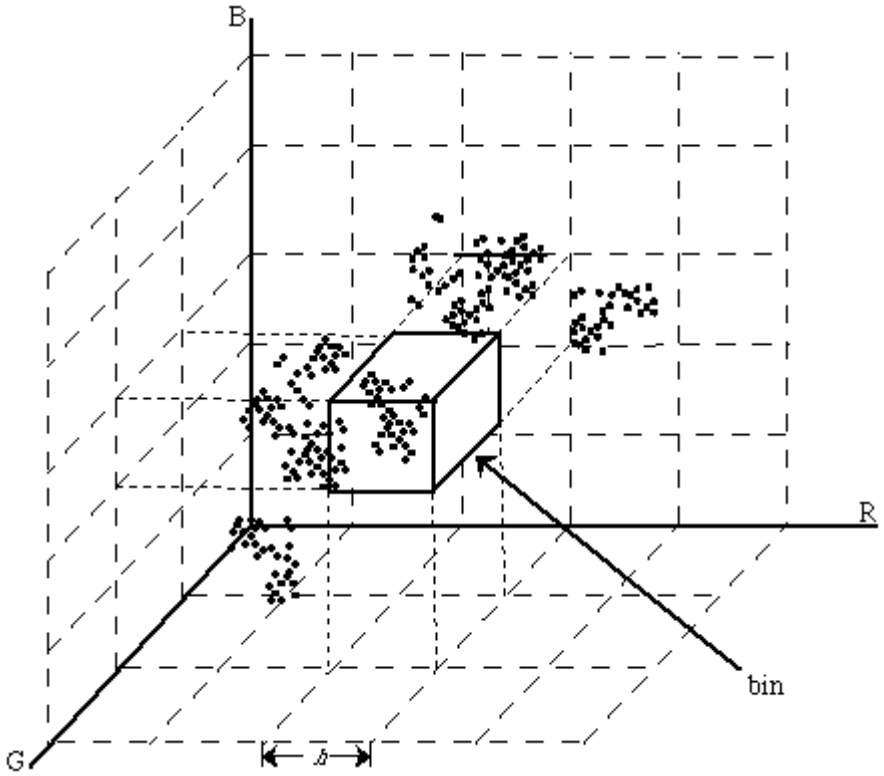


Fig. 1. Multi-dimensional histogram

Step 2) If $n_k^j < n_{cut}^j$, then b_k^j is removed. n_{cut}^j is a threshold calculated as

$$\mathrm{n_{cut}} = \alpha_{\mathrm{h_j}} \frac{\mathrm{b_{nz}}}{\mathrm{N_p}} \tag{1}$$

where $\alpha_{\mathrm{h_j}}$ is a control parameter to be determined for each h_j heuristically. n_{cut} is the threshold that divides explicitly insignificant bins. Eventually b_{cut}^j bins remain.

Step 3) Extract the top b_{sig}^j bins as significant bins from b_{cut}^j bins, where b_{sig}^j is determined as follows.

$$\mathrm{b_{sig}^j} = arg \max_{\mathrm{k=1,\ldots,b_{cut}^j}} \mathrm{Cr_{sig}}(k) \tag{2}$$

$$\mathrm{Cr_{sig}}(k) = \sum_{\mathrm{l=1}}^{\mathrm{k}} \frac{\sum_{\mathrm{m=1}}^{\mathrm{b_{cut}^j}} \mathrm{n_m^j}}{\mathrm{n_l^j}} - \frac{\mathrm{b_{cut}^j}}{\mathrm{k}} \tag{3}$$

$Cr_{sig}(k)$ is a criterion to select significant bins that include as many elements as possible considering suppression of the number of selected bins.

Step 4) If b_{sig}^j for every histogram-width candidate h_j is calculated, then go to Step 5. Otherwise, go to Step 1 and pick up another candidate whose b_{sig}^j has not been calculated.

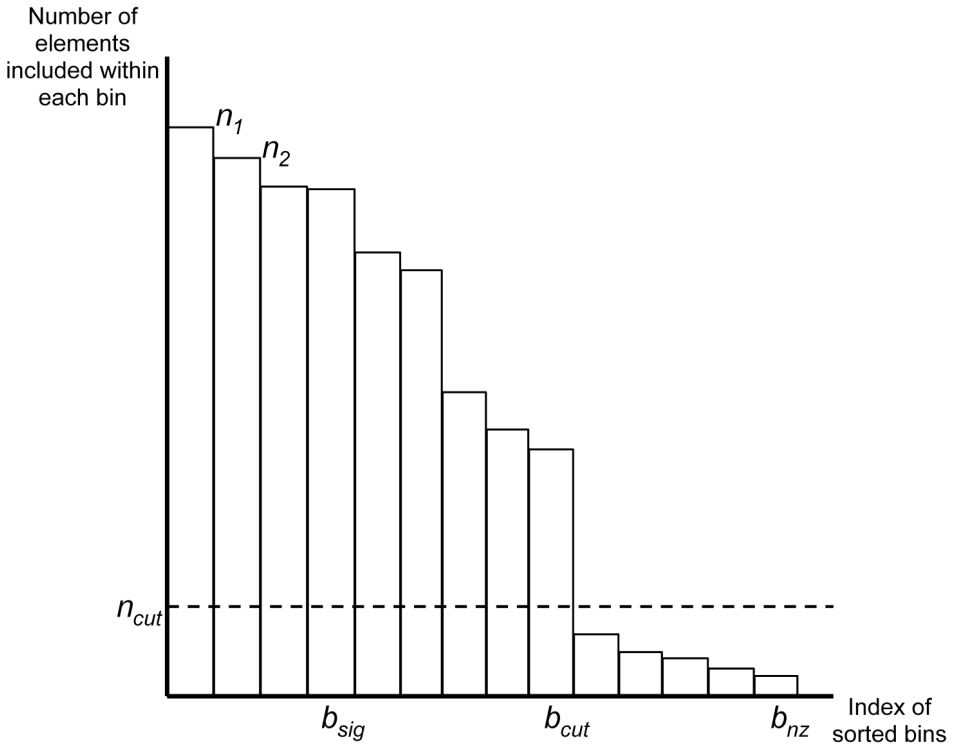


Fig. 2. Sorted frequency distribution of elements within each bin

Step 5) Calculate the optimal histogram width h^* as

$$h^* = arg \max_{h_j \in H} \sum_{l=1}^{b_{sig}^j} n_l^j \tag{4}$$

where b_{sig}^j corresponds to h_j, and b_{sig}^j corresponding to h^* is denoted as b_{sig}^*.
Consequently h^* is selected from H as the optimal width from the perspective of extracting as many significant elements as possible to compute the distribution statistics.

The significant b_{sig}^* bins are determined beforehand without considering the mutual relationships. Therefore, the adjacent bins that are supposed to belong to the same density must be merged to determine the final cluster count N_c. The clustering criterion is as follows. Calculate the average $\left(\mu_k^R, \mu_k^G, \mu_k^B\right)$ and the standard deviation $\left(\sigma_k^R, \sigma_k^G, \sigma_k^B\right)$ for the k-th bin (k=1, ..., b_{sig}^*).

Select any two bins, say b_l and b_m, from b_{sig}^* bins. If $\left|\mu_l^R - \mu_m^R\right| < \beta\overline{\sigma^R}$, $\left|\mu_l^G - \mu_m^G\right| < \beta\overline{\sigma^G}$ and $\left|\mu_l^B - \mu_m^B\right| < \beta\overline{\sigma^B}$, then the two bins belong to the same cluster, where $\overline{\sigma^X} = \sum_{i=1}^{b_{sig}^*} \sigma_i^X / b_{sig}^*$ for X=R, G, or, B and β is a parameter. Finally, cluster the b_{sig}^* bins into N_c clusters.

2.3 Proposed algorithm application to real data

The proposed algorithm was applied to real image data. Fig. 3 shows one of the original images, which is called the "Lady with a rose" image in this paper. The image size is 480 × 480. Although it is shown in grey scale, the original colour space is the RGB colour space. The histogram width candidates set used in the algorithm is H={4, 8, 16, 32}. The values of α_{hi} corresponding to each histogram width are α_4 =1.0, α_8=0.7, α_{16}=0.35 and α_{32}=0.05. In this experiment, β is set to 0.0; this means that the merging process is skipped.
Applied the proposed algorithm, the histogram width was determined to be 32 and the number of clusters was 6 finally. These values were determined in a data-driven way.
Then, the statistics of each cluster were computed by the minimum distance method, and the conventional maximum likelihood method with the estimated statistics, under the normal distribution assumption, was performed to obtain the segmented image. The final result followed by the 3×3 mode filtering operation is shown in Fig. 4.

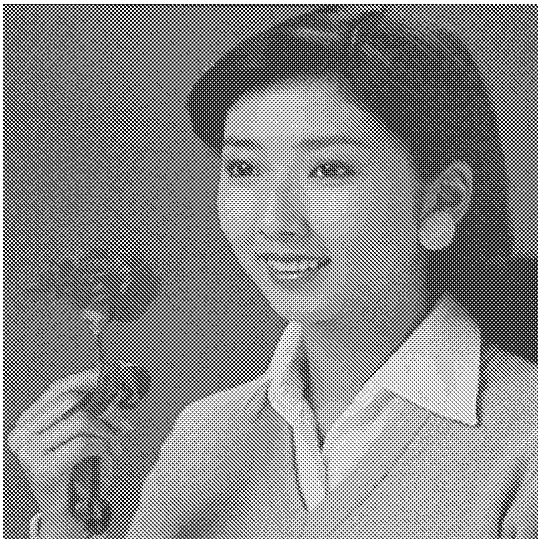


Fig. 3. Original image "Lady with a rose"

Fig. 4. Segmentation result of "Lady with a rose"

The final estimates of the means and the proportions are shown in Table 1.

	Cluster 1	Cluster 2	Cluster 3
R	37.4	17.7	73.5
G	45.7	12.1	84.0
B	52.1	6.6	104.5
Proportion	0.282	0.166	0.172
	Cluster 4	Cluster 5	Cluster 6
R	29.3	83.1	110.3
G	37.9	60.0	100.8
B	41.1	47.3	103.9
Proportion	0.132	0.157	0.091

Table 1. Final parameter estimates of the means and the proportions for "Lady with a rose"

3. Data-driven object detection

In the real world, cameras are becoming popular to detect and track moving objects not only for surveillance or security but also for digital signage or spoken dialogue systems. The background subtraction method, which can be used to detect objects moving in the foreground by determining the difference between the current frame and an image of the scene's static background, is still one of the useful methods to detect moving objects in video sequences. Although the background subtraction method is a simple and effective method to detect moving objects, it occasionally suffers from illumination changes and unexpected background changes such as shadow. To improve the original background subtraction method, we propose that knowledge application and data-driven parameter adaptation techniques be adopted.

3.1 Detection of People by Background Subtraction Method

In order to cope with the illumination changes, we adopt the normalized distance method. Here, a unit vector is defined as the projection onto the unit sphere of a vector whose elments are the intensity values of pixels in a target region. The normalized distance is defined as a distance between two unit vectors. Let $\boldsymbol{\tau}$ and $\boldsymbol{\beta}$ as vectors consiting of intensity values of pixels in an observed image and in a background image respectively. Then the distance $\boldsymbol{\delta}$ and normalized distance $\boldsymbol{\delta}'$ are shown as follows,

$$\boldsymbol{\delta} = |\boldsymbol{\tau} - \boldsymbol{\beta}| \tag{5}$$

$$\boldsymbol{\delta}' = \left| \frac{\boldsymbol{\tau}}{|\boldsymbol{\tau}|} - \frac{\boldsymbol{\beta}}{|\boldsymbol{\beta}|} \right| \tag{6}$$

Supposing that we process image frames during a period of T_{int}. We calculate $\boldsymbol{\delta}$ for each frame in T_{int}, then $Max(\boldsymbol{\delta})$, $Min(\boldsymbol{\delta})$, and $Ave(\boldsymbol{\delta})$ are calculated as maximum value, minimum value, and averaged value of $\boldsymbol{\delta}$. In the same way, $Max(\boldsymbol{\delta}')$ and $Min(\boldsymbol{\delta}')$ are calculated as maximum value and minimum value of the normalized distance $\boldsymbol{\delta}'$. Consequently, the following three discriminant functions are defined,

1. discriminant function for scene change
   $\max \boldsymbol{\delta} - \min \boldsymbol{\delta} > Th_{scene}$

2. discriminant function for background change
   Ave $\boldsymbol{\delta} > Th_{bs}$

3. discriminant function whether environment change or illumination change
   $\max \quad \boldsymbol{\delta}' - \min \quad \boldsymbol{\delta}' > Th_{ill}$

where Th_{scene}, Th_{bs}, and Th_{ill} are thresholds to be determined.
Using these discriminant functions, whether there is a moving object detection or a background updating in T_{int} can be judged as follows:

- If both (3) and (5) are true, there is a scene change by a moving object.
- If (3) is true and (5) is false, there is a scene change by an illumination change.
- If (3) is false and (4) is true, there is a background change.
- If both (3) and (4) are false, there is nothing. It is a normal background.

A challenging point in this method is adaptively setting the threshold value to differentiate foreground objects from the background image in spite of environmental changes.
To determine the threshold value, Wren et al. (Wren et al., 1997) modeled the background using a Gaussian distribution and estimated the parameters adaptively. Grimson et al. (Grimson et al., 1998) also set up parameters according to the statistical analysis of training samples of the background images. Stauffer and Grimson used a mixture model of Gaussian distributions of the images to cope with multimodal background distributions (Stauffer & Grimson, 1999).

3.2 Knowledge Application and Parameter Adaptation

We apply two techniques to the original background subtraction method in order to cope with unexpected moving objects and adaptive threshold parameter setting. Our aim is to detect and track people as moving objects. There are, however, other unexpected moving objects in the scene, such as an automatic door. To avoid detection of such an unexpected moving object, we introduce knowledge about special spots as the first technique. The positions of special spots are assumed to be known and masking is applied not to detect the unexpected moving object. This is a simple but effective technique.

To set the threshold values adaptively, we introduce a kind of steepest descent method as the second technique. The algorithm is depicted in Fig. 5.

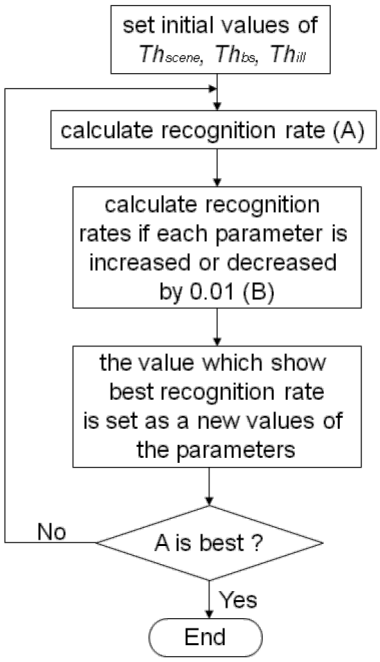


Fig. 5. Flowchart of threshold adaptation by steepest descent

In the first step of the algorithm shown in Fig. 5, initial values of Th_{scene}, Th_{bs} and Th_{ill} are set. Then the recognition rate A with the current threshold values that are the same as the initial values at the very first stage of the algorithm. After the threshold values are increased or decreased by a small value, the new recognition rates B are calculated. 0.01 is set as the small value in Fig. 5. B is a set of the recognition rates because increasing and decreasing of each threshold value are tried. A is compared with the best recognition rate in B. If A is superior to the best recognition rate, the algorithm is terminated. Otherwise, the best threshold values that correspond to the best recognition rate are substituted to the current threshold values and the same steps are carried out.

3.3 Experimental results at the real world test bed

We constructed such a test bed in the entrance of our research laboratory. In the ceiling of the test bed, there are five cameras, and the area covered by the cameras is the meshed region in Fig. 6. The camera can take a 768 × 494 pixel image and has remote pan, tilt, and zoom functions.

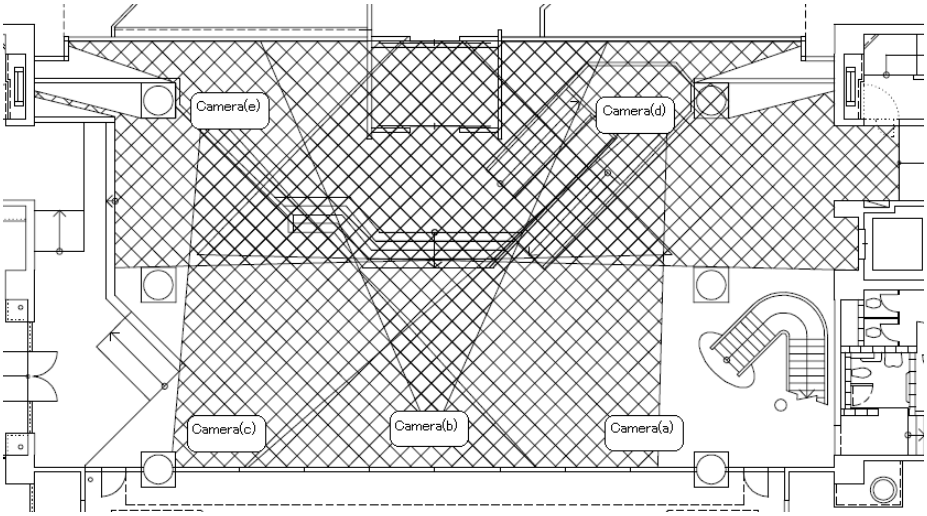


Fig. 6. Area monitored by cameras

In the test bed, we tried to detect moving objects using camera images and the background subtraction method. In the background subtraction method, first of all, we selected the initial image without any moving objects as the background image to be subtracted. Then, the difference between the current and the background images is calculated and pixels that have a difference larger than the threshold are registered as candidate pixels of an image of moving objects. This difference calculation is operated for each small image block of 80 × 60 pixels. The judgment described in the previous subsections is applied. The adjacent small image blocks of candidate pixels are merged into a larger image block. To track the moving objects, a two-dimensional histogram with hue and saturation values in an HSV color space is constructed to calculate the correspondence between objects in the current and previously captured images. When the difference between two images in the two-dimensional histogram is smaller, the probability that objects belong to the same object is higher.

We applied the above background subtraction method to a ten-second video captured in an actual situation. The number of frames captured from each camera was different because the time consumed for image compression was different.

Two experiments were carried out. The first experiment was moving object detection by the background subtraction method with knowledge application only. The second one was moving object detection by the background subtraction method with parameter adaptation as well as knowledge application. The first and the second are referred as Experiment I and Experiment II respectively. The applied knowledge is that the position of the automatic door is known.

In Experiment I, the threshold parameters were fixed as Th_{scene}=0.10, Th_{bs}=0.25 and Th_{ill}=0.05. One result of detecting a moving object is presented in Fig. 7. The people in the image

should be recognized as moving objects, and rectangles are drawn as a result. Two larger rectangles (shown in red) are hand-made markings that indicate correct answers. Several smaller rectangles in the left larger rectangle (shown in yellow) indicate detected results obtained by the background subtraction method. In the right larger rectangle, no object was detected by the background subtraction method. We call the larger rectangles ground truth rectangles and the smaller rectangles are called detected rectangles.

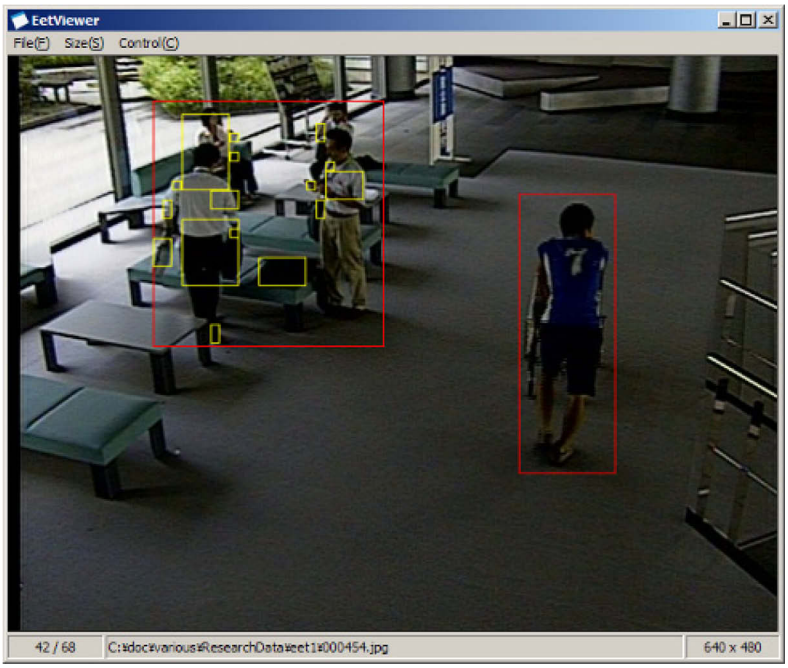


Fig. 7. Result of detecting moving objects in Experiment I

Although an identical match between ground truth and detected rectangles is desirable, detected rectangles are almost always included in or overlapped on ground truth rectangles. Here, we define two kinds of error: the type one error and the type two error. The type one error is that in which no detected rectangle is drawn where there was a ground truth rectangle. The type two error is that in which detected rectangles appeared where there was no ground truth rectangle. The total error rate can be calculated by averaging these two types of error. The rates of occurrence of two types of error and the total error rate are shown in Table 2 for cameras (a) - (e). The positions of the cameras are shown in Fig. 6.

	Type one error rate (%)	Type two error rate (%)	Total error rate (%)
Camera(a)	29.38	0.32	14.85
Camera(b)	54.35	27.94	41.15
Camera(c)	28.39	0.31	14.35
Camera(d)	10.23	16.19	13.21
Camera(e)	10.06	8.38	9.22

Table 2. Error rates with knowledge application and fixed parameters

Next, we applied threshold parameter adaptation presented in Fig. 5 as well as the knowledge application. This is Experiment II. The initial threshold parameters were set as Th_{scene}=0.10, Th_{bs}=0.25 and Th_{ill}=0.05. One moving object detection result with the parameter adaptation is presented in Fig. 8, that corresponds to Fig. 7. By comparing two images, it is found that the person in the right side was detected in Experiment II, who was missed in Experiment I. The rates of occurrence of two types of error and the total error rate are shown in Table 3.

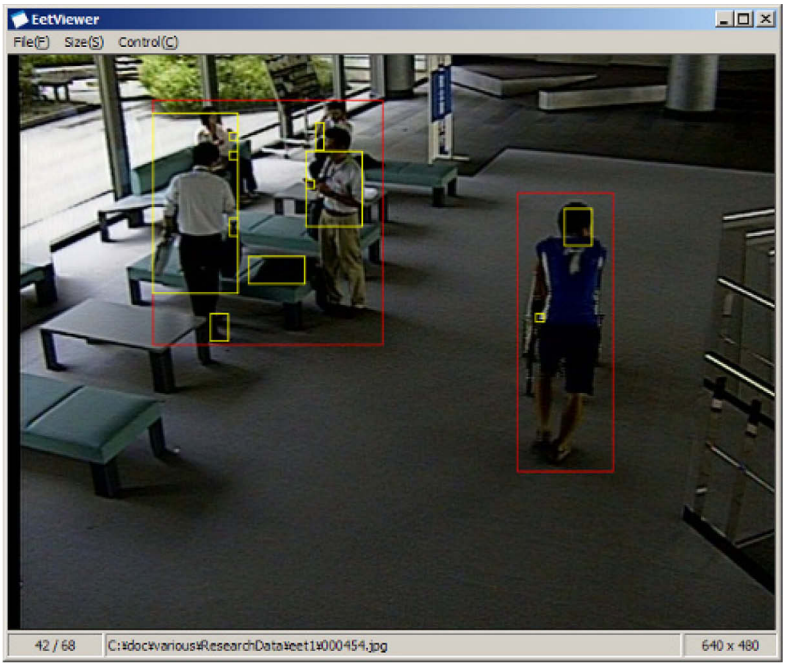


Fig. 8. Result of detecting moving objects in Experiment II

	Type one error rate (%)	Type two error rate (%)	Total error rate (%)
Camera(a)	14.69	0.27	7.48
Camera(b)	17.39	0.00	8.97
Camera(c)	19.36	1.97	10.67
Camera(d)	1.14	14.70	7.92
Camera(e)	10.06	0.08	5.07

Table 3. Error rates with knowledge application and parameter adaptation

Almost all error rates were improved. Especially improvement for camera (b) was splendid. It can be considered that the reason is owing to knowledge application. As the result of parameter adaptation, the final obtained parameters, Th_{scene}, Th_{bs} and Th_{ill}, are shown in Table 4.

	Th_{scene}	Th_{bs}	Th_{ill}
Camera(a)	0.10	0.21	0.04
Camera(b)	0.10	0.20	0.04
Camera(c)	0.10	0.21	0.04
Camera(d)	0.07	0.23	0.04
Camera(e)	0.12	0.26	0.04

Table 4. The final threshold parameters in Experiment II

4. Conclusion

Among multimedia, the roles of image and video media are becoming more important both in the cyber and real worlds. The cyber world means the information world structured by computer networks such as Internet. Videos and images are accumulated in the cyber world and the users are wandering to search for what they want. Also in the real world, cameras are becoming popular to collect the users' and environmental information. How to analyse these more efficiently is one of the issues to be solved urgently.

Image and video understanding has been studied and several approaches have been developed. In the parametric approaches, how to set the parameters is a challenging problem. In this paper, data-driven parameter adaptation was applied to image clustering and object extraction for still and video images. Although the experimental results were limited, definite improvement has been attained.

In the future, it is desired to extract contextual information from image and video media by applying image and video understanding techniques including the methods proposed in this paper. It must contribute to realize a personalized, adaptive, situation-aware service in a ubiquitous network society.

5. References

Meier, T. & Ngan, K.N. (1998). Automatic Segmentation of Moving Objects for Video Object Plane Generation. *IEEE Trans. on Circuits Syst., Video Technol.*, Vol. 8, No. 5, (Sept. 1998) pp. 525-538

Kunt, M. (1988). Progress in High Compression Image Coding. *Int. J. Pattern Recognition and Artificial Intelligence,* Vol. 2, No. 3, (1988) pp. 387-405

Won, C.S. & Derin, H. (1992). Unsupervised Segmentation of Noisy and Textured Images Using Markov Random Fields. *CVGIP: Graphical Models and Image Processing,* Vol. 54, No. 4, (July 1992) pp. 308-328

Satake, J. & Shakunaga, T. (2004). Multiple target tracking by appearance-based condensation tracker using structure information, *Proceedings of the 17th International Conference on Pattern Recognition (ICPR2004)*, Vol.1ap, No. We-ii, pp. 537-540, 2004

Thonnat, M. & Rota, N. (1999). Image Understanding for Visual Surveillance Applications, *Proceedings of Third International Workshop on Cooperative Distributed Vision (CDV-WD'99)*, No. 3, pp. 51-82, Nov. 2004

Matsuyama, T. (1999). Dynamic Memory: Architecture for Real Time Integration of Visual Perception, Camera Action, and Network Communication, *Proceedings of Third International Workshop on Cooperative Distributed Vision (CDV-WD'99)*, No. 1, pp. 1-30, Nov. 2004

Wren, C.; Azarbayejani, A.; Darrell, T. & Pentland, A. (1997). Real-time Tracking of the Human Body, *IEEE Trans. on Patt. Anal. and Machine Intell.,* Vol. 19, No.7, (1997) pp.780-785

Grimson, W.E.L.; Stauffer, C.; Romano, R. & Lee, L. (1998). Using adaptive tracking to classify and monitor activities in a site, *Proceedings of 1998 Conference on Computer Vision and Pattern Recognition (CVPR '98)*, pp. 22-29, 1998

Stauffer, C. & Grimson, W.E.L. (1999). Adaptive background mixture models for real-time tracking, *Proceedings of 1999 Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 246-252, 1999

4

Using the Flow of Story for Automatic Video Skimming

Songhao Zhu[1] and Yuncai Liu[2]
[1]play_tree@163.com, [2]whomliu@sjtu.edu.cn
Institute of Image Processing and Pattern Recognition,
Shanghai Jiao tong University
800, Don chuan Road, Shanghai 200240, China

Abstract: In this chapter, we present a novel scheme to incorporate the flow of story to select salient scenarios for generating semantically meaningful skimming of continuously recorded video such as a movie. We obtain clues of a movie from scenario editing rules and movie production techniques, which are commonly adopted in the process of video making to express the flow of the movie explicitly. The generated skimming not only provides a bird view of the original video but also help viewers understand the overall story. Furthermore, different types of video skimming can be appropriately generated with respect to different user requirements. Experimental results show that the proposed scheme is a feasible solution for the management of video repository and online review services.

1. Introduction

Nowadays the huge amount of video material stored in multimedia repositories makes the browsing, retrieval and delivery of such content a very slow and usually difficult task. According to a report from the China Central Television Web site, the total click count for movie review services reaches almost 100 million per day. In other words, movie occupies up to 50% compared to other genres of review services. This is largely due to the fact that it will take a viewer at least several minutes to watch a film episode before he can understand what happens and why it happens in the episode. In such instance, it is desirable to quickly browse video content in limited bandwidth, which can be achieved by automatic video content skimming.

Video skimming is defined as a sequence of moving images that compactly present the content of a video without losing the main information [1]. According to different characteristics of video content, we can categorize videos into two genres: event-oriented videos and story-oriented videos. An event-oriented video, such as sports or news, has a predetermined structure, and viewers can obtain entertainment or information even by only watching a few interesting parts. However, a story-oriented video, e.g., a movie, has no explicit structure, and a viewer needs to watch the whole video if he wants to know the story. Therefore, the skimming of an event-based video aims to compactly provide entertainment or information to viewers, whereas that of a story-oriented video aims to render the impression of the con-

tent of entire video in a short period of time. Several techniques have been proposed to generate skimming for event-oriented videos by exploiting the well-known structures of videos. In [2-4], a highlight of a news video can be a collection of anchor shots, while that of a soccer video can be generated as a collection of goal shots [5-6]. However, issues related to the production of an abstraction of a story-oriented video have not yet been widely studied.

One of the major challenges in generating skimming for a story-oriented video is that the cognitive process of viewers in perceiving the progress of a story while watching a video is not well understood. Most literatures [7-13] focus on the skimming production which presents the overall mood of the video, rather than help in perceiving the progress of the story. Sundaram *et al.* [14] first integrate viewers' comprehension of video contents into the generation of video skimming. More specifically, the video skimming is constructed by reducing scene duration, i.e., removing redundant shots and frames in scenes while preserving salient shots. To guide the reduction of scenes, a shot-utility function is utilized to model the degree of human understanding of a video shot with respect to its visual complexity and duration.

We propose a framework based on the progress of a story to perform the skimming of movie. As aforementioned, [14] solves the issue of video skimming at the scene level, while the proposed approach attempts to comprehend video contents from the progress and viewer's understanding of the overall story, which is realized as a sequence of scenarios and their relationships [15]. Considering that a scenario is usually captured by a scene, our approach can be considered as a higher level approach and a complement to Sundaram's scene-based scheme.

For the proposed framework based on the story progress, we first segment a video into shots based on the property of two-dimension histogram entropy of image pixels. Next, we generate semantically meaning scenarios by exploiting the spatio-temporal correlation among detected shots. Finally, we exploit general rules of special scenario and common techniques of movie production to grasp the flow of a story according to the degree of progress between scenarios to the overall story, which is evaluated in terms of the intensity of the interrelationships between scenarios.

To demonstrate the effectiveness of the proposed framework, we develop a video skimming system to conduct a set of comparison experiments using objective evaluation criteria, such as precision and recall. Furthermore, we also perform subjective tests to see the viewer's story understanding as well as to see the viewer's satisfaction. Results consistently show the advantages of the proposed framework. The basic idea of the proposed framework is shown in Figure 1.

The main contributions of the proposed framework are as follows.

- We propose a shot boundary detection approach by utilizing the property of two-dimension histogram entropy of image pixels;
- We generate scenarios of a video based on the spatio-temporal correlation between detected shots;
- We exploit clues about the progress of a story to implement the generation of video skimming.

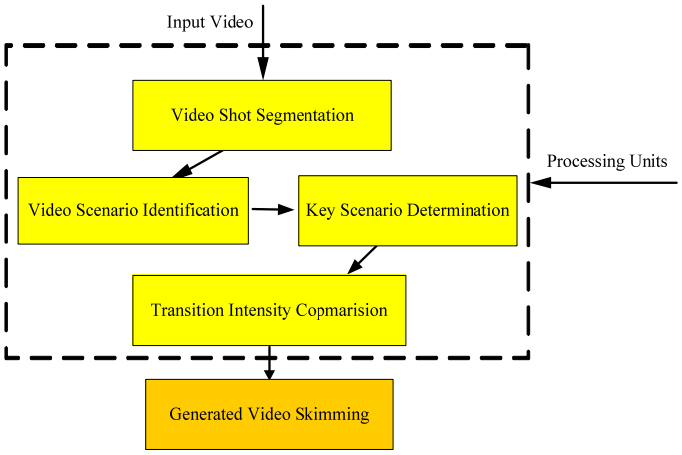


Fig. 1. Overview of the proposed framework.

The remainder of this chapter is organized as follows. In Sections 2 and 3, we introduce the process of temporal video segmentation and scenario detection, respectively. General rules of special scenario and common techniques of movie making are described in Section 4. Section 5 presents the story-oriented video skimming approach. Experimental results are provided in Section 6, followed by concluding remarks in Section 7.

2. Temporal Video Segmentation

In this section, we first depict the property of two-dimension entropy of image pixels, and then employ a progressive algorithm to detect shot boundaries.

2.1. Property of Two-Dimension Histogram Entropy

According to [16], two-dimensional histogram entropy of pixels can be used to describe the distribution of information of a grey-level image. Since the horizontal or vertical change of the gray value of pixels can not well represent the total characteristics of an image, we improve the construction of components of the two-dimensional histogram. More specifically, the gray value of pixel x and the average gray value of pixels in its 3×3 neighborhoods are replaced by the average gray value of pixels in its four neighborhoods a^x_1 and average gray value of other four pixels located in the corner of its 3×3 neighborhoods a^x_2, respectively. The formulation of the two-dimensional histogram entropy of an image I is:

$$\begin{cases} E = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} E_{ij}, \; E_{ij} = -N_{ij} * \ln N_{ij} \\ N_{ij} = H_{ij} \big/ H, \qquad H = \sum_{i=0}^{L\text{-}1} \sum_{j=0}^{L\text{-}1} H_{ij} \end{cases} \tag{1}$$

where H_{ij} presents the two-dimensional histogram of the two-dimensional channel (i, j), and L is set at 256 for gray pixel. Here, i and j represents $a^{x}{}_{1}$ and $a^{x}{}_{2}$, respectively.
Figure 2 lists some examples to show the relationship between the information contained in image and corresponding value of two-dimensional histogram entropy.
Based on our daily observation and movie editorial techniques, the information contained in images will usually change when there exists shot transition. In such instance, the issue of the detection of shot boundaries can be resolved by using the property of two-dimensional histogram entropy.

Fig. 2. Examples of image with more information having bigger two-dimension histogram entropy value.

2.2. Shot Boundary Detection

Our shot boundary detection algorithm consists of two levels: one is the coarse identification of candidate segmentation boundaries and the other is the refinement processing of candidate segmentation boundaries. One of the advantages of this progressive scheme is that it can speed up the overall processing with reduced number of video frames and effectively detect the gradual transitions (such as fade in/out, dissolve and wipe).

2.2.1. Coarse Boundary Identification

To coarsely identify candidate boundaries, we first perform temporal subsampling of the input video. In this chapter, the sub-sample rate is typically set to be 10 frames per second. Then, image space of each frame f in the temporally sub-sampled sequence is divided into five regions as shown in Figure 3 due to the following fact. In most cases, the difference between images can be determined according to the information of region ①.

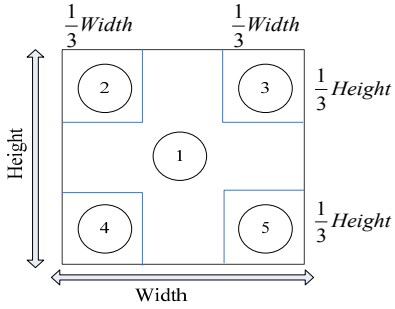


Fig. 3. Selected region for coarse candidate boundaries.

We now perform the coarse identification for candidate boundaries on the extracted main-image sequences in temporally sub-sampled sequence. Given the sequence of two-dimensional entropy difference between successive main-images, a set of candidate boundaries can be detected by appropriate thresholding. Instead of the global thresholding scheme in [17], we make use of the local thresholding approach presented by [18].

2.2.2. Exact location Determination

For those coarse candidate boundaries from the sub-sampled sequence, it is necessary to identify their exact locations at original temporal resolution. If the temporal sub-sampling rate is N (here $N=6$) and the frame number of candidate boundary is m, the precise position of shot transition will then fall into the localized neighbor region r centered on the candidate boundary m which contains frames from number $m*N-2*N$ to $m*N+2*N$. That is, the exact position of shot transition k is achieved by picking a minimum valley within the neighbor region. The overall coarse-to-fine refinement procedure under a certain sub-sample rate is shown in Figure 4.

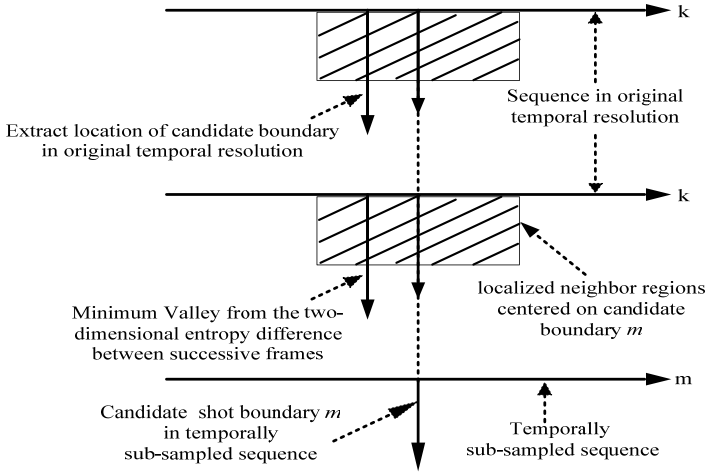


Fig. 4. Overview of the coarse-to-fine refinement procedure.

2.3. Shot Transition Verification

Since coarse candidate boundaries are detected using increased inter-frame difference in sub-sampling sequence, false detections maybe generated in the set of candidate boundary. Therefore, it is indispensable to perform verification of each identified shot transition frame k.

To robustly verify detected shot boundaries with respect to various transitions and special effects, we present a novel method to evaluate the similarity of visual content of frames within a local neighbor region r as shown in Figure 5 (a). More specifically, suppose the size of r is $2*N+1$. Then, feature set of left and right neighbor region, F_L and F_R, are:

$$\begin{cases} F_L = \left[F_{k-2*N}, \ldots\ldots, F_{k-1} \right] \\ F_R = \left[F_{k+1}, \ldots\ldots, F_{k+2*N} \right] \end{cases} \tag{2}$$

where F_i is a 5-dimension vector of 2-dimension entropy extracted from five regions as shown in Figure 3.

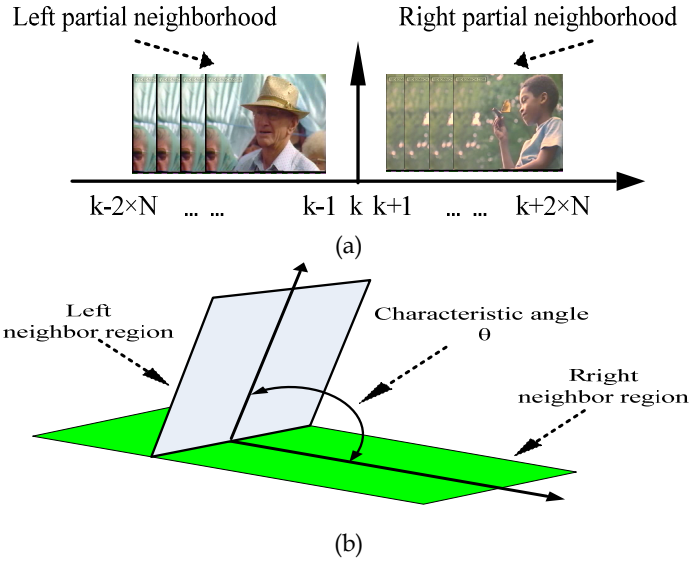
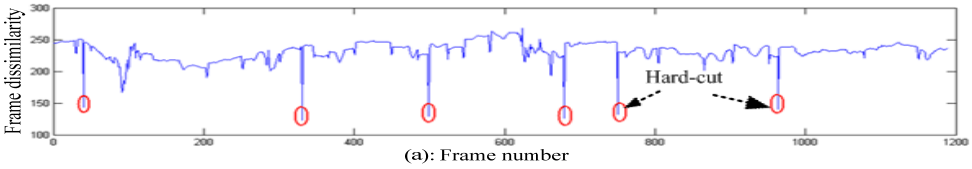


Fig. 5. Illustration of (a): local neighbor region centered on transition frame k and (b): Characteristic angle.

Principal component analysis is then utilized to preserve sub-images with more information for later computation of characteristic angle. P_L and P_R are corresponding orthonormal matrices of F_L and F_R by preserving n eigenvectors corresponding to first n largest eigenvalues. Next, singular value decomposition is performed on $P_L{}^T P_R$ to obtain the characteristic angles θ as in Figure 5 (b):

$$\begin{cases} \theta = \arccos(\delta) = \arccos\left[\max\left(\delta_1, \delta_2, ..., \delta_n\right)\right] \\ U^{-1}\left(P_L^T P_R\right)\left(V^T\right)^{-1} = \Sigma = Diag\left(\delta_1, \delta_2, ..., \delta_n\right) \end{cases} \tag{3}$$

Finally, any detected shot transition frame is considered as actual one only if its characteristic angles θ is bigger than a predefined threshold. Figure 6 shows two examples of shot detection results.



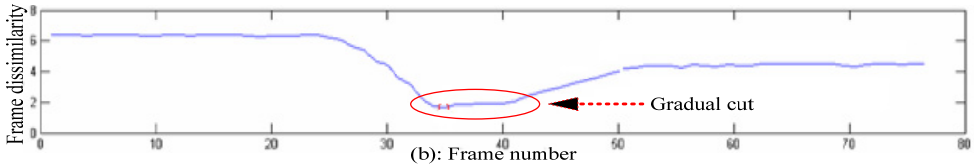


Fig. 6. Shot detection results for (a): abrupt transition and (b): gradual transition.

2.4. Key Frames Extraction within Shots

Representing the content of a video shot concisely is a necessary step for various video processing. In this chapter, a two-pass algorithm is used to complete the task of the selection of key-frames. Given a shot $Sh=\{f_1, f_2, \ldots\ldots, f_U\}$ with U frames, the process of key-frames choosing is then described as follows.

- First, f_1 is chosen as the first key-frame into the:

$$KFS = \{Kf_1\} = \{Kf_N\} = \{f_1\},\ N = 1 \tag{4}$$

where N is the total number of key-frame.

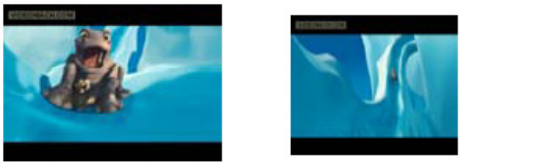
- For each frame f_m behind the current key-frame Kf_N, the value of two-dimension histogram intersection between f_m and each key-frame from $KFS=\{Kf_1, Kf_2, \ldots\ldots, Kf_N\}$ is then computed:

$$\begin{cases} FraSim(f_m, Kf_n) = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \min(H_{ij}^{f_m}, H_{ij}^{Kf_n}) \\ FraSim(f_m, KFS) = \{FraSim(f_m, Kf_n)\}, 1 \le n \le N \end{cases} \tag{5}$$

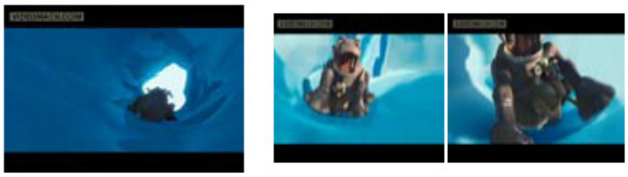
If values in $FraSim(f_m, KFS)$ are all greater than a pre-fixed threshold decided using the method in [17], then f_m is absorbed into the key-frame set KFS as a new key-frame:

$$KFS = \{Kf_1, Kf_2, \ldots\ldots, Kf_N, f_m\} = \{Kf_1, Kf_2, \ldots\ldots, Kf_N, Kf_{N+1}\} \tag{6}$$

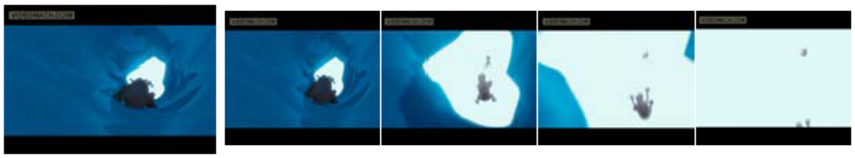
Figure 7 shows some examples of key frame extraction from different shots, where the most left image in each row is the detected shot followed by its key frame(s).



(a)



(b)



(c)

Fig. 7. Examples of key frame extraction with respect to visual content.

3. Scenario Boundary Detection

In this section, using the detected shots we construct a spatial correlation chain within certain temporal constraint, and then determine scenario boundary with the constructed spatio-temporal correlation chain.

3.1. Temporal Constraint Analysis

To avoid under-segmentation or over-segmentation of scenarios, we introduce the concept of temporal constraint similar to [20]. However, compared to [20], both the form and goal are different. Specifically, if the temporal spam between shots exceeds certain constraint, then they will not be grouped into the same scenario although their visual content is similar. For example, two shots in Figure 8 (a) are from different scenarios of movie 'X Man III' although there exists certain similarity between these two shots in the visual aspect. Two shots in Figure 8 (b) are another group of comparison shots, where the image information in terms of human understanding is also fully different.

(a) (b)

Fig. 8. Illustration the importance of temporal constraint. (a) and (b) are two groups of comparison shots both from different scenarios though they share certain visual information.

Temporal constraint (analysis window) depicts the largest number of shots contained in a scenario, which means only shots falling into the analysis window and satisfying spatial correlation can then be grouped into the same scenario.

The choice of the size of analysis window needs to be seriously considered due to its great impact on final video skimming. On the one hand, if this value is too large, shots from different scenarios may be clustered into the same one. That is, too large size will bring about the under-segmentation of scenario. On the other hand, if this value is too small, shots belonging to the same scenario may have different scenario labels. In this case, the issue of the over-segmentation of scenario will occur. Therefore, the size of analysis window is set to be fifty as default based on the heuristic rule. Final experimental results demonstrate that the best accuracy of scenario segmentation is obtained with this default value compared with other values.

3.2. Scenario boundary Identification

To accurately segment scenario, besides the information of visual aspect, the factor of temporal aspect is also taken into account. Different scenarios are obtained by the following passes.

(1) The similarity between two key frames p and q from different shots is measured by the Euclidean distance:

$$S_f(p,q) = \sqrt{\sum_{i \in allbins} \left(\left| F_p^i \right| - \left| F_q^i \right| \right)^2} \tag{7}$$

where F_p^i denotes the i^{th} two dimension entropy of frames p.

(2) The measurement of the dissimilarity between shot m with M key frames and shot n with N key frames is the average distance across all the possible pairs of key frames in each shot:

$$S_{shot}^{m,n} = \frac{\sum_{i \in M} \sum_{j \in N} S_f(i,j)}{M * N} \tag{8}$$

(3) Within one given analysis window Tw, the shot similarities between all pairs of shots are computed by:

$$S_{scenario}^{T_w} = S_{shot}^{m,n}, \ m \in T_w, \ n \in T_w \tag{9}$$

(4) For the current shot c within Tw, all the subsequent shots sharing similar visual content ($f_1, f_2 \ldots f_{s-1}$ and f_s) are:

$$S_{shot}^{c,k} < T_{scenario}, \ c < k \le T_w \tag{10}$$

where $T_{scenario}$ is the threshold. Furthermore, shot f_s is chosen as the current shot c for next processing.

(5) For shots between f_{s-1} and f_s, similar shots ($r_1, r_2 \ldots r_{t-1}$ and r_t) between shot f_s and T_w are located by:

$$S_{shot}^{c,k} < T_{scenario}, f_{s-1} < c < f_s, f_s < k \le T_w \tag{11}$$

Furthermore, shot r_t is also chosen as the current shot c for further processing.

(6) Repeat step 4 or / and 5 until the end of T_w, and there exists two types of scenario boundary. One is if the iterative process stops at step 4, then shot f_s is the current scenario boundary and shot $f_s + 1$ is the beginning shot for the next scenario. The other is if the iterative process stops at step 5, then shot r_t is the current scenario boundary and shot $r_t + 1$ is the beginning shot for the next scenario.

Figure 9 shows an example of the scenario boundary detection procedure for one testing video: interview video. Shot f_0 is first chosen as the current shot c, and shots f_1, f_2, f_3, and f_4 are visually similar shots according to inequality (7). Next, shot f_4 is selected as the current shot c, and shots r_1, r_2 and r_3 are visually similar shots satisfying inequality (8). Then, shot r_3 becomes the current shot c with respect to the scheme of scenario identification. Since there are no more shots meeting the conditions of inequality (8), shot r_3 is the last shot for current scenario X, and the procedure of the boundary detection for scenario X ends. That is, scenario X is composed of all of the shots from shot f_0 to shot r_3.

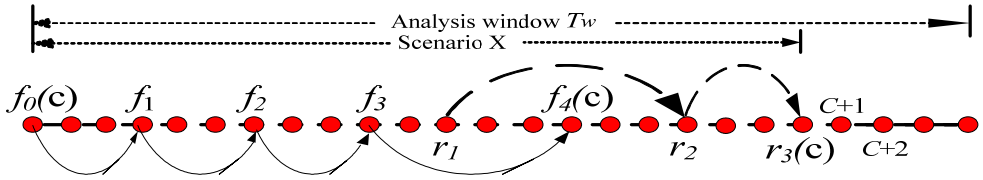


Fig. 9. Procedure of spatio-temporal coherence clustering for current scenario X. Red circles denotes shots within one given analysis Tw. Shots (f_0, f_1, f_2, f_3, f_4, and f_s) linked by solid line are similar ones generated by inequality (8). Shots (r_1, r_2, rand r_3) linked by dotted line are similar ones satisfying inequality (9). Scenario X is composed of all of the shots from shot f_0 to shot r_3.

3.3. Key Frames Extraction within Scenarios

Just like the purpose of extracting key frames from a video shot, the main content of a video scenario can also be concisely represented by corresponding key frames. Key frames of a scenario are chosen from the key frames of each shot within the scenario using the same method as discussed in subsection 2.4.

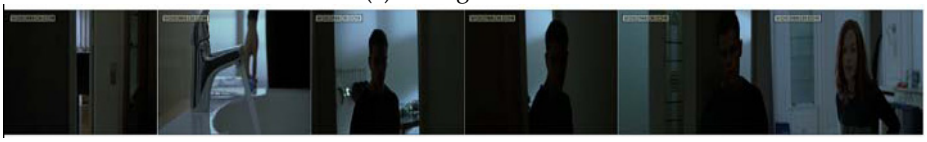
4. Scenario editing rules and movie making techniques

To construct a skimming for story-oriented videos (such as movies), it is necessary for viewers to grasp the flow of the story from scenario editing rules. According to [21], these rules can effectively describe the semantics of a scenario and so are commonly utilized in the movie production process. Moreover, story progress can be obtained from movie making techniques that are frequently adopted in movie making procedure to articulate a story. Next, we will explore the form of movie from these rules and techniques to introduce common types of key scenario and metrics of the story progress.

Based on our observation and the scenario editing rules, there exist several types of scenarios named key scenarios. These key scenarios often present vital information and hence always attract viewers' attention:

1) Dialog scenario: see Figure 10 (a); 2) Suspense scenario: see (b); 3) Action scenario: see (c).

(a) Dialog scenario



(b) Suspense scenario

(c) Action scenario

Fig. 10. Examples of different types of scenarios.

In [22], a film consisting of scenarios is regarded as a system and the flow of a film is depicted as the interrelationships between key scenarios. Generally, the interrelationship between scenarios is often captured by different types of transitions including temporal transition, spatial transition and rhythmic transition. From the point of view of film editing techniques and human understanding, the more a story progress between scenarios, the more the transition intensity is. Therefore, the intensity of scenario transition is considered as an important metric for evaluating the flow of a film as well as an important basic for the production of video skimming.

5. Video Skimming Approach

This section details the proposed story-oriented video skimming approach, which aims to enable viewers to grasp main content of original video by watching the generated short skimming. Thus, final video skimming should comprise those pairs of video segments with more contributes to the flow of a story. At the same time, the total duration of selected segments should satisfy the user-defined target duration.

The structure of this section is organized as follows. Related symbols are first introduced. Next, key scenarios are identified using visual and audio features. Then, an Intensity of Flow (IoF) function is defined to depict the transition intensity between key scenarios to the whole story. Finally, a Story Structure Tree (SST) is adopted to describe the interrelationship between scenarios in a story-oriented video, and the proposed skimming approach in subsequent sections is described with the SST.

5.1. Related Symbols

Table 1 lists the symbols used in the proposed story-oriented video skimming approach.

Symbol	Meaning
V	A story-oriented video
Sh_i	*i-th* shot in V
Ks_j	*j-th* key scenario in V
$ShKs(Sh_i,Ks_j)$	*i-th* shot of *j-th* key scenario in V
$N(Sh_i,Ks_j,Kf_s)$	Number of characters of *s-th* key frame of *i-th* shot from *j-th* key scenario in V
$TI(Ks_i, Ks_j)$	Transition intensity between key scenario *i* and *j*

Table 1. Related symbols.

5.2. Key Scenarios Identification

According to the discussion is Section 4, a story-oriented video is composed of a sequence of scenarios including key scenarios as shown in Figure 10 and remaining scenarios. Recall

that our purpose in this chapter is to achieve a highly compact skimming of a long video. Therefore, finally generated skimming contains only key scenarios while omitting remaining scenarios. Next, we will detail the process of key scenarios identification.
Based on our observation and scenario editing rules, typical characteristics of three kinds of key scenarios discussed in Section 4 are described as below.

- Dialogue scenario: faces with similar spatial position and similar size, a sequence of shots with low activity intensity, and strong similarity between shots.
- Action scenario: a set of shots with short duration, and intensive activity or audio energy.
- Suspense scenario: a set of shots with low average intensity distribution, a long period of low audio energy, and low activity intensity followed by a sudden change either in sound track or in activity intensity or both.

Next, features from different fields are first introduced and then heuristic rules are proposed to determine these three key scenarios.

5.2.1. Audio Features Selection

In order to achieve the accuracy of the scenario classification, audio information is an important and necessary clue. Because feature extraction is very important for audio content analysis, the extracted features should capture the temporal and spectral structure of different audio classes from each scenario talked above. Here, following features are selected to complete the task of audio clip classification, such as zero-crossing rate, energy envelope, spectrum flux, band periodicity, mel-frequency cepstral coefficients, spectral power, and linear prediction based cepstral coefficients.
Furthermore, in our experiment, all audio clips are divided into non-overlapping sub-clips. A sub-clip is of one second duration and is further divided into forty twenty-five millisecond-long frames, and short-time energy envelope entropy. The classification is performed based on these one-second sub-clips.

- **Zero-crossing Rate**. Zero-crossing rate is a simple measure of the audio signal frequency content. The N-length short-time zero-crossing rate of the n^{th} audio frame $s(n)$ is defined as:

$$ZCR(n) = \frac{1}{N} * \sum_{t=n-N+1}^{n} \frac{|\operatorname{sgn}\{s(t)\} - \operatorname{sgn}\{s(t-1)\}|}{2} w(n-t) \tag{12}$$

where $w(n)$ is a rectangular window, and

$$\operatorname{sgn}\left[s(n)\right] = \begin{cases} 1 & s(n) \geq 0 \\ -1 & s(n) < 0 \end{cases} \tag{13}$$

- **Energy Envelope.** Energy envelope is used to calculate the global temporal information. It can be computed in following way: third order Butterworth low-pass filtering of the analytical signal root mean square amplitude of each audio frame:

$$EE(n) = \sqrt{\frac{1}{N} \sum_{t=1}^{N} \left[s_t(n)\right]^2} \tag{14}$$

where N is the number of sample points in the n^{th} audio frame.

- **Spectrum Flux**. Spectrum flux is defined as the two-norm of the frame-to-frame spectrum amplitude difference vector:

$$SF(n) = \left\| \left|M_f(n)\right| - \left|M_f(n+1)\right| \right\| \tag{15}$$

where $|M_f(n)|$ is the magnitude of the FFT of the n^{th} frame at frequency value f. Both magnitude vectors are normalized in energy. Spectrum flux is a measure of spectral change between the adjacent two frames.

- **Band periodicity**. Band periodicity describes the property of each sub-band. In this chapter, we consider four sub-bands including 500~1000Hz, 1000~2000Hz, 2000~3000Hz, and 3000~4000Hz respectively. The periodicity property can be represented by the maximum local peak of the normalized correlation function.
- **Mel-Frequency Cepstral Coefficients (MFCCs)**. It has been proved that the mel-frequency cepstrum is a useful and highly effective feature in modeling the subjective pitch and frequency content of audio signals. The MFCCs are computed from fast Fourier transformation:

$$\begin{cases} MFCC(n) = \sqrt{\frac{2}{J}} \sum_{j=1}^{J} \left\{ \left(\log S_j\right) \times \cos\left[t\left(j-0.5\right)\frac{\pi}{J} \right] \right\} \\ t = 1, 2, ..., T \end{cases} \tag{16}$$

where the parameter J denotes the number of band-pass filters, and the parameter T means the order of the cepstrum. In our scheme, J and T are set to be 24 and 12 respectively, namely the 24 band-pass filters and 12-order MFCCs are used.

- **Spectral Power**. Spectral power of each audio frame is computed with a Hanning window $h(n)$:

$$h(n) = \sqrt{\frac{2}{3}} \times [1 - \cos(2\pi \frac{n}{N})] \tag{17}$$

The spectral power of the n^{th} audio frame $s(n)$ is:

$$SP(k) = 10\log_{10}\left[\frac{1}{N} \left\| \sum_{n=0}^{N-1} s(n)H(n)\exp(-j2\pi\frac{nk}{N}) \right\|^2 \right] \tag{18}$$

- **Linear prediction based cepstral coefficients (LPCCs)**. LPCCs are utilized to represent the timbre information of the voice components. The basic idea behind linear predictive analysis is that an audio frame can be approximated as a linear combination of past audio frames. By minimizing the sum of the squared differences over a finite interval between the actual audio frames and the linear predictive ones, a unique set of predictor coefficients can be determined.

5.2.2. Audio signal Classification

After removing silent clips, audio clips are classified into two categories firstly, i.e. speech and non-speech, based on the information of zero-crossing rate, energy envelope, and spectrum flux. Then, speech clips are classified into emotional voice and common voice based on the spectrum flux, band periodicity, mel-frequency cepstral coefficients, and non-speech

clips are classified into special sound and music based on the spectral power, and Linear prediction based cepstral coefficients. Finally, special sound is further classified into several classes, including gunshot, explosion, scream, beating, crashing of glass, and rubbing of tires using hidden markov models (HMM). Figure 11 illustrates the classification scheme.

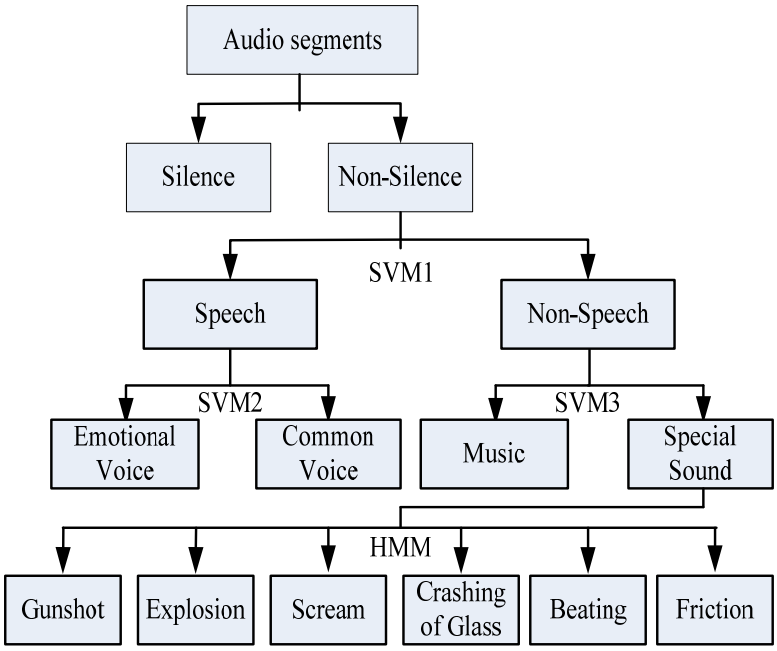


Fig. 11. Audio Classification scheme.

For support vector machines-based classification, features are first combined to construct a feature vector. Then, the mean and standard deviation of these feature vectors over all forty audio frames are calculated, and these statistics compose another feature vector. Finally, this new feature vector is normalized by its standard deviation of training data. The normalized feature vector is considered as the final representation of one-second audio signal.
The information of timbre and rhythm is utilized in the generative model for recognition, namely hidden markov models. Timbre allows one to tell the difference between sounds at the same loudness made by different objects. Each kind of timbre is denoted by one state of HMM, and represented with the Gaussian mixture density. Here, rhythm is adopted to represent the change pattern of timbres, and is denoted by transition and duration parameters in HMM.

5.2.3. Visual Features Selection

- **Face Information**. The occurrence of face is a salient feature in video, as it means the present of human in the scene. The size of a face is also a hint for the role of the person, i.e., a large face denotes that this person is in the center of attention. In the experiment, we use the face detection algorithm proposed by Li et al. [23] which performs reasonably good

for faces with different scales in the video. As a result of the face feature extraction process, we obtain the position and size of each detected face, and the number of hits.

- **Illumination Intensity**. Reference [24] indicates "the amount and distribution of light in relation to shadow and darkness and the relative tonal value of the scene is a primary visual means of setting mood." The amount of light in scene *Sc(k)*, here, is described as the average illumination intensity:

$$\begin{cases} II(k) = \dfrac{\sum_i ShAvgInt(i) \times ShLen(i)}{N(k)} \\ ShAvgInt(i) = \dfrac{\sum_j KfInt(j)}{N(i)} \end{cases} \tag{19}$$

where *ShAvgInt(i)* is the average illumination intensity of the entire key frames in the i^{th} shot, *ShLen(i)* is the length of the i^{th} shot in terms of frames, and *N(k)* is the total number of frames within the k^{th} scene. Furthermore, *KfInt(j)* is the illumination intensity of the j^{th} key frames in the i^{th} shot, and *N(i)* is the total number of frames within the i^{th} shot.

- **Activity Intensity**. The activity intensity indicates the tempo in video. For example, in conversational scenario, the activity intensity is relatively low. On the other hand, in action scenario, the activity intensity is relatively high. The activity intensity of the k^{th} scene is:

$$AI(k) = \frac{1}{N_k - 1} \sum_{l=1}^{N_{k-1}} \min \left[\sum_{i=0}^{i=L-1} \sum_{j=0}^{j=L-1} \left(H_{ij}^{l}, H_{ij}^{l+1} \right) \right] \tag{20}$$

- **Average Duration**. Similar to activity intensity, average duration in terms of frames is another measurement of the video tempo and is computed as follows:

$$AD(k) = \frac{1}{N_k} \sum_{l=1}^{N_k} ShLen(l) \tag{21}$$

5.2.4. Dialogue Scenario Determination

- **Dialogue Scenario Determination.** To locate the dialogue scenarios, we first adopt the information of face to detect shot sequence with alternately recurring visual contents, which include similar size, similar position and same number. Figure 12 shows an example of dialogue-like scenario.

Fig. 12. A dialogue-like scene detected using face information.

Given these dialogue-like scenes, we exploit their corresponding audio information to further make sure they are actual conversation contents. Specifically, we should differentiate speech signals from music and other sounds. Here, a simple classification method is utilized to complete the task of discrimination using zero-crossing rate, energy envelope, and spectrum flux as shown in Figure 11.

- **Emotional Dialogue Scenario Determination.** Among many dialogues, emotional conversations often attract viewers' attention and effect upon the flow of story. To discriminate the emotional contents from common ones, two acoustic features including average pitch frequency and temporal intensity variations are used. The first feature is estimated by 12-order linear prediction based cepstral coefficients, and the second one is represented by the variance of spectral power levels over all forty signal segments within one-second audio sub-clip.

5.2.5. Active Scenario Discrimination

- **Active Scenario Discrimination.** Similar to dialogue scenario, active scenario is another conceptually meaningful story content. Among active scenarios, gunfight scenario, beating scenario, and chasing scenario are often the most interesting events and can instantly attract viewers' attention in films. Therefore, based on the recognition of active scenario by integrating audio-visual signatures, three specific and distinct events are identified one-by-one.

According to the characteristics of active scenario as talked above; these scenes with average duration less than twenty-five frames and average activity intensity are identified as active scenarios.

Among active scenarios, gunfight, beating and chasing events are three important types and mostly the climaxes in feature films. Next, we will detect these important active scenarios using their unique audio-visual signatures.

- **Gunfight Scenario Discrimination.** Gunfire, explosion, and bleeding are the most typical visual features of gunfight scenarios as shown in Figure 13.

(a) (b) (c)

Fig. 13. The most typical visual features of gunfight scenario including (a) gunfire; (b) explosion; (c) bleeding.

Compared to gunfire cases, flames from an explosion show longer duration and cover wider areas. However, flames from an explosion and gunfire both have dominant yellow, orange and/or red color histogram. Hence, a predefined color table containing a certain range of color values is adopted to identify the gunfight-like scenario.

Since some violent actions, such as beating, gunshot and explosion can result in bleeding, bleeding is considered as another violence-related visual feature of gunfight event. We detect bloody color pixels using simple pixel-matching with the predefined color table.

Since other events may have similar visual features as gunfire, explosion and bleeding, the audio information provides a supplement to the detection of gunfight scenario. A distinct feature of gunfight scenarios is the unique sound track. Specifically, given the audio track for successive gunfight-like shots, we discriminate its class based on a hidden markov model. The likelihood ratio between the input audio track and the defined sound classes is calculated to determine which class the associated sound belongs to as shown in Figure 11.

- **Beating or Chasing Scenario Identification.** In general, beating or chasing events are inherently accompanied by unique sound (e.g., beating, rubbing of tires, etc.). In particular, for active scenario, we identify its specific class (beating or chasing) based on the likelihood ratio between its audio track and the given sound classes as shown in Figure 11.

5.2.6. Suspense Scenario Detection

- **Suspense Scenario Detection**. Suspense scenarios are often the most events and instantly attract a viewer's attention in horror and detective genres. According to the unique characteristics of suspense scenario as talked about in the introduction of Section five, a scenario can be declared as a suspense scenario if following criterions are satisfied simultaneously:

(1). Average illumination intensity is less than 50;
(2). There exist shots with audio energy envelope change suddenly from 5 to over 50;
or / both
There exist shots with activity energy change instantly from 5 to over 100.

5.3. Scenario Transition Form

As aforementioned, there exist three forms of scenario transition: temporal transition, spatial transition and rhythmic transition. Next, we formulate each of them.

- Generally speaking, viewers can understand the temporal transition with respect to the different number of characters appearing in two scenarios *k(i)* and *k(j)* with respect to the change of number of face:

$$TT(k_i,k_j)=\left|\sum_{s=1}^{S}C(s,m,k_i)-\sum_{t=1}^{T}C(t,n,k_j)\right| \tag{22}$$

where $S(m,k_i)$ is the last shot of *k(i)* and $S(n,k_j)$ the first shot *k(j)*. Temporal transition between two scenarios is discriminated f the following inequality is true:

$$TT(k_i,k_j)>\frac{1}{S+T}\left[\sum_{s=1}^{S}C(s,m,k_i)+\sum_{t=1}^{T}C(t,n,k_j)\right] \tag{23}$$

where *S* and *T* are the number of key frames in $S(m,k_i)$ and $S(n,k_j)$, respectively.

- Spatial transition depicts the change of positions in successive scenarios for the same characters, which can be determined in terms of the change color information of background regions. The background regions are obtained by excluding the face region of each character, which is feasible due to most characters are shown in close-up views in a film. The intensity of spatial transition between two scenarios *k(i)* and *k(j)* is formulated as below:

$$\begin{aligned} ST(k_i,k_j) = &\left|\frac{1}{S'}\sum_{s=1}^{S'}RA(s)-\frac{1}{T'}\sum_{t=1}^{T'}RA(t)\right|+\left|\frac{1}{S'}\sum_{s=1}^{S'}GA(s)-\frac{1}{T'}\sum_{t=1}^{T'}GA(t)\right| \\ &+\left|\frac{1}{S'}\sum_{s=1}^{S'}BA(s)-\frac{1}{T'}\sum_{t=1}^{T'}BA(t)\right|+\left|\frac{1}{S'}\sum_{s=1}^{S'}LA(s)-\frac{1}{T'}\sum_{t=1}^{T'}LA(t)\right| \end{aligned} \tag{24}$$

where *RA(s)*, *GA(s)*, *BA(s)*, and *LA(s)* are average red, green, blue, and luminance values in the background of the s^{th} *key* frame, respectively. S' and T' are the number of key frames in

S(m,ki) and *S(n,kj)* for the same character, respectively. There exists spatial transition between *k(i)* with the last shot *S(m,ki)* and *k(j)* with the first shot *S(n,kj)* when the following inequality holds:

$$ST(k_i,k_j) > \frac{1}{2}\left|\frac{1}{S'}\sum_{s=1}^{S'} RA(s)+\frac{1}{T'}\sum_{t=1}^{T'} RA(t)\right|+\frac{1}{2}\left|\frac{1}{S'}\sum_{s=1}^{S'} RA(s)+\frac{1}{T'}\sum_{t=1}^{T'} RA(t)\right| + \frac{1}{2}\left|\frac{1}{S'}\sum_{s=1}^{S'} GA(s)+\frac{1}{T'}\sum_{t=1}^{T'} GA(t)\right|+\frac{1}{2}\left|\frac{1}{S'}\sum_{s=1}^{S'} LA(s)+\frac{1}{T'}\sum_{t=1}^{T'} LA(t)\right| \tag{25}$$

- Rhythmic transition in terms of duration is adopted to represent the tense or clam atmosphere. The intensity of rhythmic transition between scenarios *k(i)* with the number of *M* shots and *k(j)* with the number of *N* shots is computed using following equation:

$$RT(k_i,k_j)=\left|\frac{1}{M}\sum_{m=1}^{M}S(m,ki)-\frac{1}{N}\sum_{n=1}^{N}S(n,kj)\right| \tag{26}$$

There can be declared as the rhythmic transition if the condition in following inequality is true

$$RT(k_i,k_j)>2\left|\frac{1}{M}\sum_{m=1}^{M}S(m,ki)+\frac{1}{N}\sum_{n=1}^{N}S(n,kj)\right| \tag{27}$$

5.4. Scenario Transition Intensity

As aforementioned, the intensity of the flow between two scenarios to the whole story is formulated as the Intensity of Flow (IoF) function. At the same time, according to the discussion in Section 4, the flow of a story consists of various aspects, such as temporal transition, spatial transition, and rhythmic transition, and each metric is normalized between 0 and 1. Therefore, the form of IoF function between scenarios *k(i)* and *k(j)* is the weighted sum of these three metrics as expressed in Equation (16):

$$IoF(k_i,k_j)=\alpha*TT_n(k_i,k_j)+\beta*ST_n(k_i,k_j)+\gamma*RT_n(k_i,k_j) \tag{28}$$

where $TT_n(k_i,k_j)$, $ST_n(k_i,k_j)$, and $RT_n(k_i,k_j)$ are the corresponding normalization form of $TT(k_i,k_j)$, $ST(k_i,k_j)$, and $RT(k_i,k_j)$, respectively. $\alpha+\beta+\gamma=1$.

5.5. Video Skimming Approach

After calculating all pairs of transition intensity of a story-oriented video and given a user-defined target length, the next step for us is to choose the pairs of scenarios with the maximum total IoF value among all possible candidate scenarios.

Many approaches can be used to create video skimming based on the values of intensity of flow among detected key scenarios. We have developed a straightforward scenario-based approach to generate skimming, which does not require complex heuristic rules. Skimming segments of scenario around key frames are selected according to the given skimming duration. The process of skimming segment selection is illustrated in Figure 14.

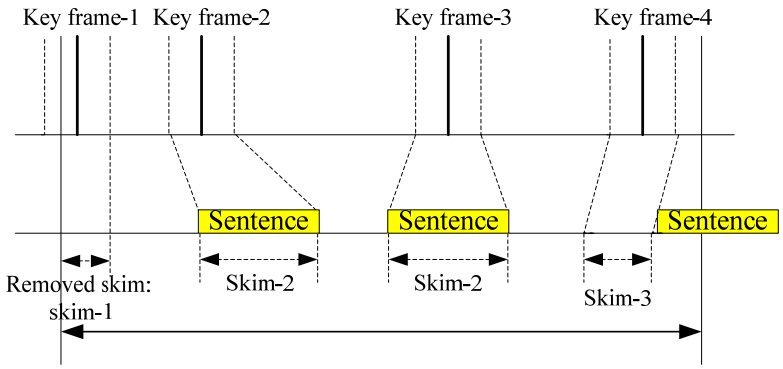


Fig. 14. Video skimming approach.

One crucial element in generating video skimming is the integrity of sentence. It must be uncomfortable for viewers to hear an interrupted sentence while watching a video skimming. Therefore, for the purpose of making a video skimming smoother, we should avoid splitting the speech within a sentence into several parts. In this case, it is indispensable to identify sentence boundary precisely for video skimming. In this chapter, sentence boundary identification comprises following steps.

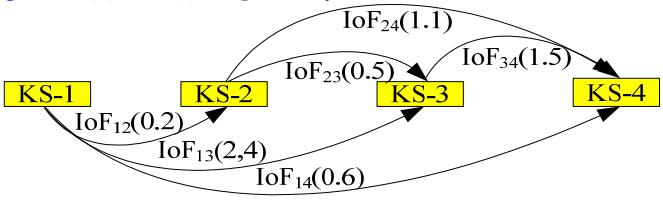
- Discriminating pause segments from non-pause ones using audio energy and zero-crossing rate.
- Smoothing results with respect to the minimum pause duration and the minimum speech duration.
- Determining sentence boundary by longer pause duration.

Besides IoF value, scenario boundary, sentence boundary and key-frames, only four simple rules are used to create video skimming, as shown in Figure 14. The process of video skimming should comply with several criteria described as follows.

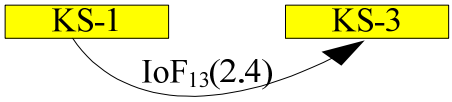
- Given a user-defined target length, pairs of key scenarios with the maximum total IoF value among all possible candidate scenarios are chosen to create video skimming.
- Any video skimming segment should not be less than one second due to the following two aspects. On the one hand, any segment no more than one second is too short to convey message. On the other hand, too short skimming segment may bring annoying impact on human understanding on video content.
- The length of each scenario skimming segment is proportional to the number of key frames in the scenario. Specifically, the default duration of any skimming segment centered on key frame is set to be one second.
- If a skimming segment is beyond the scenario boundary, it will be removed like skim-1 in Figure 14.
- The skimming segment boundaries should be adjusted appropriately in terms of the speech sentence boundaries to prevent from splitting a speech sentence into several parts. Here, the adjustment includes two aspects. One is aligning to the sentence's boundary like skim-2 in Figure 14; the other is evading the sentence's boundary like skim-3 in Figure 14.

Figure 15 shows examples of the story-oriented video skimming. Figure 15 (a) is a graph of the transition intensity among key scenarios of a story-oriented video. The durations of each

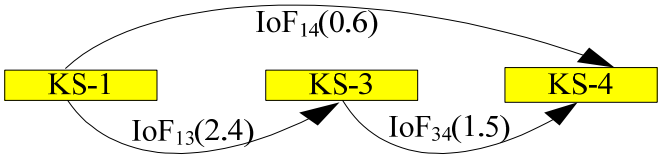
scenario skimming *ks1*, *ks2*, *ks3*, and *ks4* are 5 seconds, 10 seconds, 15 seconds, and 10 seconds, respectively. The skimming with the target durations of 7 seconds and 10 seconds are shown in Figure 15 (b) and (c), respectively.



(a): Intensity of flow of a video.



(b): A skimming with target length of 7 seconds.



(c): A skimming with target length of 10 seconds.

Fig. 15. Examples of the skimming of a story-oriented video.

6. Experimental Results

In this section, we will discuss some implementation issues in practical application and evaluate our approach on various genres of films with existing state-of-the-art methods.

6.1. Experimental Settings

Totally seven full-length movies in MPEG-1 format are selected to evaluate the performance of the proposed algorithm. Each video track is analyzed at 25 frames per second with a resolution of 320×240, while the sound track is processed at the sampling rate of 22 kHz with mono-channel, 16-bit precision. As shown in Table 2, the data set consists of various types of contents, which would demonstrate that the proposed algorithm can work on different kinds of movies. All the experiments are performed on an Intel Pentium IV 3.0 GHz machine and 1G memory computer running Windows XP.

No.	Name	Genre	Length (hh:mm:ss)
1	Mission Impossible III	Bodyguard	1:55:23
2	X Man III	Action	1:44:03
3	Walk in the Clouds	Family	1:42:14
4	The Girl Next Door	Love	1:48:10
5	The Ring	Horror	1:53:43
6	The Sound Of Music	Musical	2:54:33
7	N Death on the Nile	Detective	2:11:50

Table 2. Summary of the testing dataset.

6.2. Experiment I: Scenario Boundary Detection

The performance of scenario boundary detection (SBD) is important for video skimming because the proposed video skimming scheme is based on the scenarios. In order to evaluate the effectiveness of the proposed SBD approach, we compare it with the SBD approach proposed by Rasheed *et al.* [25] and Zhao *et al.* [26].

To get the ground truth of scenarios, ten graduate students are invited to watch the movies and then give their own scene boundaries. The ground truth used for the experiments is the intersection of their segmentation. Generally speaking, there is no such a clear boundary between two adjacent scenes in movies due to film editing effects. Therefore, the most commonly used criterion, Hanjalic's evaluation [19] is here used to match the ground truth with the detected ones: if the detected scene boundary is within four shots from the boundary detected manually, this boundary is counted as a correct boundary.

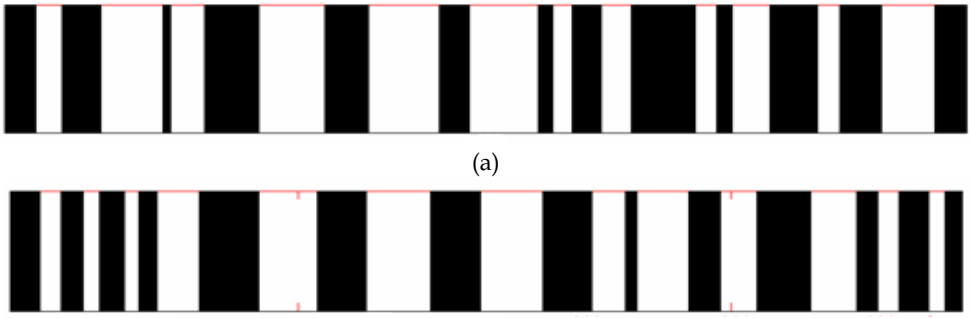
We use 'Precision', 'Recall' and 'F1-Score' to evaluate the performance of those techniques. The values of recall and precision are in the range of [0, 1]. The higher recalls indicate a higher capacity of detecting correct shots, while the higher precisions indicate a higher capacity of avoiding false matches.

No.	Proposed		[25]		[26]	
	Pre	Re	Pre	Re	Pre	Re
1	0.755	0.809	0.641	0.695	0.712	0.763
2	0.741	0.814	0.572	0.609	0.676	0.638
3	0.813	0.849	0.713	0.737	0.708	0.718
4	0.835	0.878	0.687	0.623	0.722	0.693
5	0.814	0.839	0.706	0.723	0.683	0.706
6	0.843	0.862	0.734	0.713	0.714	0.704
7	0.834	0.857	0.753	0.741	0.727	0.707

Table 3. Comparison of Scenario Boundary Detection (Pre: Precision, Re: Recall).

The results are given in Table 3. From Table 3, it can be seen that average F1-Score of the proposed approach, Backward Shot Coherence method and the Normalized Cuts method are 0.824, 0.688, and 0.704 respectively. That is, the proposed approach exhibits a gain 19.7% and 16.9% of the average F1-Score compared with the method in [25, 26] respectively. The reason why the proposed approach gains a large improvement in all the evaluating measures is based on following two phases. On the one hand, the temporal constraint is integrated into the spatial coherence clustering in the procedure of scene segmentation. On the

other hand, the process of forward and backward clustering also helps to improve the performance.



(a)

(b)

Fig. 16. Example of scenario detection. (a): Ground truth scenes and (b): detected scenes of the movie 'X Man III'.

Figure 16 shows the detail scene detection results of the movie 'X Man III'. The upper row indicates the ground truth scenes represented by alternating black / white stripes, and the bottom row is the detected results using the proposed scheme.

6.3. Experiment II: Key Scenario Identification

Since the proposed video skimming scheme is based on the flow of story between key scenarios, the detection precise of key scenarios have an important impact on final video skimming. Like the method used in providing the ground truth of scenario boundary, ten volunteers are invited to manually label key scenario.

Table 4 lists experimental results of the identification of key scenarios. As seen in Table 4, the average F1-Score is 0.855, which clearly demonstrate that the proposed scheme can detect and classify video scene into categories. That is to say, the selected features for describing scene content and the way of deciding scene type are satisfactory.

Conversation scene	Precision	0.897
	Recall	0.863
	F1	0.880
Suspense scene	Precision	0.863
	Recall	0.843
	F1	0.853
Action scene	Precision	0.846
	Recall	0.824
	F1	0.835

Table 4. Experimental results of Key Scenario Identification.

Figure 17 shows some examples of different types of scenario transitions.

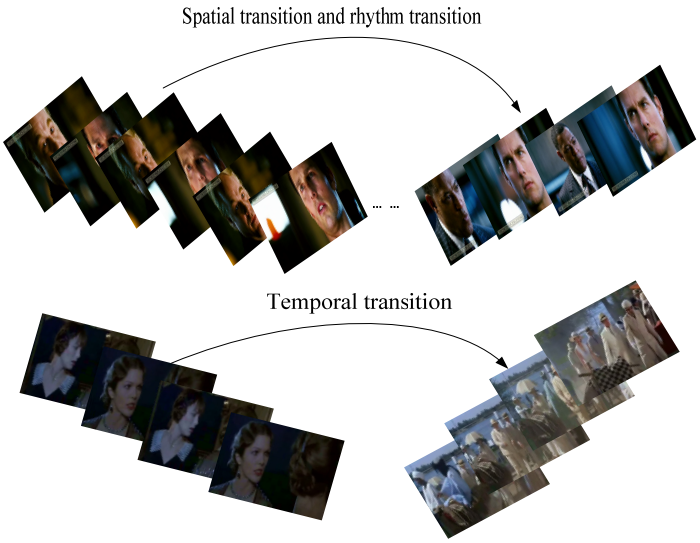


Fig. 17. Examples of scenario transition.

6.4. Experiment III: Video Skimming

Generally speaking, a good video skimming should be as short and as information as possible. However, it is difficult to achieve the two objectives at the same time. In this chapter, three criteria are adopted to evaluate the algorithm performance.

For a good video skimming, the first criterion is required to correctly select essential segments for viewers to grasp the clue of the flow of a story. We set the first criterion as informativeness including coverage and conciseness, where *coverage* means a skimming should include all the important segments of a story and *conciseness* means a skimming should comprise only the necessary segments. The second criterion is required to maintain the validity of interrelationships between segments. We set the second criterion as *coherence*, which denotes each segment of a skimming should be interrelated with others with respect to the whole story. Besides, *satisfaction* of the video skimming is also an important criterion. The last criterion appraises not only the smoothness of an image sequence, but also the integrality of speech sentence. Consequently, we will carry out two different experiments according to above discussed three criteria.

6.4.1. Verifying the First Criteria

This experiment aims to measure the precision and recall of each skimming compared to the manually produced ground truth by evaluating the coverage and conciseness of the proposed approach.

Table 5 shows the experimental results of precision and recall for each skimming with three different lengths (5, 10, 20 minutes). From this table, we can see that the overall average precision are 80.3%, 82.6%, and 84.2% for video skimming with the duration of 5, 10, and 20 minutes, respectively. The overall average recall are 80.8%, 82.9%, and 84.3% for video skimming with the duration of 5, 10, and 20 minutes, respectively. These numbers indicate that the proposed skimming scheme is effective in generating highly compact skimming.

No.	Precision			Recall		
	5 M.	10 M.	20 M.	5 M.	10 M.	20 M.
1	0.753	0.793	0.812	0.773	0.812	0.829
2	0.784	0.806	0.821	0.792	0.813	0.817
3	0.812	0.832	0.852	0.833	0.846	0.853
4	0.836	0.848	0.863	0.827	0.846	0.864
5	0.805	0.827	0.843	0.826	0.819	0.832
6	0.841	0.856	0.872	0.824	0.843	0.862
7	0.793	0.817	0.834	0.781	0.826	0.841
Avg.	0.803	0.826	0.842	0.808	0.829	0.843

Table 5. Results of informativeness. (M.: minute).

6.4.2. Evaluating the last two Criteria

We perform subject tests to measure the *coherence* and *satisfaction* by evaluating viewers' satisfaction and preference over skimming sequences with different duration. Twenty volunteered subjects, including 11 males and 9 females, are invited to participate in the experiment. They are required to specify their preferences to each skimming sequence according to a list of prepared questions, where the measurement of preference is categorized into three following levels: "bad", "neutral" and "good".

To achieve precise scores, the subjects should be kept innocent of the video content before they view the video skimming. The subjects view the video skimming sequences of 5, 10, and 20 minutes and the original video in turn, and they are required to give a score to the test sequence when he/she finishes viewing a skimming sequence. To be fair, the subjects are given the chance to modify the scores assigned to the skimming sequences any time during the review process, because they would understand the video content in more details with the passage of time. With the scores assigned by subjects, corresponding average scores are then calculated, which reflect the viewers' satisfactory degree to different video skimming sequences.

L	Q1			Q2			Q3		
	B	N	G	B	N	G	B	N	G
5	5	20	75	5	12	83	6	14	80
10	3	13	84	2	8	90	2	10	88
20	0	10	90	0	9	91	0	9	91

Table 6. Performance Evaluation of Video Skimming (L.: length, B: Bad, N: Neutral, G: Good).

The experimental results of the subject preference of video skimming are list in Table 6. From this table, we can see that the produced skimming sequences show good performance in terms of viewers' satisfaction and preference. The units of later fine columns are '%'. The prepared issues are list as follows.

- Q1: Do you think that scenario segments in the skimming are mutually coherent?
- Q2: Do you think that scenario segments in such duration are all appropriate?
- Q3: Do you think that the meaning of speech sentence in each scenario segment is unambiguous?

7. Conclusions

Automatic video skimming is a powerful tool for video browsing and retrieval. In this chapter, we present a new approach for story-oriented videos. The proposed framework automatically generates video skimming that help viewers grasp the main content within a given duration, which is achieved by exploring the clues about human understanding of a story according to general scenario writing rules and editorial techniques. After the process of the detection of scenario boundaries and the identification of key scenarios, a video skimming is produced by selecting the maximum total Intensity of Flow value among all possible candidate scenarios satisfying the given target duration.
Experiments reveal that our framework obtains good performance for all the criteria of a story-oriented video skimming: informativeness, coherence, and satisfaction. The overall average precision and recall are both over 80% compared to the manually generated ground-truth. Furthermore, the subjective tests show that viewers can achieve high satisfaction and preference over the skimming sequences. We believe that these experimental results indicate our scheme is a feasible solution for the management of video repository and online review service.

Acknowledgement

This work is supported by the National High-Tech Research and development Project of China (973) under Grant No. 2006CB303103, and also supported by the National High-Tech Research and development Project of China (863) under Grant No.2009AA01Z330 and the National Natural Science Foundation of China under Grant No. 60833009.

8. References

[1] Pfeiffer, S., Lienhart, R., Fischer, S., and Effelsberg, W. Abstracting digital movies automatically. Int. J. Visual Communication and Image Representation, 7(4) (1996) 345–353.

[2] Hanjalic, A., Kakes, G., Lagendijk, R. L., and Biemond, J. Indexing and retrieval of broadcast news programs using Dancers. SPIE J. Electronic Imaging, 10(4) (2001) 871–882.

[3] Kim, J., Hyun S., Kang K., Kim M., Kim J., and Kim H. Summarization of News Video and Its Description for Content-based Access. Int. J. of Imaging Systems and Technology, 13(5) (2003) 267-274.

[4] Lie. W., and Lai, C. News Video Summarization Based on Spatial and Motion Feature Analysis. (PCM 2004).

[5] Babaguchi, N., Kawai, Y., and Kitahashi, T. Event based indexing of broadcasted sports video by intermodal collaboration. IEEE Trans. Multimedia, 4(1) (2002) 68–75.

[6] Chen, C., Wang, J., and Wang, J. Efficient News Video Querying and Browsing Based on Distributed News Video Servers IEEE Trans. Multimedia, 8(2) (2006) 257-269.

[7] Yeung, M, and Yeo, B. Video visualization for compact presentation and fast browsing of pictorial content. IEEE Trans. Circuits System and Video Technology, 7(5) (1997) 771–785.

[8] Hanjalic, A., and Zhang, H. An integrated scheme for automated video abstraction based on unsupervised cluster validity analysis. IEEE Trans. Circuits System and Video Technology, 9(8) (1999) 1280–1289.
[9] Ma, Y., Lu, L., Zhang, H., and Li, M. A user attention model for video summarization. (ACM MM 2002).
[10] Lee, S., and Hayes, M. An application for interactive video abstraction. (ICASSP 2004).
[11] Ma, Y., Lu, L., Zhang, H. Video Snapshot: A Bird View of Video Sequence. (MMM 2005).
[12] Valdés, V., and Martínez, J. On Video Abstraction Systems' Architectures and Modeling. (SAMT 2008).
[13] Luo, H., Gao, Y., Xue, X., Peng, J., and Fan, J. Incorporating feature hierarchy and boosting to achieve more effective classifier training and concept-oriented video summarization and skimming. ACM Trans. Multimedia Computing, Communications and Applications, 4(1) (2008) 1-25.
[14] Sundaram, H., and Chang, S Computable scenes and structures in films. IEEE Trans. Multimedia, 4(4) (2002) 482–491.
[15] Bordwell, D., and Thompson, K. Film Art: An Introduction. Technology report, McGraw-Hill Companies, 1996.
[16] Abutableb, A. Automatic thresholding of gray-level pictures using two-dimensional entropies. J. Computer Vision, Graphics, Image Processing, 47(1) (1989) 22 – 32.
[17] Smoliar, S., and Zhang, H. Content-based video indexing and retrieval. IEEE Mage. Multimedia, 1(2) (1994) 62–72.
[18] Yeo, B., and Liu, B. Rapid scene analysis on compressed videos. IEEE Trans. Circuits and Systems for Video Technology, 5(6) (1995) 533–544.
[19] Hanjalic, A., Lagendijk, R., and Biemond, J. Automated high-level movie segmentation for advanced video-retrieval systems. IEEE Trans. Circuits System and Video Technology, 9(4) (1999): 580–588.
[20] Bouthemy, P., Garcia, C., Ronfard, R., Tziritas, G., Veneau, E., and Zugaj, D. Scene Segmentation and Image Feature Extraction for Video Indexing and Retrieval. (VIIS 1999).
[21] Arai, H. Fundamental Techniques for Scenario Writing. Da-Bo Munhwa, 1987.
[22] Bordwell, D., and Thompson, K. Film Art: An Introduction. McGraw-Hill Companies. 1996.
[23] Chaisorn, L., Chua, T., and Lee, C. The segmentation of news video into story units. (ICME 2002).
[24] Aner, A., and Kender, J. Video summaries through mosaic-based shot and scene clustering. (ECCV, 2002).
[25] Rasheed, Z., and Shah, M. Scene detection in Hollywood movies and TV shows. (CVPR 2003).
[26] Zhao, Y., Wang, T., Wang, P., Hu, W., Du, Y., Zhang, Y., and Xu, G. Scene Segmentation and Categorization Using NCuts. (CVPR 2007).

5

Image Matching and Recognition Techniques for Mobile Multimedia Applications

Suya You, Ulrich Neumann, Quan Wang and Jonathan Mooser
Computer Graphics and Immersive Technologies Lab
Computer Science Department
University of Southern California
USA

1. Introduction

Image spatial-temporal matching is a fundamental task in visual media processing used to comprehend two or more images taken, for example, at different times, from different sensors, or from different aspects. Many multimedia systems and applications require image matching, or closely related operations as intermediate steps, including image database classification and retrieval, Internet image search engine, and multimedia content analyzing and understanding. The rapid convergence of multimedia, computation and communication technologies with technique for device miniaturization is ushering us into a mobile, wireless and pervasively connected multimedia future. It is now quite common for mobile platform to integrate a mobile phone, digital camera, music player, and PDA into one device, heralding many exciting new applications and services, as exemplified by several new multimedia phone services recently launched, such as the Nokia Point & Find, MyClick in telecommunication markets. The services employ the image recognition and search techniques that allow users to capture designated images such as product advertisements and quickly match them to the vendor's databases to obtain detailed product information associated with the images.

While capability trend is clear, the main challenge posted by such multimedia systems is the developed image matching and recognition algorithms have to be reliable, fast, robust and capable to handle the realistic conditions in our real-world. Technically, any image matching and recognition process generally consists of three components: (1) feature selection and detection - finds stable matching primitives over spatial-temporal space to achieve scale and pose invariance, (2) feature representation and description - represents the detected features into a compact, robust and stable structure for image matching, and (3) optimal matching and search - uses the feature descriptions and additional constrains to locate, index, and recognize the targets and scenes of interest.

This chapter presents several advanced image matching and recognition technologies, of particular emphasis on mobile multimedia applications that are feasible with current or near term technology and applications. The presented work represents the latest state-of-arts where we have a strong base of techniques and knowledge in this area. In Section 2, we

present a high-performance matching technique required by real-time multimedia applications. In Section 3, a highly compact image feature description and matching technique is presented, targeting the mobile multimedia applications. Section 4 describes an efficient image search technique that can dramatically improve on previous approaches over hundred-times faster for recognition of objects from a large library of database. Finally, Section 5 presents an application system, called Augmented Museum Exhibitions that combines the mobile computation, Augmented Reality and image matching/recognition techniques to demonstrate the effectiveness and utility of the presented technologies.

2. Real-Time Image Matching Based on Multiple View Kernel Projection

Technically, any image matching process generally consists of three components. (1) Feature detection finds stable matching primitives over spatial scale space to achieve scale and pose invariance. Recently, local features have been widely employed due to their distinctiveness and ability to handling complex imaging conditions such as occlusions and cluttered backgrounds (Ke & Sukthankar, 2004; Lepetit et al., 2004; Lowe, 2004; Mikolajczik & Schmid, 2003; Tuzel et al., 2006). The approach described in this Section extracts highly distinctive local features, i.e. interest points, as basic primitives to represent image information and perform image matching. (2) Feature description represents the detected features into a compact, robust and stable structure for image matching. Among the various proposed approaches, Kernel Projection using Walsh-Hadamard kernels has demonstrated better performance in terms of robustness and processing time (Hel-Or Y. & Hel-Or H., 2005). Kernel Projection, however, does not naturally have the important property of geometry invariant. Therefore cannot handle geometric distortions caused by viewpoint or pose changes. We solved this problem by introducing a novel approach called Multiple View Kernel Projection (MVKP) to represent and describe the detected local features. The unique feature representations are compact and show superior advantages in terms of distinctiveness, robustness to occlusions, and tolerance of geometric distortions. (3) Optimal matching and search uses the feature descriptions and additional constrains to locate, index, and recognize the targets and scenes of interest. Local feature based approaches typically produces a large number of features needed to match (Lowe, 2004; Mikolajczik & Schmid, 2003; Tuzel et al., 2006; Lepetit et al., 2004). Methods that search exhaustively are highly computationally expensive, unsuitable for real-time applications. To resolve this problem, we use an effective approach, Fast Filtering Vector Approximation (FFVA) that can efficiently match a very large high-dimensional database of image features in real-time. The FFVA technique will be described late in Section 4.

We integrated all above components to produce a complete image matching system for a range of applications including automated object recognition, non-text image retrieval, and wide-baseline image matching and registration. We have extensively tested the system with both synthetic and real datasets. Comparing with several existing methods, the real-time MVKP system demonstrates both effectiveness and robustness.

2.1 Relevant work

Among the image matching approaches based on local features, early works mostly focused on the information provided by one single view of the object. Schmid and Mohr [Schmid & Mohr, 1997)] introduced a rotationally invariant descriptor for local image patch based on

local greylevel invariants. The ground-breaking work of D.G. Lowe [Lowe, 2004)] demonstrated that rotation as well as scale invariance can be achieved by first using difference-of-Gaussian function to detect stable interest points, then construct the local region descriptor using assigned orientation and several histograms. The proposed SIFT method produced significant influence on later works. For example, Ke and Sukthankar (Ke & Sukthankar, 2004) applied PCA to image gradient patch in order to reduce the descriptor's dimensionality. GLOH (Mikolajczik & Schmid, 2003) is an extension of the SIFT descriptor by computing it for a log-polar location grid with 3 bins in radial direction.

To achieve viewpoint invariant, another line of research is to combine the information of multiple views and train the system in an offline stage so that it will learn the main characters of the same object under different viewing conditions. Consequently, the online matching process can be much faster, even real-time.

Concerning the data source of multiple views, some works use affine transformation to synthesize a number of views from one single input view [Lepetit et al., 2004; Lepetit et al., 2005; Boffy et al., 2006; Ferrari et al., 2005] while others take real images captured from camera as input (Ozuysal et al., 2006; Winn & Criminisi, 2006). Our approach also employed the similar idea by introducing a multiple view training stage to generate a number of synthetic views from single image. We choose the synthesized-view approach due to the ground truth of training images' correspondences it can provide. We use Kernel Projection scheme to extract the significant components containing in the synthesized images and to establish compact feature descriptors.

Lepetit (Lepetit et al., 2004) treats the multiple-view point matching problems as a classification problem. They synthesize small patches of each individual feature point served as training input. PCA and k-mean algorithms are applied to those patches to provide the local descriptor. After the offline training stage, the same keypoint detector and PCA projection matrix are used on the query image patches. Eventually, the feature vectors of training and query images are matched by simple linear scan. Later on in their continuous work (Lepetit et al., 2005), classification tree is used to replace the PCA and k-mean as well as the final nearest neighbor search. The branching of the trees is decided by simple comparison of nearby intensity values and the final classification is determined by statistic analysis at the leaf node. The online matching process is fast enough for real-time application. However, the forest construction is very slow (10-15 minutes) and it is pointed out that their actual results can vary depending on the viewpoint and illumination conditions (Boffy et al., 2006).

(Ozuysal et al., 2006) is an extension of the randomized tree (RT) focusing on non-planar object tracking without 3-D model. With the help of RT structure, features can be updated and selected dynamically, called "harvest". The training views are obtained by moving the object slowly in front of the camera. Tuzel (Tuzel et al., 2006) proposes the covariance of d-features as a new descriptor, computed from a set of integral images. A distance metric for the new descriptor is also given. Boffy (Boffy et al., 2006) uses additional information about the appearance of the object under the actual viewing condition to update the classification trees at run-time. They also use special designed spatially distributed trees to enhance the reliability and speed.

Projection and rejection scheme has long been proved to be efficient for pattern matching and general classification problems. Various projection vectors have been studied. Among the previous works, researchers emphasized on the discrimination abilities of the projection

kernels (Elad M. et al., 2000; Keren et al., 2001), while Y. Hel-Or, et al. (Hel-Or Y. & Hel-Or H., 2005) argued that besides the discrimination, it is also important to choose projection kernels that are "very fast to apply". For this purpose, they choose Walsh-Hadamard (WH) kernels and achieved a speed enhancement by almost two orders of magnitude. Furthermore, experimental results indicate their Projection Kernel method is robust to noise and lighting changes. However, as a fast window matching technique, the method cannot handle geometric distortion brought by view angle changes.

2.2 The Walsh-Hadamard Kernels Projection

The projection scheme in our MVKP method is based on Walsh-Hadamard (WH) kernels, which is a special case of Gray-Code Kernels (Ben-Artzi et al., 2004) and general projection kernels in Euclidean space.

2.2.1 General projections in Euclidean space

Suppose there are two sets of image patches with size $k \times k$. Each patch can be directly expressed as a k^2 dimensional vector. Therefore, the similarity between two patches can be measured as the Euclidean distance between the two corresponding vectors. Obviously, such similarity is impractical to compute especially when the number of patches to be measured is large. The projection strategy is to project the original vectors onto a smaller set of projection kernels, which are fast to compute and still maintain the distance relationship.

Assume $\vec{b}_1, \vec{b}_2, \vec{b}_3 \ldots$ are orthonormal projection basises in k^2 dimension Euclidean space (Figure 1(a)). *P* is a point in the k^2 dimension space with projected components $\vec{v}_1, \vec{v}_2, \vec{v}_3 \ldots$ respectively. Scalars $c_i = \vec{v}_i^T \vec{v}_i$. Let $d(P)$ represents the squared Euclidean distance from *P* to the origin *O*, then we have:

$$d(P) = \sum_{i=1}^{k^2} c_i^2 \tag{1}$$

It is trivial from the above equation or followed from the Cauchy-Schwartz inequality since the Euclidean distance is a norm, that lower bounds of $d(P)$ can be calculated using a number of projection scalars. The lower bounds layers can be expressed as the following:

$$\sum_{i=1}^{1} c_i^2 \le \sum_{i=1}^{2} c_i^2 \le \sum_{i=1}^{3} c_i^2 \le \cdots \le \sum_{i=1}^{k^2} c_i^2 = d(P) \tag{2}$$

With the increasing number of projections kernels involved in the calculation, the lower bound becomes tighter. When all the k^2 kernels are involved, the lower bound becomes the actual squared Euclidean distance. For the projection scheme to be efficient, there are two factors need to be considered: On the one hand, the projection basises should be ordered in a way such that the lower bound can become tight after only a small number of projections. On the other hand, equally important is the requirement that "the kernels should be efficient to apply enabling real-time performance" [Hel-Or Y. & Hel-Or H., 2005].

2.2.2 The Walsh-Hadamard kernel

The WH kernel is one special case of the Gray-Code projection kernels satisfying the above two requirements. First, the WH projection kernels are very efficient to generate and apply. One-dimensional kernels can be generated using binary tree while consecutive kernels are α-related (Ben-Artzi et al., 2004). In the context of 2-D image processing, two-dimensional kernels can be generated as the outer product of one-dimensional kernels. All the coordinates of WH kernel's basis vectors are either +1 or –1. Consequently, projection onto WH kernels involves only dimensionality number of additions or subtractions, which can be performed very fast.

Second, when the kernels are ordered according to increasing frequency of sign changes, the experimental results show that a tight lower bound can be achieved using only a small number of kernels. Thus, we can greatly reduce the complexity of similarity computation while still captures the major difference between feature vectors. Figure 1(b) shows a list of two-dimensional WH kernels in increasing order of frequency. (Ben-Artzi et al., 2004) introduced an efficient algorithm to compute the ordering of the kernels, which captures the increase in spatial frequency.

2.3 MVKP for Real-time Feature Matching

Kernel projection using WH kernels is able to measure the similarity between two large sets of image patterns in real-time, however, it can not handle geometric variance caused by view angle changes. In order to achieve invariance and tolerance to geometric distortions, we combine the WH kernel projection method with a multiple view training stage. The training stage is aimed at providing the system with additional information concerning affine distortions, such that the same object can still be matched under different view angles.

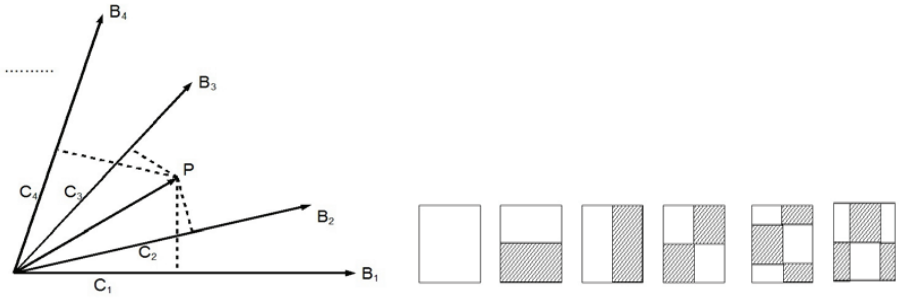


Fig. 1. Kernel projection. (a) General projection. (b) 2-dimensional 4×4 WH kernels in increasing order of frequency. Blank represents value "1" and shadow represents "-1".

2.3.1 Offline training stage

During the offline training stage, the MVKP method takes one object image as input, smooth it using Gaussian filter, generate 50-100 synthetic training images from it and then describe the main characters for each selected object location. The output of the training stage is a set of feature vectors, subsets of which corresponds to each selected object location. Figure 2 illustrates the major components of the training stage.

The method first synthesizes a number of training views of the input object image using affine transformation. A general affine transformation can be expressed as:

$$x' = H_A x = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} x$$
$$A = R(\theta)R(-\phi)DR(\phi) \text{ and } D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (3)$$

where R is the rotation matrix and t is a translation with components $t1$ and $t2$. Matrix A corresponds to a rotation of θ first, followed by a rotation of $-\varphi$ then scale changes of $\lambda 1$ and $\lambda 2$ in horizontal and vertical direction respectively. At last, the image is rotated back by φ . The six affine transformation parameters are generated randomly to cover the whole parameter space for rotation and shear angles. So we choose the ranges $\theta \in [-\pi,\pi]$, $\varphi \in [-\pi/2,\pi/2]$, $\lambda 1$,$\lambda 2 \in [0.4, 1.6]$, $t1$,$t2$ = 0, 1, 2, or 3.

Searching for local maximum Eigen-values within 3×3 local patches identifies the local feature points. The patches with a local minimum smaller than a threshold are discarded. The detector is designed to guarantee that one feature point will not be too close (for example, 3 pixels) to one another. Otherwise, two features might have a very similar description and consequently fail the distance ratio criteria. After all the keypoints in all the synthetic views are detected, we can tell how many of them belong to the same object location in the object image, since all the affine transformations are synthesized. It is assumed that the object locations that appeared more often on the synthetic views have a higher probability to be detected in the query image containing the same object (Lepetit et al., 2005). Therefore, we select 100-200 "mostly common appeared" object locations for future feature matching use. Each object location is represented as a link list containing the coordinates in the corresponding views. Within each synthetic view, we extract a 32 × 32 patch around each detected and selected feature point. Because the projection of the image patch onto the first WH kernel conveniently gives its DC value. Robustness to lighting changes can be achieved by simply disregarding the first projection kernel. In addition to that, we normalize (translate and rescale) each patch's intensity values to the same range in order to enhance the performance against different lighting conditions.

The lists of extracted image patches contain the information of various possible appearances for all feature locations. The last step of the training stage is to describe the extracted patches into feature vectors. Each patch's intensity values, forming a very-high-dimension vector, are provided to the kernel projection method so that the final descriptors belongs to the same object location can be more effective, compact and contain the information under various viewing situations. WH kernels are used for the kernel projection. In our experiments, we found that typically the first 20 WH kernels are enough for a reliable feature description. After kernel projection, k-mean can be used to further reduce the size of the feature set. For all the feature vectors representing the same object location, 10-20 clusters are formed, and the center vector of each cluster is used to represent that cluster.

2.3.2 Feature set construction for query image

Given a query image containing the same object, our goal is to find the correspondences between the query image and the object image. After the offline training stage, we have lists

of object feature vectors. Each of them corresponds to an interest selected object location. Now we need to construct a similar feature set for query image.
After the query image is read and smoothed, the same feature point detector is applied. Because this is an online stage desired to be as fast as possible, we only select a number of "strongest" feature points reported by the detector. Let x_1 be the number of selected stable object locations in the training stage and x_2 is the number of selected feature points in this stage, y is the number of final reported correspondence (NFRC), then we have:

$$y \le \min(x_1, x_2\} \tag{4}$$

Typically, x_2 is around 500, assume x_1 is 100, then the NFRC will be no larger than 100.
After the keypoint detection, the intensity values of the image patch around each keypoint give us original vector description. Those original vectors are normalized to the same intensity range to enhance the robustness against lighting changes. The normalized vectors are projected onto a number of (the same number as in the training stage) WH kernels resulting in compact final descriptors. As the first part of online query process, the feature set construction is comparatively much faster. The most time-consuming part is to find the correspondence between feature sets.

2.3.3 Establishing feature correspondences

Give the feature descriptors covering various viewing conditions for each object location and the feature descriptors for the query image, the final task is to establish the correct correspondence between two features sets efficiently. The rejection scheme in (Hel-Or Y. & Hel-Or H., 2005) can't be directly adapted to our problem because it requires all the query image patches be continuously distributed. Thus we use a different technique based on lower bound rejections to accomplish the task.
We employ Euclidean distance as similarity metric due to its simplicity and low computational cost. Nearest Neighbor (NN) search techniques have been studied under this context. The authors of (Lepetit et al., 2004) use linear scan because of its simplicity and accuracy, while in (Jeffrey & Lowe, 1997), an approximate NN-search method over traditional Kd-tree structure is introduced in order to efficiently index the high-dimensional (128-D) feature vectors.
To decide the proper NN-search technique for the MVKP method, first we investigated the feature properties generated by WH kernel projection. The following is an example of three feature vectors generated by kernel projection at the training stage:
Feature vector #1: 22875, 2962, -1843, -935, 1037…
Feature vector #2: 17886, -2797, 1175, 315, -1008…
Feature vector #3: 19568, -3567, 1338, 347, 1572 …
The dimensionality of our feature vector typically ranges from 20 to 100 (depending on how many kernels are used) while the magnitude is comparatively large. It can also be seen from the experiments that, our features vectors are sparser distributed in the space compared with feature vectors in (Lepetit et al., 2004), where features are more likely to cluster together. Our feature vectors are more distinctive and further away one from another. Accordingly, we use fast FFVA method to perform the NN-search, which is described in Section 4.

2.4 Experimental Results and Evaluations

The method has been tested using synthetic and real images as well as the combination of both. Real images are captured using a DSLR camera with high light-sensitivity settings (ISO=800~1600) and incamera noise reduction off, which gives the input pictures hardware (CCD) generated randomly distributed noise points with random intensity. To obtain the synthetic images, we either performed synthetic viewpoint and lighting changes on the real images or download computer generated images from the Internet.

We compared the MVKP method with SIFT method, the classification method using PCA (Lepetit et al., 2004) represented by CPCA and the randomized tree based method (Lepetit et al., 2005) represented by CRTR. In all the experiments the number of generated training views for MVKP is 100, for CRTR is 1000, the number of selected objection feature location is 200, the maximum keypoint number returned by the detector is 500, image patch size is 32 by 32 and the number of k-mean kernels to represent each object location is 20. The test computer is a desktop PC with 1.4GHz CPU.

2.4.1 Effect of projection kernels

To evaluate the effect of projection kernels, we use two original 256-dimensional feature vectors (one from training stage and the other from the query image) and project them onto the first 5, 10, … , 125 WH kernels respectively. Each time, we calculate a lower bound of the squared Euclidean distance (around 3×108 for this test) between the projected vectors.

Figure 2 demonstrates the kernels' effectiveness. Compared with the result of standard basis vectors, the projections onto the first 20~50 WH kernels already captures the majority difference between the two vectors.

2.4.2 Feature distinctiveness

This experiment shows that the feature vectors generated from WH kernel projections are sparser distributed in the space compared with CPCA method. In other words, our feature vectors are more distinctive one from the other, resulting more reliable feature matching and allowing vector approximation based NN-search technique like FFVA working more efficiently.

Figure 3 shows that for CPCA and MVKP method, the number of reported correspondences under the same distance ratio α=0.5. For the same feature sets size, MVKP has a much higher reported correspondence number indicating MVKP's feature vectors are less likely to cluster together than CPCA's. Therefore, it is easier to find a distinctive matching in MVKP's feature space.

2.4.3 Matching accuracy and robustness

For synthetic image test, even without the consistent check step like RANSAC, the MVKP method is able to return a large number of correct matching in real-time. The really challenging experiments are those using real images or the combination of real-world and computergenerated images. For pure real image tests, the pictures of the same object are captured from different view angle, under different lighting condition and maybe partial occluded. We also had live-demo comparison when query images arriving in real-time captured from a live web camera.

Fig. 4. Evaluation with real image sets. Note that the training images (right side) are computer-generated graphics downloaded from the Internet. The testing images (left sides) are captured with digital camera.

Overall, the MVKP method (although only 100 views are used for training) shows better accuracy, comparable number of reported correspondences and faster query speed compared with CRTR (1000 views training). In some very difficult testing cases, CRTR gives no output at all while our method still have rather good results. Figure 4 shows examples of evaluation results.

2.4.4 Matching speed

CPCA treats feature matching as a classification problem and achieves an online matching speed 5 times faster than SIFT. It is fast enough for many application areas but still not real-time. The introduction of randomized tree (CRTR method) brought the performance into the real-time range and also obtained more robust performance. Figure 5 shows the results of our query speed test using large-size synthetic images.

The number of feature vectors generated from training stage is 4,000 while the number of query image feature vectors ranges from several hundreds to 5,000. The original real images include aerial image, ad poster, small indoor item and complex scene. Synthetic scale, rotation, shear and lighting changes are applied to those real images to generate query images. Linear scan is used for both CPCA and our MVKP methods. Even when the simple and slow linear scan is used, MVKP's query speed is much faster than CPCA and comparable with CRTR. The typical training time for CRTR is around 15 minutes, for CPCA is around 1 minute and for MVKP is only around 30 seconds.

Analyzing the matching time composition for MVKP method we found that the time spent for NN-search using linear scan is more than 70% of the total online matching time. The description step is within 0.1 seconds thanks to the fast applicable WH projection kernels. When the feature sets size is large or the application demands a large number of correspondences, a more efficient NN-search technique is the key to the whole system performance. We choose FFVA method due to the feature vector's two inherent properties: large magnitude and sparse distribution. Our experimental result shows a significant speed

enhancement (more than double the speed) over the linear scan method. Although an approximated NN-search technique, FFVA has accuracy close to linear scan.

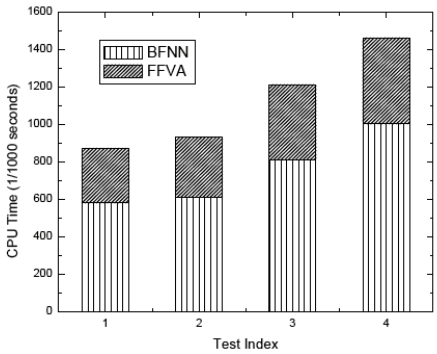


Fig. 5. Matching speed evaluation with comparisons of BFNN versus FFVA

3. Highly-Compact Image Feature Description and Matching Technique

Many methods for feature descriptions have been suggested, which can incorporate various degrees of resistance to common perturbations such as viewpoint changes, geometric deformations, and photometric transformations. Among the approaches, the SIFT descriptor has been shown to outperform other descriptors (Lowe, 2004). The SIFT descriptor is based on the gradient distribution in salient region, and constructed from a 3D histogram of gradient locations and orientations. A 128-dimension vector representing the bins of the oriented gradient histogram is used as descriptor of salient feature.

However, the high dimensionality of SIFT descriptor is a significant drawback, especially for online or large-scale dataset applications. For a typical outdoor scene, for example, the SIFT usually produces several hundreds of local features, yielding a large high-dimensional feature space needs to be searched, indexed, and matched.

Several researchers have addressed the problem of dimensionality reduction for feature descriptors. For example, Bay et al. (Bay et al., 2006) proposed an approach (SURF) that combined the Hessian matrix-based measure for the detector and Haar-wavelet responses for the descriptor, resulting in a 64-dimension feature representation. PCA-SIFT proposed in (Ke & Sukthankar, 2004) reduced the dimensionality of descriptor to the range of 36, while remaining a comparative performance to the original SIFT. The key of PCA-SIFT is to apply the standard Principal Components Analysis technique to the gradient patches extracted around local features, therefore yielding a compact feature representation. However, the PCA-SIFT needs an offline stage to train and estimate the covariance matrix used for PCA projection. This typically requires the system to collect and train a large, diverse collection of images prior to use, (it often needs to re-train and re-estimate the covariance matrix when the image database is expanded or the scenes have significant changes), thereby impeding its widespread use and benefits.

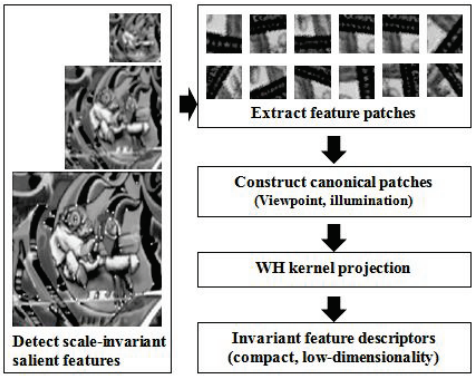


Fig. 6. CDIKP algorithm structures

This Section presents our efforts in developing an efficient local feature and its invariant descriptor for scene recognition. Our main contributions lie in a novel approach that uniquely combines the scale-invariant feature detection with a robust kernel-based representation technique to produce highly efficient feature representation. We named the approach Compact Descriptor through Invariant Kernel Projection (CDIKP). The produced feature descriptors are highly-compact (20-Dimension) in comparisons to the state-of-the-art (e.g. SIFT: 128-D, SURF: 64-D, and PCA_SIFT: 36-D), do not require any pre-training step, and show superior advantages in terms of distinctiveness, robustness to occlusions, invariance to scale, and tolerance of geometric distortions.

3.1 Approach

Figure 6 depicts the main steps of the CDIKP approach, which are detailed in following sections.

3.1.1 Scale-invariant Feature Detector

The approach selects multi-scale salient features/regions with the scale-invariant detector, similar as (Lowe, 2004) where 3D peaks are detected in a DoG scale-space. The peaks in a DoG pyramid have been shown to provide the most stable interest regions when compared to a range of other interest point detectors.

Three spatial filters are used in the detector. First, a high frequency-passed filter is employed to detect all the candidate features with local maximum responds in the DoG pyramid. The second filter is a distinctiveness filter that removes the unstable features usually lying along the object edges or linear contours. The third filter is an interpolation filter that iteratively refines the feature locations to sub-pixel accuracy. Finally, the dominant orientation and scale are computed and assigned to each detected feature. The dominant orientation and scale will be used for view normalization to achieve viewpoint invariant.

3.1.2 Scale-invariant Feature Detector

Discrimination power is an important factor required for object recognition with high data variability. We base our feature descriptor on the projection kernel scheme described in above Section 2, because the projection kernel techniques have demonstrated strong discrimination performance and they are well established analytical tools that are useful in variety of contexts including discriminative classification, scene recognition and categorization. Another attractive feature of the projection kernel techniques is their innate data compaction that can efficiently map high dimensional data to a compact representation with much lower dimensionality. This is a very attractive property for image description from which we could produce compact, lower-dimensional descriptors.

Choosing an appropriate kernel function is a key for efficient projection kernel schemes. As stated above in Section 2, two important factors have to be considered: the kernel functions should be ordered in a way such that the lower bound becomes tight after only a small number of projections, and the kernels should be efficient to enable fast computation.

We decided to use Walsh-Hadamard (WH) kernel because of its good performances in discrimination and computational efficiency. Mathematically, the WH kernel vectors can be recursively constructed as a set of orthogonal base vectors made up entirely of 1 and −1. Computation of the WH transform involves only integer additions and subtractions. Given an image patch of size $k \times k$, its WH transform is computed by projecting the patch onto k^2 WH kernel vectors. It has been shown in (Hel-Or Y. & Hel-Or H., 2005) and also confirmed by our experiments that the first few WH projection vectors can capture a high proportion of information contained in the image (Figure 2, 7). These unique properties of WH kernel projection lead us an efficient tool to build compact descriptors.

3.1.3 Generate Descriptors with WH Projections

WH kernel projection, however, does not naturally have the important property of geometry invariance, thus it cannot handle geometric distortions caused by viewpoint or pose changes. We solve this problem by performing a viewpoint normalization step on the basis of the feature's dominant orientation and scale.

Constructing the canonical views of features is relatively simple and fast. We first extract local patches centered at the feature locations from the Gaussian pyramid constructed in the above step of feature detection. The size of patch varies with the scale at which the feature was detected. Under the assumption of local planarity, a new canonical view of the local patch (with fixed size and scale) is synthesized by image warping with the feature's dominant orientation and scale. This corresponds to a regular re-sampling process in an affine space. Note that the size of the canonical patch is fixed and has to be in the power of 2, as required by WH transform. Our extensive experiments show that the size of 32x32 gives the optimal results (Figure 8).

To reduce the effect of photometric changes, we use gradient for each patch in WH transform. We have evaluated several gradient computation and forms, and found that the Gaussian weighted first order derivatives of pixel intensity along horizontal and vertical directions seemed to yield the most robust results to compensate the substantial changes of illumination. Thus, we first calculate the first-order derivatives in x and y directions within a local patch, and then weight the directional derivatives using a weighted Gaussian kernel. In this way, we obtain a pair of x-y gradient maps for each local patch that is canonically normalized to viewpoint and photometric variances.

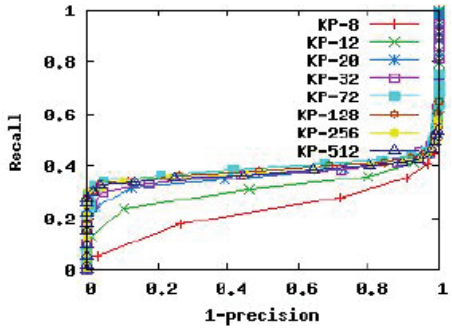


Fig. 7. Impact of the lengths of WH projection vectors on feature matching.

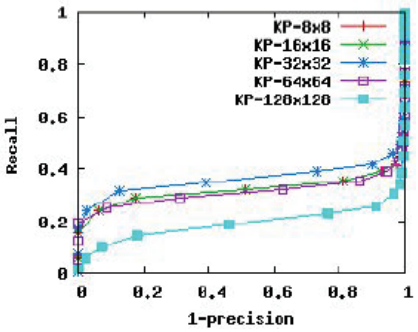


Fig. 8. Impact of patch sizes on feature match performance.

We then use the WH kernel projection to extract significant components contained in the local patches to generate feature descriptors. Since we obtain two 1024-element gradient maps for each patch/feature, we apply the WH transform twice to the gradient maps: one for x-component and one for y-component. Finally, the first 10 projection vectors of each WH transform are extracted and combined to produce a 20-dimension feature descriptor that is compact, distinctive, and viewpoint and illumination invariant.

3.2 Performance Evaluation and Applications

We evaluated the proposed approach using various datasets including synthesized data, a standard evaluation set, and our own datasets acquired under varying circumstances. We evaluated the effectiveness of our approach, in comparison to other descriptors, in the terms of distinctiveness, robustness and invariance.

3.2.1 Synthesized Data Evaluation

We collected a dataset of images, and intentionally distorted them with various geometric and photometric transformations. For a pair of test images, we ran the CDIKP algorithm to automatically select distinctive features, generate descriptors, and find the feature matches. The results were evaluated using the standard metric of recall-precision graphs. We conducted performance comparisons to standard SIFT, and PCA-SIFT. In our tests, we tried to use the same set of parameters for all the three methods.

Figure 9 shows results of the CDIKP approach to a scene under different distortions, where (a) is the matched features for the original image being rotated 70 degree; (b) for 250% scaling; (c) for 0.4-x, 0.1-y shearing; and (d) for 100% illumination change and adding 15% Gaussian noise.

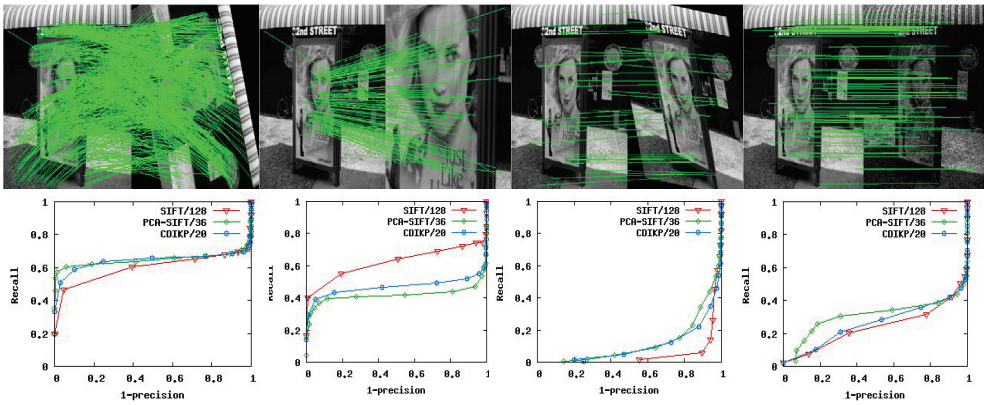


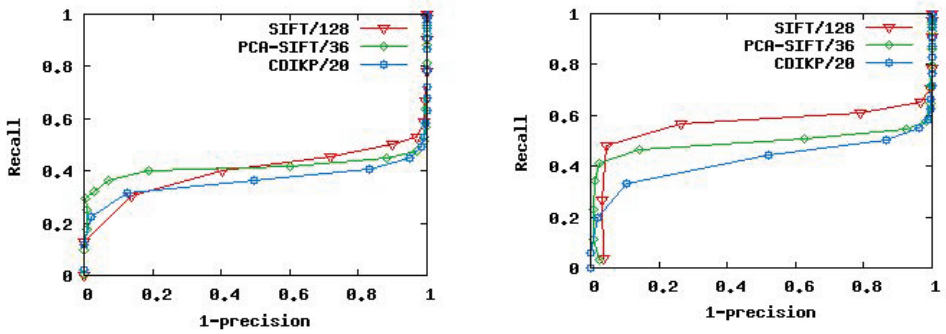
Fig. 9. Performance evaluation under different imaging and viewpoint variances

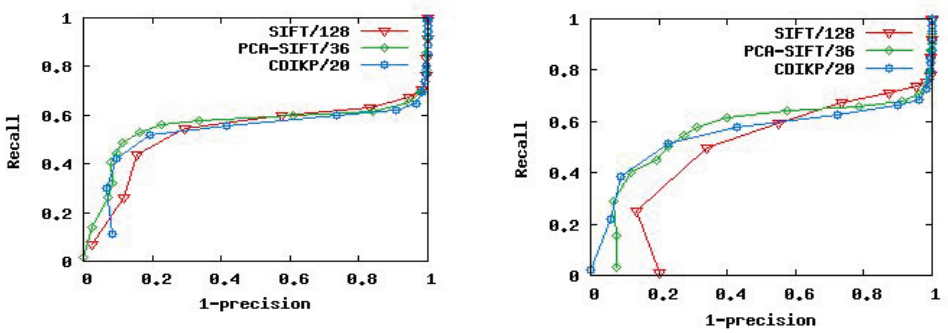
3.2.2 Standard Test Dataset with Ground Truth

Fig. 1. Evaluation with INRIA test dataset

We tested our approach using the INRIA dataset (Mikolajczik & Schmid, 2003). These are images of real scenes with recovered deformation parameters used as test ground. Figure 10 shows the results for several cases: (a) rotation and scale (Boat), (b) viewpoint changes (Wall), (c) image blur (Bikes), and (d) lighting changes (Leuven). We can see from these results that the CDIKP descriptor remains a very comparative performance, sometimes outperforms SIFT in recall for the same level of precision. Meanwhile, it is more compact and efficient to compute.

3.2.3 Scene Recognition Application





(c) (d)

Fig. 10. Evaluation with INRIA test dataset

We used the approach for object recognition application intending to the content-based image retrieval on mobile-platform. Figure 11 demonstrates the scenario of applying the approach to automatically localize and recognize various commercial logos in nature mobile environments. The application uses image recognition and search techniques that allow users to capture designated images such as product advertisements and quickly match them to the vendor's databases to obtain detailed product information associated with the images. These examples demonstrate the value of the proposed approach for mobile multimedia applications such as product advertising and shopping.

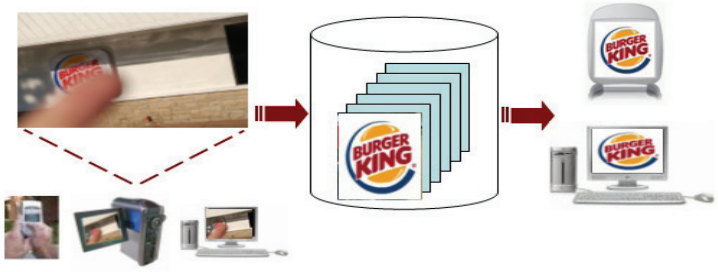


Fig. 11. Automated image matching and searching for mobile multimedia applications

4. Fast Similarity Search for High-Dimensional Dataset

Similarity search is crucial to multimedia database retrieval applications, for example, searching for correspondences between objects or retrieving multimedia contents from databases. The similarity search involves finding the most similar objects in a dataset to a query object based on a defined similarity metric. To achieve a robust and effective querying, many current multimedia systems use highly distinctive features as basic primitives to represent original data objects and perform data matching. While these feature representations have many advantages over original data including distinctiveness, robustness to noise, and invariance and tolerance to geometric and illumination distortions,

they typically produce high-dimensional feature spaces that need to be searched and processed. Methods that search exhaustively over the high-dimensional spaces are time-consuming, resulting in painfully slow evolution of such multimedia databases. Efficient search strategies are needed to rapidly and robustly screen the vast amounts of data that contain features and objects of critical interest to users' applications.

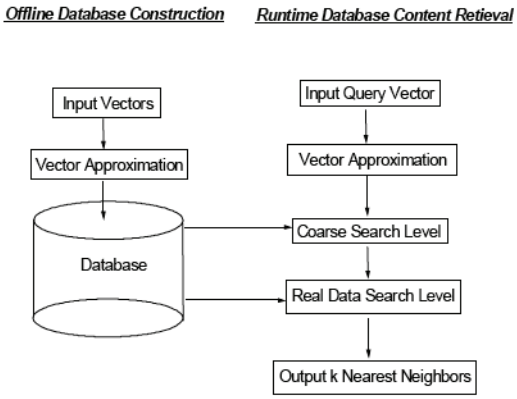


Fig. 12. Algorithmic structure FFVA approach

This session addresses the challenging problem of searching and matching high-dimensional feature sets (e.g. over 100 dimensions for each feature vector) for the applications of multimedia database retrieval and pattern recognition. Traditional tree-structure techniques hierarchically partition or cluster the entire data space into several subspaces and then use special tree structures to index objects. These types of approaches are suitable for nearest neighbor (NN) searching of datasets with low dimensionality, but their performance could rapidly degrade when directly adapting to high dimensionality. The limitations of simply modifying or adapting these techniques to highdimensional datasets are severe. This is referred to "curse of dimensionality" (Bellman ,1961) and places a practical limit on the partitioning based techniques. It has been shown in (Weber et al., 1998) and our experiments, that using the hierarchical partitioning and indexing structures for searching beyond a certain dimension becomes even worse than an exhaustive sequential-scan.

Recently, there have been great efforts in developing vector approximation (VA) techniques such as VA-File (Blott & Weber, 1997) intending to overcome the limitations of the treestructure approaches. Instead of partitioning the input data space hierarchically, the vector approximation methods directly index the objects based on linear and flat structure. While the VA approaches have several inherent problems, they demonstrate better performance in high-dimensional feature retrieval, and do not suffer from the problem of dimensionality curse.

This seccsion describes an improved vectorapproximation method, called Fast Filtering Vector Approximation (FFVA), for rapidly searching and matching high-dimensional features from large multimedia databases. FFVA is an NN-search technique that facilitates rapidly indexing and recovering the most similar matches, i.e. k-NN, in a high-dimensional database of features or spatial data. Comparing with several existing techniques including exhaustive linear scan, KD-tree (Friedman et al., 1977), Best-Bin-First (Jeffrey & Lowe, 1997),

and VA-File (Blott & Weber, 1997, Weber et al., 1998), the FFVA has demonstrated better performances in terms of query exactness, data access rate, query speed, and memory requirement.

4.1 FFVA for Similarity Search

FFVA is an improved version of VA-File method to achieve fast vector approximation and indexing. Figure 12 illustrates the structure of FFVA and its major components for an efficient nearest neighbor search.

The basic data structure of FFVA and standard VAFile method is a space-partition-table (SPT). Each dimension of input feature vectors is quantized as a number of bits used to partition it into a number of intervals on that dimension. In the whole vector space, each rectangle cell with the bit-vector representation approximates the original vectors that fall into that cell, resulting in a list of vector approximations of the original vectors. It is noteworthy that our FFVA method clusters the original vectors to the corners of SPT cells, thereby enable us to use a list of corners, instead of cells, to approximate the original vectors. Such strategy is efficient for the following fast lower bounds filtering.

There are two major levels involved in FFVA NNsearch: 1) coarse search level to sequentially scan the approximations list and eliminate a large portion of data, and 2) real data search level to calculate accurate distances of resultant candidates and decide the final knearest neighbors.

In previous works, lower-bound (the Euclidean distances from the SPT cell's nearest corner to a query point) are used for the coarse search. Experiments show that the approximation quality and computation cost of the bound-distances determinate the performance of the entire searching system (Tuncel et al., 2002; Ferhatosmanoglu et al., 2000).

Figure 13 shows the total time (sum of 900 queries) of a typical VA-file method query using real data sets (128-D features). According to the graph, the calculation of lower-bounds takes around 90% of the total querying time. This result is consistent with the experimental results in (Ciaccia & Patella, 2000). Therefore, a more efficient method for lower-bounds computation is essential for improving the entire searching performance.

In the coarse search level, only the vector approximations are accessed and *block distance* is used as similarity metric, which is calculated by employing the Manhattan distance of corresponding corner points of two SPT cells:

$$BD = \sum_{i=1}^{n} | v_{1i} - v_{2i} | \tag{6}$$

where v_{1i} and v_{2i} are the coordinates of two corresponding corner points in a n-dimension space.

Table 1: FFVA algorithmic structure

Input: query and database vectors

Output: *kNNQ* containing k-nearest neighbors

/* **Note:** *kNNQ* is a list containing the k-nearest neighbors found so far, sorted according to ascending order of exact distance from a query point */

1. Calculate or load (if pre-computed) the approximations of database vectors: *aprov* and query point: *aproq*;
2. Initialize *kNNQ*;
3. *max_db* = maximum possible value;
4. For each approximation *aprov* of database vectors

```
{
    current_bd = block distance between aprov and
     aproq;
     if (current_bd < max_db)
     {
         Calculate the corresponding actual distance;
         Insert this vector into kNNQ, if it is closer
         to query point than the last element in
         kNNQ;
         Update max_db to be the distance from
         query point to the last element of k-NN
         found so far;
     }
}
```

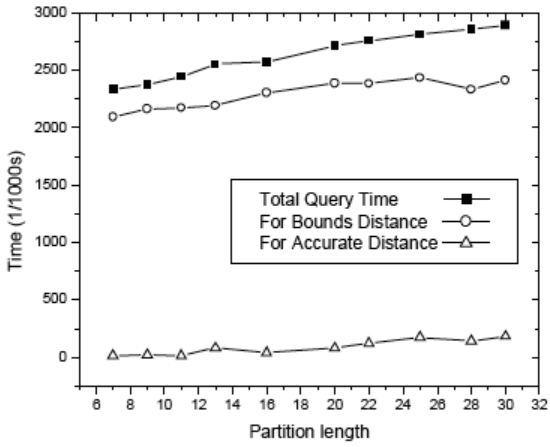


Fig. 13. Query time of VA-File

Let "*max_bd*" represents the longest exact distance (squared Euclidean distance) from a query point to current k-nearest neighbors. Since in our experiments, all the vector coordinates are integers so the exact distance is strictly lower-bounded by the block distance, whenever we encounter an approximation whose block distance from the query point is larger than *max_bd*, it can be guaranteed that at least *k* better candidates have already been found. Therefore, we can eliminate data with block distances larger than *max_bd*. Updating max_bd is also fast, as we dynamically sort and maintain the k-NN structure.
Only those candidates with block distance no larger than *max_bd* will enter the real data search level. In this level, their original vectors are accessed in order to calculate their exact distances. If the exact distance turns out to be shorter than any of current k-NN distance, the k-NN as well as *max_bd* will be updated.
Table 1 outlines the algorithmic structure of the FFVA approach.

4.2 Experimental results

In this section, we provide extensive performance evaluation and comparison of the proposed FFVA approach with four commonly used k-NN search techniques: exhaustive linear scan, standard KD-tree, BBF, and standard VA-File. The tests cover: search accuracy, data access rate, query speed, and memory requirement.
Both synthetic and real data are used in our experiments. The real data sets are large number of high-dimensional feature vectors (128-dimension for each feature) generated from a range of real images containing various object and scenes. SIFT (Scale Invariant Feature Transforms) approach (Lowe, 2004) was used to extract the image features described as 128- dimension vectors for feature matching.
In our test, the partition length for VA-File and FFVA methods is 15. For BBF, the size of buffer for "best bins" is 25 and the limit number of examined nodes is 100. During the similarity search, top two nearest neighbors (2-NNs) were returned. To evaluate the matching correctness, we used the distance ratio as evaluation criteria, that is: the second closest neighbor should be significant far away from the closest one. A threshold value of 0.6 was used throughout the experiments.

4.2.1 Search accuracy

In the accuracy test, we randomly pick up 1000 vectors from a synthetic database served as query vectors (test set). Since the test set is a subset of the database, ideally a perfect match of a query vector should have zero distance.

Dimension	60	70	80	90	100	110	115
Correct matches	1000	999	1000	1000	998	1000	1000
Dimension	120	125	130	135	140	145	150
Correct matches	1000	1000	1000	1000	1000	1000	1000

Table 2. Matching accuracy of FFVA

Table 2 summarizes the number of correct matches (out of 1000 queries) of the proposed FFVA method to the synthetic database. Gaussian noise (variance = 0.3) is added to the

uniform distributed vectors in the test set. We also conducted tests on various real data sets. Overall, the matching performance is consistent with the results of above synthetic data.

4.2.2 Search accuracy

Data access rate is an important aspect to evaluate the effectiveness of an approach for very large highdimensional database retrieval problem. KD-tree and BBF approaches utilize hierarchy tree structure to skip the nodes that are not along the searching path. FFVA and VA-File methods use compact vector approximations to avoid accessing the majority of the original database vectors.

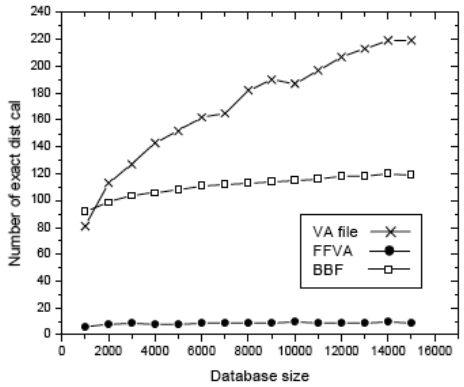


Fig. 14. Data access rate test (results are the averages of 1000 queries)

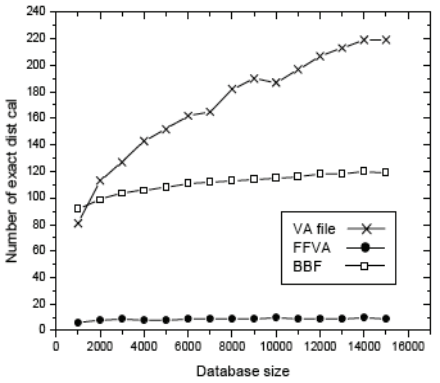


Fig. 15. Query speed test (synthetic data)

Figure 14 illustrates the test results of data access rates for three methods. The FFVA NN-search method clearly demonstrates the best performance. Its lowerbound is much tighter than that of standard VA-File method. As a result the proposed lower-bound computation based on block distance is efficient in reducing the amount of necessary data access and distance calculation. In this experiment of synthetic data, less than 10 exact distance calculations are needed for FFVA to find the 2-nearest neighbors among 15,000 120-dimension feature vectors, while other two methods (BBF and standard VA-file) spend 10-20 times more for the exact distance calculations.

4.2.3 Query speed

We tested the query speed of various data sets. In this test, we assume that all the feature vectors and their data structures are loaded into main memory (i.e. full memory access) to perform NN-search. The total query time also include the times spent on the tree construction (BBF) and vector approximation (VA-file and FFVA) processes.

Figure 15 shows the testing results (total query time of 100 queries) of synthetic data with dimension fixed at 100. Under full memory access assumption, BBF demonstrates the best performance among the three approaches when the database excesses certain size.

Figure 16 shows the test results of real data. The test data are collections of 128-dimension image features extracted from image pairs using SIFT approach. The search time per query

is total query time divided by the number of features in the query image. An exact NN-search using hierarchy structure becomes even slower than exhaustive sequential-scan when the dimensionality is high (Weber et al., 1998). Standard VA-file approach spends over much time for bounds-distances calculation when facing non-uniform real data, while FFVA and BBF demonstrate better and comparable performances.

4.2.4 Memory requirement

Memory usage has to be considered in developing an effective algorithm for search of large databases on mobile platform. BBF or similar tree-structure approach typically needs to load and store entire data sets into system memory for online processing: constructing tree structure, iteratively tracing the tree branches, and searching for the optimal tree nodes. This strategy apparently is impractical for searching very large databases when the data size easily overwhelms the limit of available memory space.

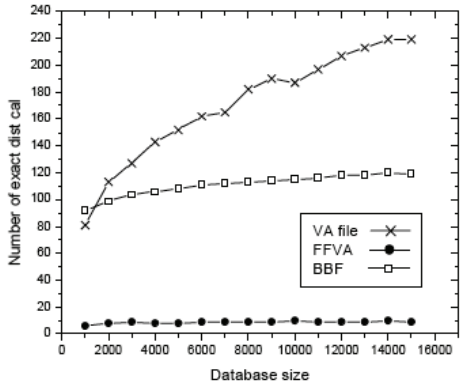


Fig. 16. Query speed test (real data)

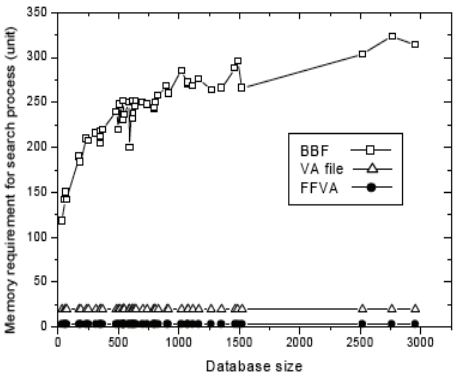


Fig. 17. Memory requirement test (note: one unit approximately equals to 512 bytes)

One of the major advantages of FFVA and VA-File is their low memory requirement provided by the flat and linear SPT data structure. Figure 17 shows the memory usages of three methods, BBF, VA-file and FFVA to query a real feature database containing 3,000 feature vectors.

We also tested the relationship among query time, data dimensionality and memory blocks (i.e. memory pages). These results clearly indicated that the memory usages of FFVA are far more effective than that of BBF. Therefore, FFVA is suitable for the application of large database retrieval or the systems that have limited memory spaces such as mobile computing devices.

5. Application: Augmented Museum Exhibitions

This Section presents an application system, called Augmented Museum Exhibitions that combines the mobile computation, Augmented Reality and image matching/recognition techniques to demonstrate the effectiveness and utility of the presented technologies.

Today, museums commonly offer a simple form of virtual annotation in the form of audio tours. Attractions ranging from New York's Museum of Modern Art to France's Palace of Versailles provide taped commentary, which visitors may listen to on headsets as they move from room to room. While a fairly popular feature, audio tours only offer a linear experience and cannot adjust to the particular interests of each visitor.
An AR museum guide is able to add a visual dimension. Pointing a handheld device at an exhibit, a user might see overlaid images and written explanations. This virtual content could include background information, schematic diagrams, or labels of individual parts, all spatially aligned with the exhibit itself (Figure 18).
The virtual content could be interactive as well. The AR content for a piece of art, for example, might allow the user to switch between background information about the artist, a description of the historical context of the piece, and a list of related works on display in the museum. An application for a science museum could display the inner workings of a machine, with the user able to adjust the level of details.

5.1 Augmented Reality and Related Works

Augmented Reality (AR) is a natural platform on which to build an interactive museum guide. Rather that relying solely on printed tags or pre-recorded audio content to aid visitors, an AR system can overlay text and graphics on top of an image of an exhibit and thus provide interactive, immersive annotations in real-time.
Graffe, et al., for example, designed an AR exhibit to demonstrate how a computer works (Grafe et al., 2002). Their system relies on a movable camera that the user can aim at various parts of a real computer. A nearby screen then displays the camera image annotated with relevant part names and graphical diagrams. Schmalstieg and Wagner presented a similar system using a handheld device (Schmalstieg & Wagner 2005). As the user walks from place to place, AR content provides information not only about the current exhibit, but also acts as a navigational tool for the entire museum.

Fig. 18. An example of how a museum visitor might use an augmented exhibit implementation on a cell phone.

Both of the above systems rely on printed markers for recognition and tracking. That means for every object to be incorporated in the AR application, a marker needs to printed and placed in the environment in such a way that it is always clearly visible. If at any time, no marker is visible inside the camera's field of view, then no AR content can be rendered. This

can lead to frustrations when a particular exhibit is partially or even fully visible but its associated marker is obscured, perhaps because another visitor is standing in the way.
Our work seeks to avoid the need for artificial markers by recognizing the target objects themselves, in this case 2D drawings and paintings. Thus, as long as an exhibit is visible to the user the application can render the associated AR content.
A number of approaches have been proposed for building natural feature based AR (Neumann & You, 1999; Coors et al., 2000; Simon et al., 2000). In this presented work, we use a modified MVKP for real-time image matching as described in Section 2.
Our information retrieval system is based on a simplified version of the multi-tier client/server architecture described in (Mooser et al., 2007). The user interacts with a client application that recognizes exhibits and sends their unique ID numbers to the server. The server then responds with all of the relevant data for that exhibit. Thus, even with a large number of clients, content for the entire application can be controlled from a single server.

5.2 System Overview

The vision-based augmented exhibition system we proposed is composed of four major components:
Acquiring query image: the system accepts query images captured from simple camera attached to a mobile device. It can also accept single JPG image or video clip.
Adapted MVKP: there are two main tasks for this component. First, given a painting, it builds the feature set for the painting. Low-resolution images (around 200x150) are enough and there is no need to extract the painting from the image in order to remove the background. Second, given a query image, it matches the painting to the database. If one of the trained paintings is matched, it establishes a feature correspondence between query image and database image. The output is the painting's ID and 3D pose with respect to the camera.
Remote Server: After the server receives the painting ID through a local Internet, it retrieves the corresponding information from its database (XML file), which it sends back to the client.
Overlaid Display: The client application, upon receiving the associated annotations from the server, displays them as overlaid virtual content on top of the current camera image. The virtual contents include the name of the painting and the artist as well as a URL pointing to related information on the Internet. The visitor can click on the URL, which will open a web browser and bring up even more information.

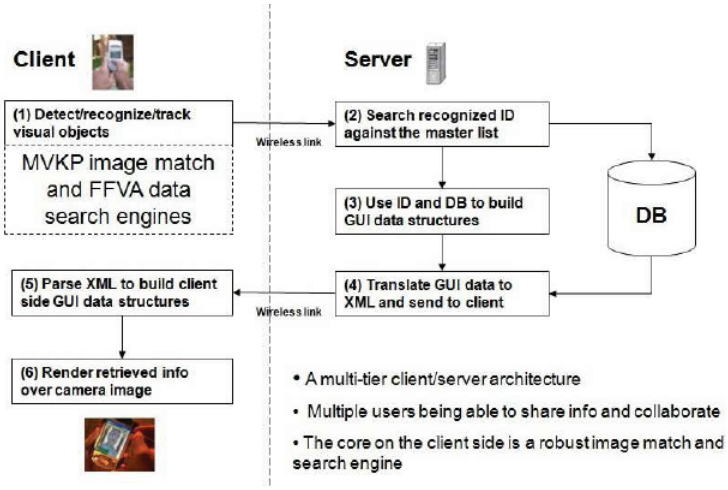


Fig. 19. System overview of the developed augmented exhibition system

Figure 19 illustrates the overall structure of our augmented exhibition system. Two major components: adapted MVKP and remote server are described in the following section.

5.3 Modified MVKP and Information Retrieval

Based on the practical requirements of the application, we chose to use MVKP as the foundation for painting recognition and 3D pose recovery. As described in Section 2, the major advantages of MVKP are: (1) robustness to lighting changes and image noises, (2) invariance to geometric distortion, (3) ability to handle complex conditions like occlusion and cluttered background, (4) sufficient accuracy for pose recovery, (5) particularly good for rigid planar objects like art paintings, (6) real-time, reliable performance, and (7) feature distinctiveness when considering a large feature database. All of these advantages make it ideal for the application of vision-based augmented exhibition system.

We also introduce several important adaptations to the MVKP method to better accommodate the requirements of AR.

In our augmented exhibition system, the outputs of MVKP method are a painting's ID and its 3D pose. The ID is then sent to a remote server through a WiFi LAN connection to retrieve the related complementary information to be displayed as virtual content on top of the painting.

5.3.1 MVKP Adaptations

Originally, the MVKP method was used to find correspondences between two input images, which means: (1) there is no need to detect the existence of interest object and there is no search among multiple objects involved, and (2) thresholds like the one in the distance ratio criteria can be set manually since user knows the query image beforehand. However, we have to make several important adaptations to the original MVKP method to meet the application requirements of augmented exhibition system.

First, for the augmented exhibition system, there can be hundreds of various painting displayed in the museum and some of them are high-textured paintings and some are not. Figure 20 shows two representative paintings. The right painting returns 50% more feature points than the left one after running the same detector. For those painting with low texture, the number of feature points returned by the detectors will also be low, which means the threshold in distance ratio criteria should also be low for it to work properly. Furthermore, there are other factors like feature distinctiveness of a specific painting that also affect the same threshold. And there are thresholds sharing the same dilemma other than distance ratio, for example, those thresholds in RanSac algorithm.

To tackle this problem, we introduce Dynamic Threshold to MVKP method. Take the threshold of distance ratio criteria for example. First we set up a global goal about how many correspondences we'd like to keep after applying the distance ratio filter. At the run time, we periodically (10 times in our experiments) check the number of correspondences the method has found so far, compare it with the global goal, and adjust the threshold accordingly. Experimental results show that, with the help of the automatic adjusted thresholds, for high textured painting we can keep the number of correspondences low and accordingly the computational cost low. For low textured painting we will still have enough correspondences to recognize the painting and recover its pose.

Second, the user of the augmented exhibition system can point the camera to anywhere inside the museum where the query image might contain no painting at all. If there is one, we need to search and decide which painting it is. Based on our experiments, we found the size of the largest consistent correspondences set after running the RanSac is the best criteria to determine which painting, if any, is contained inside the query image.

Last but not the least, for image matching methods based on local features, especially when the query image has significant view point and lighting changes, consistent check methods like RanSac are necessary in order to combine the global information. One problem involving RanSac in AR system is stability. RanSac randomly chooses three correspondences to fit an affine transformation, and for performance consideration, terminates after a limited number of iterations. Therefore, there is no guarantee that the correct affine transform can always be found. Failure of RanSac typically means one or two frames "target lost", which should be avoided for AR applications.

To solve this problem, we assume that when a certain painting is detected in one frame by the system, it is more likely that the same painting will appear in the following frames. In practice, after one painting is detected, the system will focus only on that painting's features in the following frames even after it encounters a RanSac failure. The system will revert to general search mode only when RanSac process fails a certain number of consecutive times. Through this implementation technique, we achieve stable and smooth displays for the augmented exhibition system. Besides this simple technique, every frame of the input is processed independently and there is no tracking technique involved in our current system.

5.3.2 Information Retrieval

Our system is based on a client/server architecture, where the client performs all of the visual processing and recognition and the server maintains a database of all known exhibits and their associated information. When the client positively identifies an exhibit, it sends a unique ID to the server. The server looks up the ID in its database and retrieves the relevant data, which may include the name of the work, the name of the artist, and possibly links to

related web pages. It sends this data back to the client to be displayed over the current camera image of the exhibit.

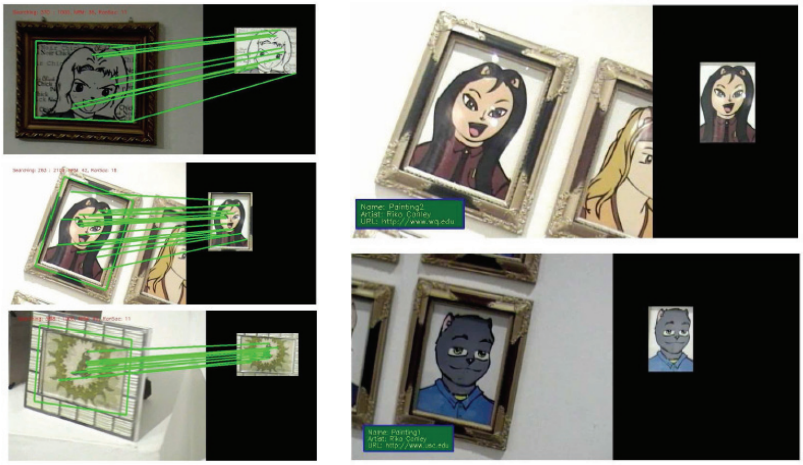


Fig. 20. Real museum test: (left) with image correspondences and recovered pose displayed under various lightings and viewpoints, and (right) with retrieved information displayed.

The advantage of using a client/server model is that changes to the underlying information can be changed in one place. Whenever a client application recognizes an exhibit that it has not seen recently, it sends a new request to the server to retrieve the latest data. Due to the ready availability of wireless LAN technologies such as WiFi, it is easy to have a mobile client make periodic request to a server. Only one send-receive round trip is needed for each exhibit, so the client and server do not need to maintain a persistent open communications channel.

5.4 Results

We implemented the system and conducted experiments with both synthesized data and real data.

Figure 20 shows sample results on real paints. In our test, we capture videos of those paints from a gallery at the USC School of Fine Art and process the videos in our augmented exhibition system with four painting trained. During the video capture, we intentionally includes many challenging cases like out-of-plane rotation of the camera, moving highlights on the painting, sudden change of illumination, intense shaking of the video camera, etc. Overall, our augmented exhibition system demonstrates fast and reliable performance in a real exhibition environment.

6. Conclusions

In this chapter, we presented several advanced image matching and recognition technologies, of particular emphasis on mobile multimedia applications that are feasible with current or near term technology and applications. The presented work represents the latest state-of-arts in this area. We introduced a high-performance matching technique, MVKP suitable for real-time

multimedia applications. Together with the MVKP, we developed an efficient image search technique that can dramatically improve on previous approaches over hundred-times faster for recognition of objects from a large library of database. Targeting the mobile multimedia applications, we also developed a highly-compact image feature description and matching technique. Finally, we presented an application system, called Augmented Museum Exhibitions that combines the mobile computation, Augmented Reality and image matching/recognition techniques to demonstrate the effectiveness and utility of the presented technologies.

7. Acknowledgments

The presented work are supported by several grants including the Center of Excellence for Research and Academic Training on Interactive Smart Oilfield Technologies (CiSoft), a joint University of Southern California-Chevron-initiative; and the Aerospace Institute for Engineering Research (AIER), a joint research initiative made up of the USC Viterbi School of Engineering, Korea Aerospace University, Inha University, and global corporations Airbus and Korean Air.
This work made use of Integrated Media Systems Center (IMSC) Shared Facilities supported by the National Science Foundation under Cooperative Agreement No. EEC-9529152. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

8. References

Bay H., Tuytelaars T. & Gool L. V. (2006). Surf: Speeded up robust features, *Proceedings of European Conference on Computer Vision,* Vol. 110, No. 3, pp. 346-359, May, 2006

Bellman R. (1961). Adaptive Control Processes: *A Guided Tour*. Princeton University Press

Ben-Artzi G., Hel-Or H. & Hel-Or Y. (2004). Filtering with Graycode kernels, *Proceedings of the* 17th *International Conference on Pattern Recognition*, vol. 1, pp.556-559, December, 2004

Blott S. & Weber R. (1997). A Simple Vector-Approximation File for Similarity Search in High-dimensional Vector paces. *Technical Report 19, ESPRIT project HERMES (no.9141)*, March 1997

Boffy A., Tsin Y. & Genc Y. (2006). Real-Time Feature Matching using Adaptive and Spatially Distributed Classification Trees. *Proceedings of the British Machine Vision Conference*, 2006

Ciaccia P. & Patella M. (2000). PAC Nearest Neighbor Queries: Approximate and Controlled Search in High-Dimensional and Metric Spaces, *Proceedings of the International Conference on Data Engineering*, 2000

Coors V., Huch T. & Kretschmer U. (2000). Matching buildings: Pose estimation in an urban environment, International Symposium on Augmented Reality, pp. 89 – 92, 2000

Elad M., Hel-Or Y. & Keshet R. (2000). Pattern detection using maximal rejection classifier. *Proceedings of the International Workshop on Visual Form*, pages 28–30, May 2000

Ferhatosmanoglu H., Tuncel E., Agrawal D. & Abbadi A. E. (2000). Vector Approximation based Indexing for Nonuniform High Dimensional Data Sets, *Proceedings of the ACM Conference on Information and Knowledge Management*, 2000

Ferrari V., Tuytelaars T. & Luc V. G. (2005). Wide-baseline Multiple-view Correspondences. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June 2003.

Friedman J. H., Bentley J. L. & Finkel R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209-226

Grafe M., Wortmann R. & Westphal H. (2002). AR-based Interactive Exploration of a Museum Exhibit, *International Workshop on Augmented Reality Toolkit*, pp. 5 – 9, 2002

Hel-Or Y. & Hel-Or H. (2005). Real-time pattern recognition using projection kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 2005

Jeffrey S. B. & Lowe D. (1997) Shape indexing using approximate nearest-neighbor search in high-dimensional spaces, *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June 1997

Keren D., Osadchy M. & C. Gotsman. (2001). Antifaces: A novel, fast method for image detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):747–761, 2001

Ke Y. & Sukthankar R. (2004) PCA-SIFT: A more distinctive representation for local image descriptors. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 511-517, June, 2004

Lepetit V., Pilet J. & Fua P. (2004). Point matching as a classification problem for fast and robust object pose estimation. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June 2004

Lepetit V., Lagger P. & Fua P. (2005) Randomized trees for real-time keypoint recognition. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June, 2005

Lowe D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Visions,* Vol. 60, No. 2, page numbers (91-110), ISBN 0920-5691

Mikolajczik K. & Schmid C. (2003) A performance evaluation of local descriptors, *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 257–263, June 2003

Mooser J., Wang L., You S. & Neumann U. (2007). An augmented reality interface for mobile information retrieval, *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 2226 – 2229, 2007

Neumann U. & You S. (1999). Natural feature tracking for augmented reality, *IEEE Transactions on Multimedia*, pp. 53 – 64, 1999

Ozuysal M., Lepetit V., Fleuret F. & Fua P. (2006) Feature Harvesting for Tracking-by-Detection. *Proceedings of European Conference on Computer Vision,* May 2006

Schmalstieg D. & Wagner D. (2005). A Handheld Augmented Reality Museum Guide, *IADIS Mobile Learning*, 2005.

Schmid C. & Mohr R. (1997) Local grayvalue invariants for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 530–534, May 1997

Simon G., Fitzgibbon A. & Zisserman A. (2000). Markerless tracking using planar structures in the scene, *International Symposium on Augmented Reality*, pp. 120– 128, 2000

Tuncel E., Ferhatosmanoglu H. & Rose K. (2002). VQ-Index: An Index Structure for Similarity Searching in Multimedia Databases, *ACM Multimedia*, 2002

Tuzel O., Porikli F. & Meer P. (2006). Region Covariance: A Fast Descriptor for Detection and Classification, *Proceedings of European Conference on Computer Vision*, pp: 589-600, May, 2006

Weber R., Schek H. & Blott S. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Space, *Proceedings of the 24th International Conference on Very Large Data Bases*, pp: 194 – 205, 1998, ISBN:1-55860-566-5

Winn J. & Criminisi A. (2006) Object Class Recognition at a Glance. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June, 2006

6

Structured Max Margin Learning on Image Annotation and Multimodal Image Retrieval

Zhen Guo†, Zhongfei (Mark) Zhang†, Eric P. Xing‡ and Christos Faloutsos‡
†*Computer Science Department, SUNY at Binghamton, Binghamton, NY 13905*
‡ *School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213*
†*{zguo,zhongfei}@cs.binghamton.edu,* ‡*{epxing,christos}@cs.cmu.edu*

1. Introduction

Image retrieval plays an important role in information retrieval due to the overwhelming multimedia data brought by modern technologies, especially the Internet. One of the notorious bottlenecks in the image retrieval is the semantic gap (16). Recently, it is reported that this bottleneck may be reduced by the multimodal approach (2; 9) which takes advantage of the fact that in many applications image data typically co-exist with other modalities of information such as text. The synergy between different modalities may be exploited to capture the high level concepts.

In this chapter, we follow this line of research by further considering a max margin learning framework. We assume that we have multiple modalities of information in co-existence. Specifically, we focus on imagery and text modalities whereas the framework may be easily extended to incorporate other modalities of information. Accordingly, we assume that we have a database consisting of imagery data where each image has textual caption/annotation. The framework is not just for image retrieval, but for more flexible across-modality retrieval (e.g., image-to-image, image-to-text, and text-to-image retrieval). Our framework is built upon the max margin framework and is related to the model proposed by Taskar et al. (17). Specifically, we formulate the image annotation and image retrieval problem as a structured prediction problem where the input $\mathbf{x}$ and the desired output $\mathbf{y}$ are structures. Furthermore, following the max margin approach the image retrieval problem is formulated as a quadratic programming (QP) problem. Given the multimodal information in the image database, the dependency information between different modalities is learned by solving for this QP problem. Across-modality retrieval (image annotation and word querying) and image retrieval can be done based on the dependency information. By properly selecting the joint feature representation between different modalities, our approach captures the dependency information between different modalities which is independent of specific words or specific images. This makes our approach scalable in the sense that it avoids retraining the model starting from scratch every time when the image database undergoes dynamic updates which include image and word space updates.

While this framework is a general approach which can be applied to multimodal information retrieval in any domains, we apply this approach to the Berkeley Drosophila embryo image

database[1] for the evaluation purpose. Experimental results show significant performance improvements over a state-of-the-art method.

2. Related Work

Multimodal approach has recently received substantial attention since Barnard and Duygulu et al. started their pioneering work on image annotation (2; 8). Recently there have been many studies (3; 5; 7; 9; 14; 22) on multimodal approaches.

The structure model covers many natural learning tasks. There have been many studies on the structure model which include conditional random fields (12), maximum entropy model (13), graph model (6), semi-supervised learning (4) and max margin approach (1; 11; 18; 19). The max margin principle has received substantial attention since it was used in the support vector machine (SVM) (20). In addition, the perceptron algorithm is also used to explore the max margin classification (10).

Our main contribution is to develop an effective solution to the image annotation and multimodal image retrieval problem using the max margin approach under a structure model. More importantly, our framework has a great advantage in scalability over many existing image retrieval systems.

3. Supervised Learning

We begin with the brief review of the supervised learning in the max margin framework. Suppose that there is a probability distribution P on $\mathcal{X} \times \mathcal{Y}, \mathcal{X} \subset \mathbb{R}^n$ according to which data are generated. We assume that the given data consist of l labeled data points $(\mathbf{x}_i, y_i), 1 \leq i \leq l$ which are generated according to P. For the purpose of simplicity, we assume the binary classification problem where the labels $y_i, 1 \leq i \leq l$, are binary, i.e., $y_i = \pm 1$.

In the supervised learning scenario, the goal is to learn a function f to minimize the expected loss called risk functional

$$R(f) = \int L(\mathbf{x}, y, f(\mathbf{x}))dP(\mathbf{x}, y) \tag{1}$$

where L is a loss function. A variety of loss functions have been considered in the literature. The simplest loss function is $0/1$ loss

$$L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) = \begin{cases} 0 & \text{if } y_i = f(\mathbf{x}_i) \\ 1 & \text{if } y_i \neq f(\mathbf{x}_i) \end{cases} \tag{2}$$

In Regularized Least Square (RLS), the loss function is given by

$$L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2$$

In SVM, the loss function is given by

$$L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) = \max(0, 1 - y_i f(\mathbf{x}_i))$$

For the loss function Eq. (2), Eq. (1) determines the probability of a classification error for any decision function f. In most applications the probability distribution P is unknown. The problem, therefore, is to minimize the risk functional when the probability distribution function

[1] http://www.fruitfly.org

$P(\mathbf{x}, y)$ is unknown but the labeled data $(\mathbf{x}_i, y_i), 1 \leq i \leq l$ are given. Thus, we need to consider the empirical estimate of the risk functional (21)

$$R_{emp}(f) = C \sum_{i=1}^{l} L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) \tag{3}$$

where $C > 0$ is a constant. We often use $C = \frac{1}{l}$. Minimizing the empirical risk Eq. (3) may lead to numerical instabilities and bad generalization performance (15). A possible way to avoid this problem is to add a stabilization (regularization) term $\Theta(f)$ to the empirical risk functional. This leads to a better conditioning of the problem. Thus, we consider the following regularized risk functional

$$R_{reg}(f) = R_{emp}(f) + \gamma\Theta(f)$$

where $\gamma > 0$ is the regularization parameter which specifies the tradeoff between minimization of $R_{emp}(f)$ and the smoothness or simplicity enforced by small $\Theta(f)$. A choice of $\Theta(f)$ is the norm of the RKHS representation of the feature space

$$\Theta(f) = \|f\|_K^2$$

where $\|.\|_K$ is the norm in the RKHS $\mathcal{H}_K$ associated with the kernel K. Therefore, the goal is to learn the function f which minimizes the regularized risk functional

$$f^* = \arg\min_{f \in \mathcal{H}_K} C \sum_{i=1}^{l} L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) + fl\|f\|_K^2 \tag{4}$$

The solution to Eq. (4) is determined by the loss function L and the kernel K. A variety of kernels have been considered in the literature. Three most commonly-used kernel functions are listed in the Table 1 where $\sigma > 0, \kappa > 0, \vartheta < 0$. The following classic Representer Theorem (15) states that the solution to the minimization problem Eq. (4) exists in $\mathcal{H}_K$ and gives the explicit form of a minimizer.

Theorem 1. *Denote by $\Omega : [0, \infty) \to \mathbb{R}$ a strictly monotonic increasing function, by $\mathcal{X}$ a set, and by $\Lambda : (\mathcal{X} \times \mathbb{R}^2)^l \to \mathbb{R} \cup \{\infty\}$ an arbitrary loss function. Then each minimizer $f \in \mathcal{H}_K$ of the regularized risk*

$$\Lambda((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), \cdots, (\mathbf{x}_l, y_l, f(\mathbf{x}_l))) + \Omega(\|f\|_K)$$

admits a representation of the form

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \mathbf{x}) \tag{5}$$

with $\alpha_i \in \mathbb{R}$.

kernel name	kernel function
polynomial kernel	$K(\mathbf{x}, \mathbf{x}_i) = (\langle \mathbf{x}, \mathbf{x}_i \rangle + c)^d$
Gaussian radial basis function kernel	$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2})$
sigmoid kernel	$K(\mathbf{x}, \mathbf{x}_i) = \tanh(\kappa\langle \mathbf{x}, \mathbf{x}_i \rangle + \vartheta)$

Table 1. Three most commonly-used kernel functions

According to Theorem 1, we can use any regularizer in addition to $\gamma\|f\|_K^2$ which is a strictly monotonic increasing function of $\|f\|_K$. This allows us in principle to design different algorithms. The simplest approach is to use the regularizer $\Omega(\|f\|_K) = \gamma\|f\|_K^2$. Given the loss function L and the kernel K, we substitute Eq. (5) into Eq. (4) to obtain a minimization problem of the variables $\alpha_i, 1 \leq i \leq l$. The decision function f^* is immediately obtained from the solution to this minimization problem.
Different loss functions lead to different supervised learning algorithms. In the literature, two of the most popular loss functions are the squared loss function for RLS and the hinge loss function for SVM.

3.1 Regularized Least Square Approach

We first outline the RLS approach which applies to the binary classification and the regression problem. The classic RLS algorithm is a supervised method where we solve:

$$f^* = \arg\min_{f \in \mathcal{H}_K} C \sum_{i=1}^{l} (y_i - f(\mathbf{x}_i))^2 + fl\|f\|_K^2$$

where C and γ are the constants.
According to Theorem 1, the solution is of the following form

$$f^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i^* K(\mathbf{x}_i, \mathbf{x})$$

Substituting this solution in the problem above, we arrive at the following differentiable objective function of the l-dimensional variable $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_l]^\top$:

$$\boldsymbol{\alpha}^* = \arg\min C(Y - K\boldsymbol{\alpha})^\top (Y - K\boldsymbol{\alpha}) + \gamma \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}$$

where K is the $l \times l$ kernel matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{Y}$ is the label vector $\mathbf{Y} = [y_1 \cdots y_l]^\top$.
The derivative of the objective function over $\boldsymbol{\alpha}$ vanishes at the minimizer

$$C(KK\boldsymbol{\alpha}^* - K\mathbf{Y}) + \gamma K\boldsymbol{\alpha}^* = 0$$

which leads to the following solution.

$$\boldsymbol{\alpha}^* = (CK + \gamma\mathbf{I})^{-1} C\mathbf{Y}$$

3.2 Max Margin Approach

In the max margin approach, one attempts to maximize the distance between the data and classification hyperplane. In the binary classification problem, the classic SVM attempts to solve the following optimization problem on the labeled data.

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \xi_i \\ s.t. \quad & y_i\{\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b\} \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, \cdots, l \end{aligned} \tag{6}$$

where Φ is a nonlinear mapping function determined by the kernel and b is a regularized term.

Again, the solution is given by

$$f^*(\mathbf{x}) = \langle \mathbf{w}^*, \Phi(\mathbf{x}) \rangle + b^* = \sum_{i=1}^{l} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$$

To solve Eq. (6) we introduce one Lagrange multiplier for each constraint in Eq. (6) using the Lagrange multipliers technique and obtain a quadratic dual problem of the Lagrange multipliers.

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \mu_i \mu_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l} \mu_i \\ s.t. \quad & \sum_{i=1}^{l} \mu_i y_i = 0 \\ & 0 \le \mu_i \le C \quad i = 1, \cdots, l \end{aligned} \tag{7}$$

where μ_i is the Lagrange multiplier associated with the i-th constraint in Eq. (6).
We have $\mathbf{w}^* = \sum_{i=1}^{l} \mu_i y_i \Phi(\mathbf{x}_i)$ from the solution to Eq. (7). Note that the following conditions must be satisfied according to the Kuhn-Tucker theorem (21):

$$\mu_i (y_i (\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) + \xi_i - 1) = 0 \quad i = 1, \cdots, l \tag{8}$$

The optimal solution of b is determined by the above conditions.
Therefore, the solution is given by

$$f^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$$

where $\alpha_i^* = \mu_i y_i$.

4. Structured Max Margin Learning

In an image database where each image is annotated by several words, the word space is a structured space in the sense that the words are interdependent on each other. As shown later, the feature space of images is also a structured space. Therefore, it is not trivial to apply the max margin approach to image databases and several challenges exist. In this chapter, we focus on the max margin approach in the structured space and apply it to the learning problem in image databases.
Assume that the training set consists of a set of training instances $S = \{(I^{(i)}, W^{(i)})\}_{i=1}^{L}$, where each instance consists of an image object $I^{(i)}$ and the corresponding annotation word set $W^{(i)}$. We define a block as a subimage of an image such that the image is partitioned into a set of blocks and all the blocks of this image share the same resolution. For each block, we compute the feature representation in the feature space. These blocks are interdependent on each other in the sense that adjacent blocks are similar to each other and nonadjacent blocks are dissimilar to each other. Therefore, the feature space of images is actually a structured space.
Since the image database may be large, we apply k-means algorithm to all the feature vectors in the training set. We define VRep (visual representative) as a representative of a set of all the blocks for all the images in the database that appear visually similar to each other. A VRep is

used to represent each cluster and thus is represented as a feature vector in the feature space. Consequently, the training set becomes VRep-annotation pairs $S = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$, where N is the number of the clusters, $\mathbf{x}^{(i)}$ is the VRep object and $\mathbf{y}^{(i)}$ is the word annotation set related to this VRep object. We use $\mathcal{Y}$ to represent the whole set of words and $\mathbf{w}_j$ to denote the j-th word in the whole word set. $\mathbf{y}^{(i)}$ is the M-dimensional binary vector ($M = \|\mathcal{Y}\|$) in which the j-th component $\mathbf{y}_j^{(i)}$ is set to 1 if word $\mathbf{w}_j$ appears in $\mathbf{x}^{(i)}$, and 0 otherwise. We use $\mathbf{y}$ to represent an arbitrary M-dimensional binary vector.

We use score function $\mathbf{s}(\mathbf{x}^{(i)}, \mathbf{w}_j)$ to represent the degree of dependency between the specific VRep $\mathbf{x}^{(i)}$ and the specific word $\mathbf{w}_j$. In order to capture the dependency between VReps and words it is helpful to represent it in a joint feature representation $\mathbf{f} : \mathcal{X} \times \mathcal{Y} \to \Re^d$. The feature vector between $\mathbf{x}^{(i)}$ and $\mathbf{w}_j$ can be expressed as $\mathbf{f}(\mathbf{x}^{(i)}, \mathbf{w}_j)$ and the feature vector between $\mathbf{x}^{(i)}$ and word set $\mathbf{y}$ is the sum for all the words: $\mathbf{f}_i(\mathbf{y}) = \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}) = \sum_{j=1}^{M} \mathbf{y}_j \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{w}_j)$. In this feature vector, each component may have a different weight in determining the score function. Thus, the score function can be expressed as a weighted combination of a set of features $\mathbf{f}\mathbf{f}^\top \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{w}_j)$, where $\mathbf{f}\mathbf{f}$ is the set of parameters.

The learning task then is to find the optimal weight vector $\mathbf{f}\mathbf{f}$ such that:

$$arg \max_{\mathbf{y} \in \mathcal{Y}^{(i)}} \mathbf{f}\mathbf{f}^\top \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}) \approx \mathbf{y}^{(i)} \quad \forall i$$

where $\mathcal{Y}^{(i)} = \{\mathbf{y} | \sum \mathbf{y}_j = \sum \mathbf{y}_j^{(i)}\}$. We define the loss function $l(\mathbf{y}, \mathbf{y}^{(i)})$ as the number of different words between these two sets. In order to make the true structure $\mathbf{y}^{(i)}$ as the optimal solution, the constraint is reduced to:

$$\mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}^{(i)}) \geq \mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}) + l(\mathbf{y}, \mathbf{y}^{(i)}) \ \forall i, \forall \mathbf{y} \in \mathcal{Y}^{(i)}$$

We interpret $\frac{1}{\|\mathbf{f}\mathbf{f}\|} \mathbf{f}\mathbf{f}^\top [\mathbf{f}_i(\mathbf{y}^{(i)}) - \mathbf{f}_i(\mathbf{y})]$ as the margin of $\mathbf{y}^{(i)}$ over another $\mathbf{y} \in \mathcal{Y}^{(i)}$. We then rewrite the above constraint as $\frac{1}{\|\mathbf{f}\mathbf{f}\|} \mathbf{f}\mathbf{f}^\top [\mathbf{f}_i(\mathbf{y}^{(i)}) - \mathbf{f}_i(\mathbf{y})] \geq \frac{1}{\|\mathbf{f}\mathbf{f}\|} l(\mathbf{y}, \mathbf{y}^{(i)})$. Thus, minimizing $\|\mathbf{f}\mathbf{f}\|$ maximizes such margin.

The goal now is to solve the optimization problem:

$$\begin{aligned} \min \quad & \|\alpha\|^2 \\ s.t. \quad & \mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}^{(i)}) \geq \mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}) + l(\mathbf{y}, \mathbf{y}^{(i)}) \ \forall i, \forall \mathbf{y} \in \mathcal{Y}^{(i)} \end{aligned}$$

4.1 Min-max formulation

The above optimization problem is equivalent to the following optimization problem:

$$\begin{aligned} \min \quad & \|\alpha\|^2 \\ s.t. \quad & \mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}^{(i)}) \geq \max_{\mathbf{y} \in \mathcal{Y}^{(i)}} (\mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}) + l(\mathbf{y}, \mathbf{y}^{(i)})) \quad \forall i \end{aligned} \tag{9}$$

We take the approach proposed by Taskar et al. (17) to solve it. We consider the maximization sub-problem contained in the above optimization problem.

We have

$$\begin{aligned}\mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}) + l(\mathbf{y}, \mathbf{y}^{(i)}) &= \alpha^\top \sum_j \mathbf{y}_j \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{w}_j) + \sum_j \mathbf{y}_j^{(i)}(1 - \mathbf{y}_j) \\ &= \mathbf{d}_i + (\mathbf{F}_i\alpha + \mathbf{c}_i)^\top \mathbf{y}\end{aligned}$$

where $\mathbf{d}_i = \sum_j \mathbf{y}_j^{(i)}$ and $\mathbf{F}_i$ is a matrix in which the j-th row is $\mathbf{f}(\mathbf{x}^{(i)}, \mathbf{w}_j)$; $\mathbf{c}_i$ is the vector in which the j-th component is $-\mathbf{y}_j^{(i)}$.
This maximization sub-problem then becomes:

$$\begin{aligned}\max \quad & \mathbf{d}_i + (\mathbf{F}_i\alpha + \mathbf{c}_i)^\top \mathbf{y} \\ s.t. \quad & \sum_j \mathbf{y}_j = \sum_j \mathbf{y}_j^{(i)}\end{aligned}$$

We map this problem to the following linear programming(LP) problem:

$$\begin{aligned}\max \quad & \mathbf{d}_i + (\mathbf{F}_i\alpha + \mathbf{c}_i)^\top \mathbf{z}_i \\ s.t. \quad & \mathbf{A}_i\mathbf{z}_i \leq \mathbf{b}_i \quad \mathbf{z}_i \geq 0\end{aligned}$$

for appropriately defined $\mathbf{A}_i, \mathbf{b}_i$, which depend only on $\mathbf{y}, \mathbf{y}^{(i)}$; $\mathbf{z}_i$ is the relaxation for $\mathbf{y}$. It is guaranteed that this LP program has an integral (0/1) solution.
We consider the dual program of this LP program:

$$\begin{aligned}\min \quad & \mathbf{d}_i + \mathbf{b}_i^\top \lambda_i \\ s.t. \quad & \mathbf{A}_i^\top \lambda_i \geq \mathbf{F}_i\alpha + \mathbf{c}_i \quad \lambda_i \geq 0\end{aligned} \tag{10}$$

Now we can combine (9) and (10) together:

$$\begin{aligned}\min \quad & \|\alpha\|^2 \\ s.t. \quad & \mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}^{(i)}) \geq \mathbf{d}_i + \mathbf{b}_i^\top \lambda_i \quad \forall i \\ & \mathbf{A}_i^\top \lambda_i \geq \mathbf{F}_i\alpha + \mathbf{c}_i \quad \forall i\end{aligned} \tag{11}$$

This formulation is justified as follows. If (10) is not at the minimum, the constraint is tighter than necessary, leading to a sub-optimal solution α. Nevertheless, the training data are typically hardly separable. In such cases, we need to introduce slack variables ξ_i to allow some constraints violated. The complete optimization problem now becomes a QP problem:

$$\begin{aligned}\min \quad & \|\alpha\|^2 + C\sum_i \xi_i \\ s.t. \quad & \mathbf{f}\mathbf{f}^\top \mathbf{f}_i(\mathbf{y}^{(i)}) \geq \mathbf{d}_i + \mathbf{b}_i^\top \lambda_i - \xi_i \quad \forall i \\ & \mathbf{A}_i^\top \lambda_i \geq \mathbf{F}_i\alpha + \mathbf{c}_i \quad \forall i \\ & \alpha \geq 0 \quad inf > \lambda_i \geq 0 \quad inf > \xi_i \geq 0 \quad \forall i\end{aligned} \tag{12}$$

After this QP program is solved, we have the optimal parameters α. Then we have the dependency information between words and VReps by the score function. For each VRep, we have a ranking-list of words in terms of the score function. Similarly we have a ranking-list of VReps for each word.

4.2 Feature representation

For a specific VRep $\mathbf{x}^{(i)}$ and a specific word $\mathbf{w}_j$, we consider the following feature representation $\mathbf{f}$ between them: $(\frac{\delta_{ij}}{n_j}, \frac{n_j}{N}, \frac{\delta_{ij}}{m_i}, \frac{m_i}{M})$. Here we assume that there are N VReps and M words. n_j denotes the number of VReps in which $\mathbf{w}_j$ appears. m_i denotes the number of words which appear in VRep $\mathbf{x}^{(i)}$. δ_{ij} is an indicator function (1 if $\mathbf{w}_j$ appears in $\mathbf{x}^{(i)}$, and 0 otherwise). Other possible features may depend on the specific word or VRep because some words may be more important than others. We only use the features independent of specific words and specific VReps and we will discuss the advantage later.

4.3 Image Annotation

Given a test image, we partition it into blocks and compute the feature vectors. Then we compute the similarity between feature vectors and VReps in terms of the distance. We return the top n most-relevant VReps. Since for each VRep, we have the ranking-list of words in terms of the score function, we merge these n ranking-lists and sort them to obtain the ranking-list of the whole word set. Finally, we return the top m words as the annotation result.

4.4 Word Query

For a specific word, we have the ranking-list of VReps. we return the top n VReps. For each VRep, we compute the similarity between this VRep and each test image in terms of the distance. For each VRep, we have the ranking-list of test images. Finally, we merge these n ranking-lists and return the top m images as the query results.

4.5 Image Retrieval

Given a query image, we annotate it using the procedure in Sec. 4.3. For each annotation word j, there is a subset of images S_j in which this annotation word appears. Then we have the union set $S = \bigcup S_j$ for all the annotation words.
On the other hand, for each annotation word j, the procedure in Sec. 4.4 is used to obtain the related image subset T_j. Then we have the union set $T = \bigcup T_j$. The final retrieval result is $R = S \bigcap T$.

4.6 Database Updates

Now we consider the case where new images are added to the database. Assume that these new images have annotation words along with them. If they do not, we can annotate them using the procedure in Sec. 4.3. For each newly added image, we partition it into blocks and for each block we compute the nearest VRep in terms of the distance and the VRep-word pairs are updated in the database. This also applies to the case where the newly added images may include new word.
Under the assumption that the newly added images follow the same feature distribution as those in the database, it is reasonable to assume that the optimal parameter α also captures the dependency information between the VReps and the newly added words because the feature representation described in Sec. 4.2 is independent of specific words and specific VReps. Consequently, we do not need to re-train the model from scratch. In fact, the complexity of the update is $O(1)$. As the database scales up, so does the performance due to the incrementally updated data. This is a great advantage over many existing image retrieval systems which are unable to handle new vocabulary at all. The experimental result supports and verifies this analysis.

5. Experimental Result

While this approach is a general approach which can be applied to multimodal information retrieval in any domains, we apply this approach to the Berkeley Drosophila embryo image database for the evaluation purpose. We compare the performance of this framework with the state-of-the-art multimodal image annotation and retrieval method MBRM (9).
There are totally 16 stages in the whole embryo image database. We use stages 11 and 12 for the evaluation purpose. There are about 6000 images and 75 words in stages 11 and 12. We split all the images into two parts (one third and two thirds), with the two thirds used as the training set and the one third used as the test set. In order to show the advantage discussed in Sec. 4.6, we use a smaller training subset (110 images) to obtain the optimal parameter α. For these 110 images, there are 35 annotation words. Then we use the test set for evaluation. This experiment result is shown as "Our Framework (1)" in the figures. Then we add the remaining training images to the database and use the test set for evaluations again. This experiment result is shown as "Our Framework (2)" in the figures. When the new images are added to the image database, the new annotation words along with them are also added to the image database.

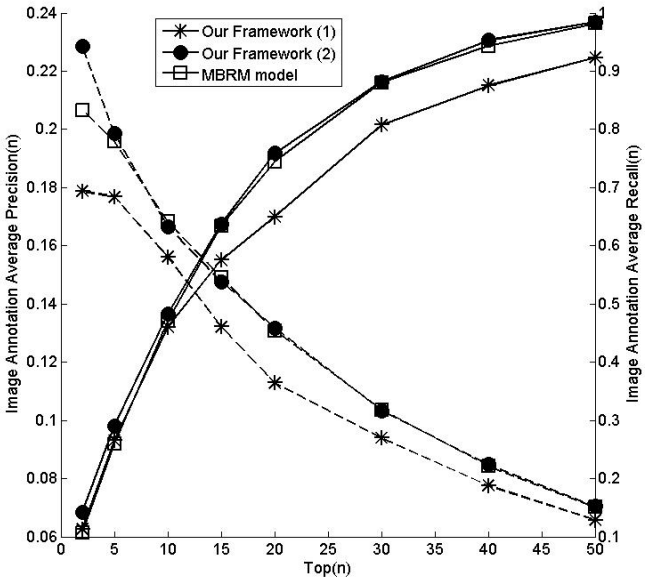


Fig. 1. Evaluation of image annotation between our framework and MBRM model.

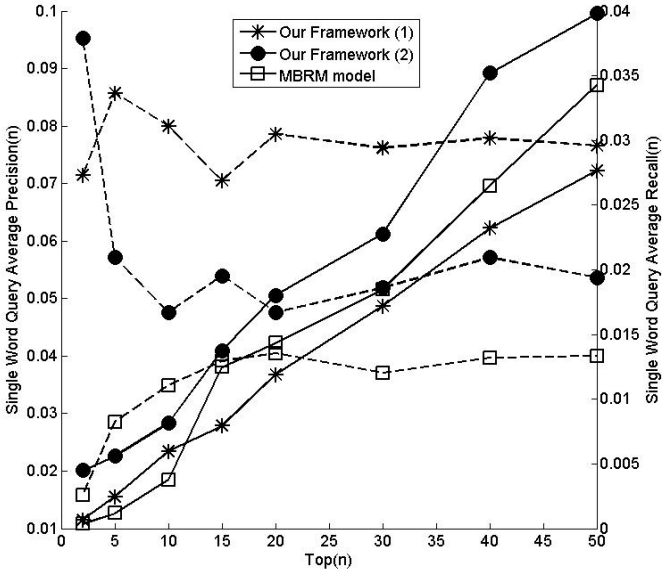


Fig. 2. Evaluation of single word query between our framework and MBRM model.

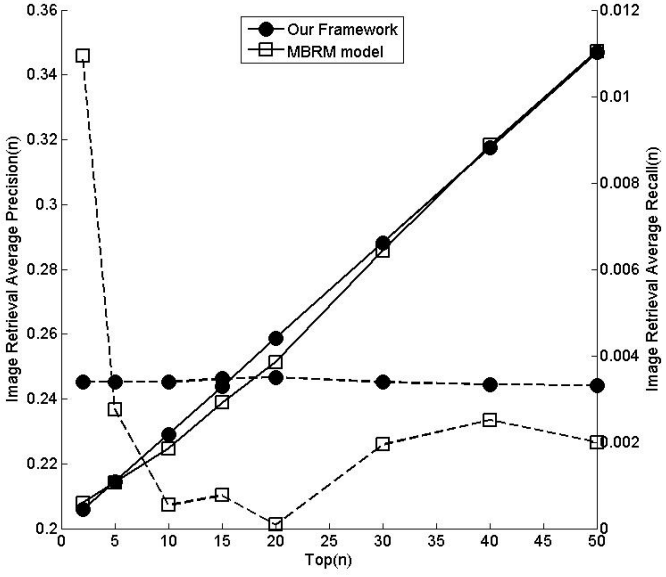


Fig. 3. Evaluation of image retrieval between our framework and MBRM model.

In the figures, the dashed lines are for precisions and the solid lines are for recalls. In the image annotation result shown in Fig. 1, the performance becomes better when the new images are added to the image database. This is consistent with the analysis in Sec. 4.6. When the image database scales up to the size as the same as that used by the MBRM model, our framework works slightly better than MBRM. In the word query result shown in Fig. 2, our framework performs significantly better than MBRM. Similarly in the image retrieval performance shown in Fig. 3, our framework works much better than MBRM.

6. Conclusion

In this chapter, we discuss a multimodal framework on image annotation and retrieval based on the max margin approach. The whole problem is mapped to a quadratic programming problem. Our framework is highly scalable in the sense that it takes a constant time to accommodate the database updating without needing to retrain the database from the scratch. The evaluation result shows significant improvements on the performance over a state-of-the-art method.

Acknowledgment

This work is supported in part by the NSF (IIS-0535162, IIS-0812114).

7. References

[1] Altun, Y., Tsochantaridis, I., & Hofmann, T. (2003). Hidden markov support vector machines. *Proc. ICML*. Washington DC.

[2] Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D. M., & Jordan, M. I. (2003). Matching words and pictures. *Journal of Maching Learning Research*, *3*, 1107–1135.

[3] Blei, D., & Jordan, M. (2003). Modeling annotated data. *Proceedings of the 26th annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 127–134).

[4] Brefeld, U., & Scheffer, T. (2006). Semi-supervised learning for structured output variables. *Proc. ICML*. Pittsburgh, PA.

[5] Chang, E., Goh, K., Sychay, G., & Wu, G. (2003). Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Trans. on Circuits and Systems for Video Technology*, *13*, 26–38.

[6] Chu, W., Ghahramani, Z., & Wild, D. L. (2004). A graphical model for protein secondary structure prediction. *Proc. ICML*. Banff, Canada.

[7] Datta, R., Ge, W., Li, J., & Wang, J. Z. (2006). Toward bridging the annotation-retrieval gap in image search by a generative modeling approach. *Proc. ACM Multimedia*. Santa Barbara, CA.

[8] Duygulu, P., Barnard, K., de Freitas, N., & Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *Seventh European Conference on Computer Vision* (pp. 97–112).

[9] Feng, S. L., Manmatha, R., & Lavrenko, V. (2004). Multiple bernoulli relevance models for image and video annotation. *International Conference on Computer Vision and Pattern Recognition*. Washington DC.

[10] Freund, Y., & Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Maching Learning*.

[11] III, H. D., & Marcu, D. (2005). Learning as search optimization: Approximate large margin methods for structured prediction. *Proc. ICML*. Bonn, Germany.

[12] Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. ICML*.

[13] McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy markov models for information extraction and segmentation. *Proc. ICML*.

[14] Pan, J.-Y., Yang, H.-J., Faloutsos, C., & Duygulu, P. (2004). Automatic multimedia cross-modal correlation discovery. *Proceedings of the 10th ACM SIGKDD Conference*. Seattle, WA.

[15] Schölkopf, B., & Smola, A. (2002). *Learning with kernels support vector machines, regularization, optimization and beyond*. MIT Press, Cambridge, MA.

[16] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *22*, 1349–1380.

[17] Taskar, B., Chatalbashev, V., Koller, D., & Guestrin, C. (2005). Learning structured prediction models: A large margin approach. *Proc. ICML*. Bonn, Germany.

[18] Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin markov networks. *Neural Information Processing Systems Conference*. Vancouver, Canada.

[19] Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. *Proc. ICML*. Banff, Canada.

[20] Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer.

[21] Vapnik, V. N. (1998). *Statistical learning theory*. John Wiley & Sons, Inc.

[22] Wu, Y., Chang, E. Y., & Tseng, B. L. (2005). Multimodal metadata fusion using causal strength. *Proc. ACM Multimedia* (pp. 872–881). Hilton, Singapore.

7

Invariant Image Retrieval: An Approach Based On Multiple Representations and Multiple Queries

Noureddine Abbadeni
College of Computer and Information Sciences, KSU
Riyadh, KSA
nabbadeni@ksu.edu.sa; noureddine.abbadeni@usherbrooke.ca

Abstract
We present an approach based on multiple representations and multiple queries to tackle the problem of invariance in the framework of content-based image retrieval. We consider the case of textures. This approach, rather than to consider invariance at the representation level, considers it at the query level. We use two models to represent the visual content of textures, namely the autoregressive model and a perceptual model based on a set of perceptual features. The perceptual model is used with two viewpoints: the original image's viewpoint and the autocovariance function viewpoint. After a brief presentation and discussion of these multiple representation models/viewpoints, we present the invariant texture retrieval algorithm. This algorithm considers results fusion (merging) at two different levels: 1. The first level consists in merging results returned by different models/viewpoints (representations) for the same query in one results list using a linear results fusion model; 2. The second level consists in merging each fused list of different queries into a unique fused list using a round robin fusion scheme. Experimental retrieval results and benchmarking based on the precision/recall measures applied on a large image database show interesting results.

1. Introduction

Content-based image retrieval (CBIR) is a research area that has been active for more than a decade and many research results and prototypes have been carried out since then (4), (1), (23), (26), (14), (27), (24), (11), (28). Despite the important number of works related to CBIR, the problem of invariance has not received sufficient attention yet. Invariance, a difficult problem in computer vision and image analysis, deals with, in the framework of image retrieval, the ability to retrieve all relevant images to a query image even if some of them have been transformed according to different geometric and photometric transformations such as rotation, scaling, illumination, viewpoint change and contrast change as well as non-rigid transformations (30). Some works in the literature have indicated that, even when a texture is transformed with some geometric or photometric transformation, human subjects still perceive it as the same texture (30), (21), (22). Two images with similar textures must still be similar, for some applications, if we rotate one texture or if we change the contrast of another texture for example. Therefore invariance might be an important problem depending on applications and users' needs.

This chapter is organized as follows: In section 2, related works are briefly presented and discussed; In section 3, the different content representation models/viewpoints are briefly presented and discussed; In section 4, the multiple queries approach is presented and discussed; In section 5, results fusion strategies are presented and discussed; In section 6, the invariant texture retrieval algorithm is applied to a large texture database and benchmarked using the well-known precision and recall measures; And, finally, in section 7, a conclusion and further investigations related to this work are briefly discussed.

2. Related work

Several works published in the literature deal with the problem of invariance, especially in the case of textures. These research works can be classified in two main categories (21), (30):
1. The first class of works considers invariance at the representation level, i.e. they try to develop a model which is invariant with respect to a limited number of transformations. A well-known example of such an approach is the rotation-invariant simultaneous autoregressive model ($RISAR$) proposed by Mao and Jain (25). Another example is the Wold-like model (23) which may present a good level of invariance with respect to translation, rotation and scaling. Other statistical, structural and model-based texture analysis methods which have some degree of invariance are discussed in (30). The main problem with this approach is its limited scope since it is very difficult to develop an invariant model that takes into account a large number of transformations. In general, models of this class consider one or two well-defined transformations and ignore the long list of all possible transformations. A more complete review of research works in this category can be found in (30).
2. The second class of invariant models consider invariance at the query level, rather than the representation level. Among the works that considered this approach, we cite (17) and (19). However, these works considered the problem of image retrieval in general and did not consider the problem of invariance in particular. We did not find in the literature any work that deals with image retrieval and explicitly considers invariance at the query level. In this approach, the problem is the number of queries and the number of representations needed for each query. Taken in an absolute way, there is no satisfying solution to this problem. However, in the framework of CBIR, we believe that a good solution to this problem must be based on users' needs. In fact, users' needs may play an important role in choosing the set of prototypes to represent an image. For example if a user is interested in finding the same texture with different orientations, we may use a restricted number of query images that can represent different orientations. If the user is interested in scaling, we may use a restricted number of query images that can represent different scales, etc.
The new invariant texture retrieval algorithm presented in this chapter fits within the second approach. We use multiple models/viewpoints to represent textural content of images. At the query level, we use multiple queries to represent user's need. We use appropriate results fusion models to merge results returned by each model/viewpoint and by each query. Results fusion is performed at two different levels (3): 1. The first level consists in merging results returned by different representations for the same query; 2. The second level consists in merging results returned by each query for the same user's need.

3. Multiple Representations

To represent texture content, we use two different models, the autoregressive model and a perceptual model based on a set of perceptual features (8). The autoregressive model used

is a causal simultaneous AR model with a non-symmetric half-plan (NSHP) neighborhood. The perceptual model is considered with two viewpoints: the original image's viewpoint and the autocovariance function (associated with original images) viewpoint. Each of the viewpoints of the perceptual model used is based on four perceptual features, namely coarseness, directionality, contrast and busyness. So, we have a total of three models/viewpoints; each model/viewpoint results in a vector of parameters of size four for a total of twelve parameters. Briefly, the autoregressive model is characterized, in particular, by a forecasting property that allows to predict the grey-level value of a pixel of interest in an image by using the grey-level values of pixels in its neighborhood. The autoregressive model, when used to model a textured image, allows to estimate a set of parameters (their number corresponds to the number of neighbors considered), each one corresponds to the contribution of its corresponding pixel in the forecasting of the pixel of interest (the total of contributions of all pixels in an image is 100%).

The perceptual model, which is perceptual by construction, is based on a set of four computational measures that simulate four perceptual features: coarseness, directionality, contrast, and busyness. Briefly, coarseness was estimated as an average of the number of extrema; Contrast was estimated as a combination of the average amplitude of the gradient, the percentage of pixels having their amplitude superior to a certain threshold and coarseness itself; Directionality was estimated as the average number of pixels having the dominant orientation(s); And finally, busyness was estimated based on coarseness. The computational measures proposed for each perceptual textural feature were evaluated by conducting a set of experimentations taking into account human judgments and using a psychometric method. Thirty human subjects were asked to rank a set of textures according to each perceptual feature. Then, for each perceptual feature, we consolidated the different human rankings into one human ranking using the sum of rank values. For each feature, the consolidated human ranking obtained was compared to the ranking given by the corresponding computational measure using the Spearman coefficient of rank-correlation. Experimental results showed very strong correspondence between the proposed computational measures and human rankings. Values of Spearman coefficient of rank-correlation r_s found were as follows: for coarseness, $r_s = 0.913$; for directionality, $r_s = 0.841$; for contrast, $r_s = 0.755$; and finally, for busyness, $r_s = 0.774$. Compared to related works, our results were found more satisfactory (8), (9).

These models and viewpoints were not constructed to be invariant to geometric and photometric transformations. The autoregressive model and the perceptual model based on the autocovariance function viewpoint have poor performance in the case of invariance due to the fact that both the autocovariance function and the autoregressive model are very sensitive to variance and transformations that may occur in a texture. The perceptual model based on the original image's viewpoint performs well in the case of invariance even if it was not constructed to deal with the problem of invariance. In fact, the measure of directionality proposed is not very sensitive to rotation since we compute a degree of directionality and not a specific orientation. Directionality is, however, sensitive to scaling since it is computed as a number of oriented pixels: the more a texture is coarse, the less is the number of oriented pixels and the less the texture is directional. The measure of coarseness proposed is robust with respect to rotation but it is sensitive to scaling. The same remarks given on coarseness are applicable to busyness. The measure of contrast proposed is, obviously, sensitive to illumination conditions. For more details on the perceptual model, refer to (8), (9) and (6); for more details on the autoregressive model, refer to (10) and (5).

4. Multiple Queries

In the framework of content-based image retrieval, we propose the use of multiple queries and representations to tackle the problem of texture invariance. That is, rather than trying to develop invariant content representation models, which may be very complex and of limited scope, we consider the problem of invariance at the query and the representation levels. Each user's need is represented by a set of query images that are different from each other in terms of orientation, scale, contrast, etc. depending on the user's needs. Search results returned for each query are fused to form a unique results list using appropriate results fusion models. For example, if a user wants to retrieve all textures that are visually similar to a given texture, even those not having the same orientation(s) as the given texture, we can represent this user's need by a set of query images with different orientation(s) and fuse the results returned for each query image to form a unique results list that corresponds to the original user's need.
Results fusion returned for different queries allows to consider the fact that relevant images to a given query are not necessarily near the query in the feature space, especially in the case of invariance. In fact, since content representation models are not, in general, invariant with respect to different transformations, features computed on the same image, which has been rotated for example, are not the same as the original image and can be very different. So, such features are not in the neighborhood of the query image in the feature space and can be even located in disjoint regions that can be more or less far from each other in the feature space (17). The use of only one query image will not retrieve an important set of relevant images (depending on the degree of variance in the database) since, in this case, these relevant images are not necessarily near the considered query image in the feature space. Figure 1 shows the fact that many query images will search in different regions, that can be disjoint, in the feature space (19).

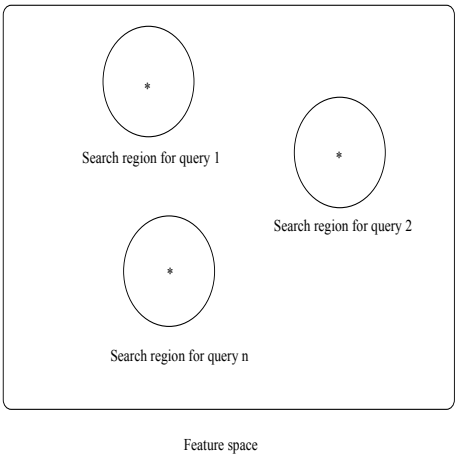


Fig. 1. Different regions for different queries in the same feature space. When query images are different, the corresponding results may be located in different regions in the feature space.

We should also mention that with multiple queries, there is some computational overhead since we use several queries for the same user's need. However, with such an approach, the size of the parameters vector does not change for images in the database. Only a user's need is represented with several queries, and thus with several parameters vectors. So, with such an approach, efficiency is not altered in an important way. Compared to the other approach of tackling invariance, i.e. representation-level invariance, and making abstraction of its difficulty and its limited scope to handle a wide range of transformations, the number of parameters is generally increased in an important way and this results generally in an important loss of efficiency.

5. Results fusion

Results fusion has been used in the information retrieval community for a long time now (12; 13; 20; 29). Results fusion concerns two levels: 1. results fusion from multiple models/viewpoints (representations) for the same query; 2. And, results fusion from different queries.

5.1 Multiple models/viewpoints fusion

In the invariant texture retrieval algorithm we are proposing, each query is represented by three different set of parameters corresponding to the different models/viewpoints. Results fusion returned by each of these models/viewpoints, for the same query, can be done using an appropriate fusion model. We have experimented many results fusion models. The model which gave the best results was the linear fusion model, denoted $FusCL$, and defined as follows:

$$FusCL_{ij} = \frac{\sum_{k=1}^{K} GS_{M_{ij}^k}}{K} \tag{1}$$

Equation (1) is based on the similarity value returned by the similarity measure $GS_{M_{ij}^k}$ (scores) and expresses the merging of results returned by different models/viewpoints M^k as an average of the scores obtained by an image in its different rankings corresponding to different models/viewpoints. K is the number of models/viewpoints considered, i is the query image and j corresponds to images returned for query i according to model/viewpoint M^k.
The $FusCL$ model exploits two main effects: 1. The first effect known as the chorus effect in the information retrieval community (29), that is, when an image is returned as relevant to a query by several models/viewpoints, this provides stronger evidence of relevance than if it is returned by only one model/viewpoint; 2. The second effect called the dark horse effect (29), that is, when a model/viewpoint ranks exceptionally an image, which is not relevant to a query, in top positions, this can be attenuated by the fused model if the other models/viewpoints do not rank it in top positions (it is very rare for a non-relevant image to be ranked at top positions by several models/viewpoints).

5.2 Multiple queries fusion

In the case of multiple queries, we can use the same fusion models as in the case of multiple models/viewpoints fusion. However, those models did not give good results in practice since the chorus and dark horse effects are not very significant in the case of invariance. In fact, when using multiple queries, each query with a different orientation, scale, or contrast for example, the results returned for each query contain, actually, a little number of common

images. Models that can be used, in this case, are models able to exploit the skimming effect, i.e. take from each list (for each query), the best results. Among such models, we have the $FusMAX$ model and the $FusRR$ model.

FusMAX. We define the MAX model, denoted $FusMAX$, as follows:

$$FusMAX_{ij} = MAX(GS_{M_{ij}^k}) \tag{2}$$

The MAX model will consider the model/viewpoint that gives the highest value of similarity. $GS_{M_{ij}^k}$ is the similarity measure defined in section 6.2. The MAX model may give good results since it exploits the so-called skimming effect (29) which corresponds better to the case of invariance.

FusRR. Another fusion technique that can be used, since it exploits also the skimming effect, is a technique called round robin (we denote it RR or $FusRR$) (12), (13). The RR technique makes use of the rank of an image in the returned results list rather than the value of the similarity function. The RR technique simply considers images that are ranked in the first position in each list, corresponding to each query, and gives them all the same first position in the fused list, then taking images that are ranked in the second position in each list, corresponding to each query, and give them all the same second position in the fused list and so on. We can stop this process after a threshold of the similarity value and/or the rank or after having taken a sufficient number of images. Obviously, if we find the same image in different lists, which may occur occasionally, only one image is considered. Note that the RR technique is different from the MAX model. Experimental results will show that the RR fusion technique exploits the skimming effect in a more effective way than the MAX model does.
The RR technique, as mentioned, does not make use of similarity values. If necessary, we can use the rank given to images in the fused list and compute an artificial similarity value for each image in the fused list. To do so, we can use the rank-based similarity measure defined as follows (20):

$$Sim_{Rank} = 1 - \frac{Rank - 1}{N} \tag{3}$$

where $Rank$ is the rank given to an image in the fused list and N is the number of images in the fused list. With such an artificial similarity function, an image ranked at position 1 will have $Sim_{Rank} = 1$, an image ranked at the last position will have Sim_{Rank} near 0 (actually $1/N$) and all other images will have an artificial similarity value between 0 and 1.

6. Application to image retrieval

6.1 Invariant texture retrieval algorithm

A general scheme of our invariant texture retrieval algorithm is given in figure 2. The algorithm is based on two levels of results fusion as we have already mentioned. The first level consists in fusing results returned by different models/viewpoints for the same query. The second level consists in fusing results returned for different queries. Multiple models/viewpoints fusion, for the same query, is based on fusion models exploiting the chorus and dark horse effects such as the $FusCL$ model while multiple queries fusion is based on fusion models exploiting the skimming effect such as $FusMAX$ and $FusRR$ models.

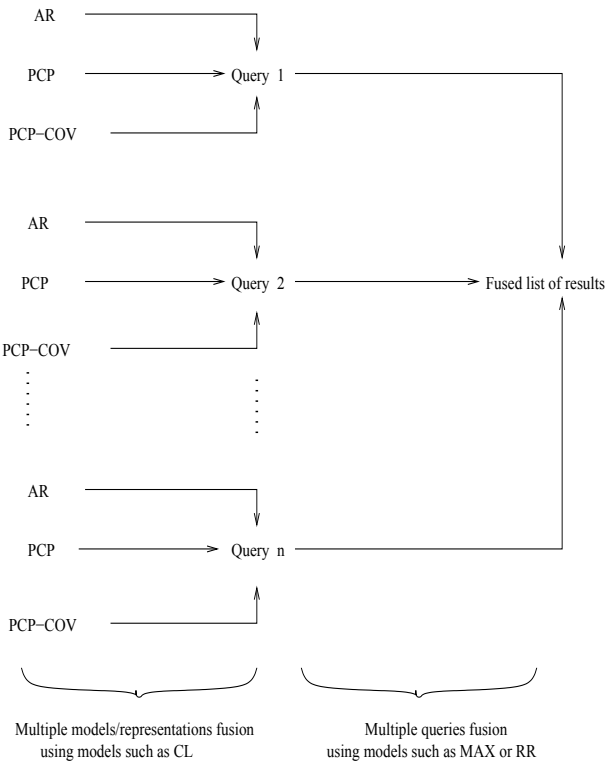


Fig. 2. General scheme of our invariant texture retrieval algorithm. The algorithm has 2 levels of fusion. Each query is represented by 3 different models/viewpoints: AR, PCP, and $PCP - COV$. The first level of fusion consists in merging the retrieval results returned by each model/viewpoint, for the same query, using the $FusCL$ model. The second level of fusion consists in merging the retrieval results for each query in one final fused list of results using either the $FusMAX$ model or the $FusRR$ model.

6.2 Similarity measure

The similarity measure used is based on Gower coefficient of similarity (18) we have developed in our earlier work (7). The non-weighted similarity measure, denoted GS, can be defined as follows:

$$GS_{ij} = \frac{\sum_{k=1}^{n} S_{ij}^{(k)}}{\sum_{k=1}^{n} \delta_{ij}^{(k)}} \tag{4}$$

where $S_{ij}^{(k)}$, the partial similarity, is the score of comparison of images i and j according to feature k and $\delta_{ij}^{(k)}$ represents the ability to compare two images i and j on feature k. $\delta_{ij}^{(k)} = 1$ if

images i and j can be compared on feature k and $\delta_{ij}^{(k)} = 0$ if not[1]. $\sum_{k=1}^{n} \delta_{ij}^{(k)} = n$ if images i and j can be compared on all features $k, k = 1..n$. Note that, in our case, we will use all features; thus, $\delta_{ij}^{(k)} = 1$ for all k, i, j.

$S_{ij}^{(k)}$ is defined as follows:

$$S_{ij}^{(k)} = 1 - \frac{|x_{ik} - x_{jk}|}{R_k} \tag{5}$$

where x_{ik} represents the value of feature k of image i. R_k represents a normalization factor. R_k is computed on the database considered for experimentations and is defined as follows :

$$R_k = Max(x_{ik}) - Min(x_{ik}) \tag{6}$$

The weighed version of the similarity measure can be defined as follows :

$$GS_{ij} = \frac{\sum_{k=1}^{n} w_k S_{ij}^{(k)}}{\sum_{k=1}^{n} w_k \delta_{ij}^{(k)}} \tag{7}$$

where w_k corresponds to the weight associated with feature k.

We have used, in the case of the AR model, the inverse of variance of each estimated parameter k computed on the considered experimental database as a weight for this parameter. In the case of the perceptual model, we have used two approaches of weighting: 1. In the first approach, as in the case of the AR model, we used the inverse of variance of each feature as a weight for this feature; 2. In the second approach, we used the Spearman coefficients of rank-correlation found for each perceptual feature (section 2) as a weight for this feature. The use of the inverse of the variance as a weight is motivated by the fact that the discrimination capacity of that feature increases when the variance is small and decreases when the variance is large. Therefore, using the inverse of variance as a weight will give more importance to features having a more important discrimination capacity (16). The use of Spearman rank-correlation coefficient as a weight is motivated by the fact that we want to give more importance to perceptual features that are highly correlated with human judgments. Note that the Spearman coefficient of rank-correlation is independent from the image database considered for benchmarking while the inverse of variance is dependent on this image database.

6.3 Benchmarking database

For experimental results and benchmarking, we have used an image database coming from Ponce's group at UIUC. [2] This database contains 22 classes of 40 640x480 images each class for a total of 880 images. [3] Images within the same classes have been taken with different photometric and geometric conditions. In each class there is a high degree of variance between

[1] Note that, in our case, we are able to compare all images on all features. Thus, $\delta_{ij}^{(k)}$ will never be 0. The formula is a general expression of the similarity measure and can be used in other cases where images or objects cannot be always compared on all features.

[2] http://www-cvr.ai.uiuc.edu/ponce_grp/data/texture_database

[3] Note that, originally, this database contains 25 classes. However, during the conversion of images from their original JPEG format to PGM format, we have found some errors in some images in classes 17, 18 and 23. We have decided to eliminate these 3 classes and consider only 22 classes rather than 25 classes.

images in orientation, scale, contrast as well as non-rigid deformations. A sample from this database is given in figure 3. [4]

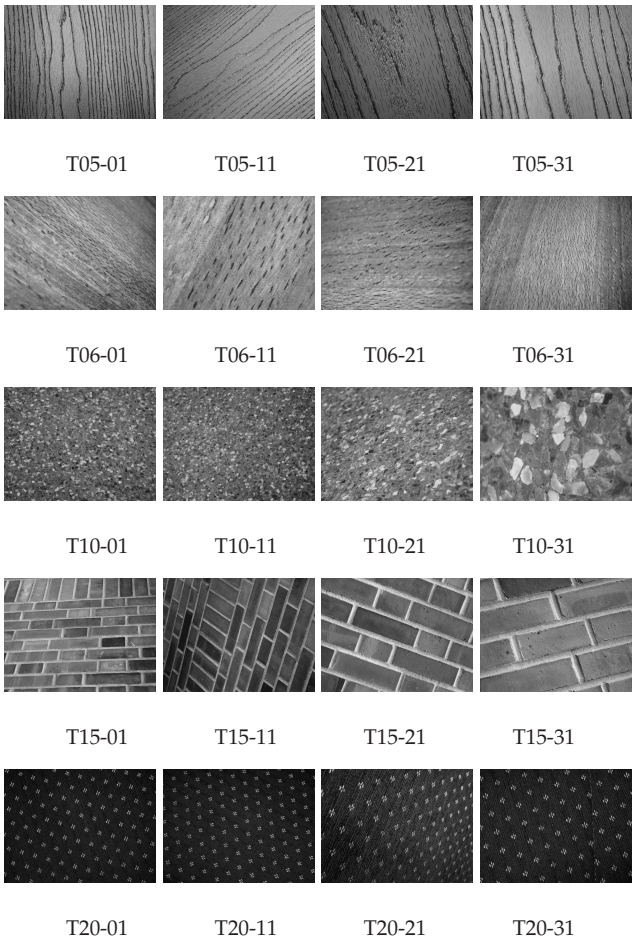


Fig. 3. Sample of images from the Ponce's group texture database: 4 sample images from 5 different classes T05, T06, T10, T15, and T20.

In experimental results, the AR model with an $NSHP$ (non-symmetric half-plan) neighborhood weighted with the inverse of variance of each feature gave the best results among the different variants of the AR models. Both the PCP model (perceptual model based on original images) and the $PCP - COV$ model (perceptual model based on the autocovariance function) in their weighted version using Spearman coefficient of rank-correlation gave the best results among the different variants of the perceptual model. In the rest of this chapter, we consider

[4] http://www-cvr.ai.uiuc.edu/ponce_grp/data/texture_database/samples

these three best versions for results fusion:

- **AR** or **AR NSHP-V**: The autoregressive model weighted with the inverse of variance.
- **PCP** or **PCP-S**: The perceptual model based on the original image's viewpoint and weighted with Spearman coefficients of rank-correlation.
- **PCP-COV** or **PCP-COV-S**: the perceptual model based on the autocovariance function viewpoint and weighted with Spearman coefficients of rank-correlation.

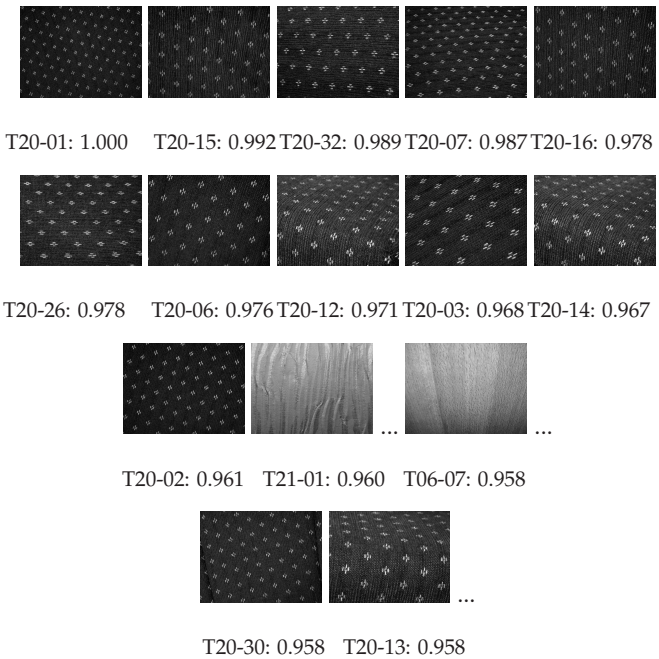


Fig. 4. Search results returned for query T20-01 using the *PCP* model weighted with Spearman coefficients. T20-01 is the query image. Results are given in a decreasing order according to their similarity value.

6.4 Experimental results

When considering different models/viewpoints separately, the *PCP* model (based on the original image's viewpoint) is the best model since, as we mentioned earlier, it exhibits some robustness in the case of invariance comparatively to the other models/viewpoints. In the following figures, we show some examples of results obtained with the *PCP* model weighted with Spearman coefficient of rank-correlation (see Figures 4, 5 and 6).

When examining these results, we can point out that the *PCP* model, with only 4 parameters, gave good results even if it was not constructed originally to be invariant with respect to some transformations. However, using one model/query gives in some cases bad results, that is some images not similar to the query image are returned with a high similarity value. The other two models, the autoregressive model and the $PCP - COV$ (perceptual model based on

the autocovariance function viewpoint), have not good performance (more details are given in the following figures and tables).

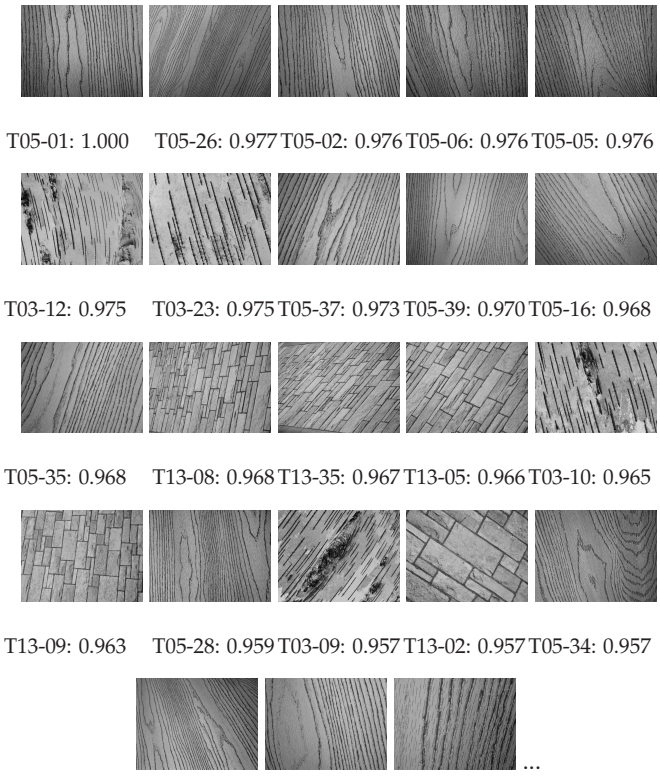


Fig. 5. Search results returned for query T05-01 using the *PCP* model weighted with Spearman coefficients. T05-01 is the query image. Results are given in a decreasing order according to their similarity value.

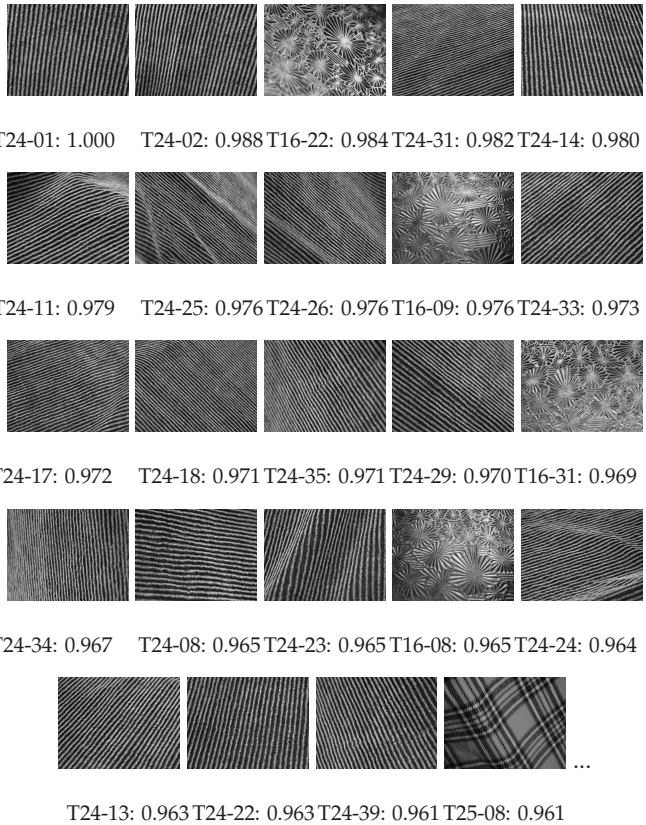


Fig. 6. Search results returned for query T24-01 using the *PCP* model weighted with Spearman coefficients. T24-01 is the query image. Results are given in a decreasing order according to their similarity value.

6.5 Precision and recall measures

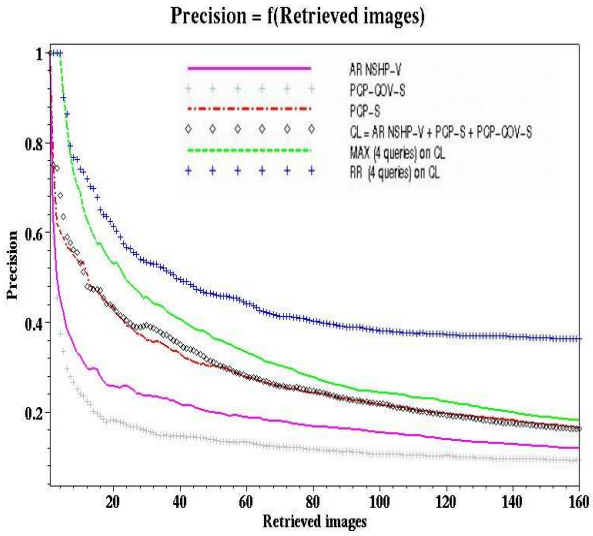
Precision and recall measures are common standard techniques to benchmark search relevance in information retrieval systems in general Dun. 97. Precision, which can be defined as the number of relevant and retrieved images divided by the number of retrieved images, measures the ability of the model to reject non-relevant images with respect to a query. Recall, which can be defined as the number of relevant and retrieved images divided by the number of relevant images in the database for the considered query, measures the ability of a model to retrieve all relevant images. Precision and recall measures are computed for each query at each position. Then, average precision and average recall are computed respectively as an average across a set of representative queries. Two complementary graphs are then given: 1. Average precision as a function of the number of retrieved images (positions) allowing to see the ability of the model to reject non relevant images as the model retrieves more images; 2. Average recall as a function of the number of retrieved images (positions) allowing to see the

ability of the model to retrieve all relevant images as the model retrieves more images.

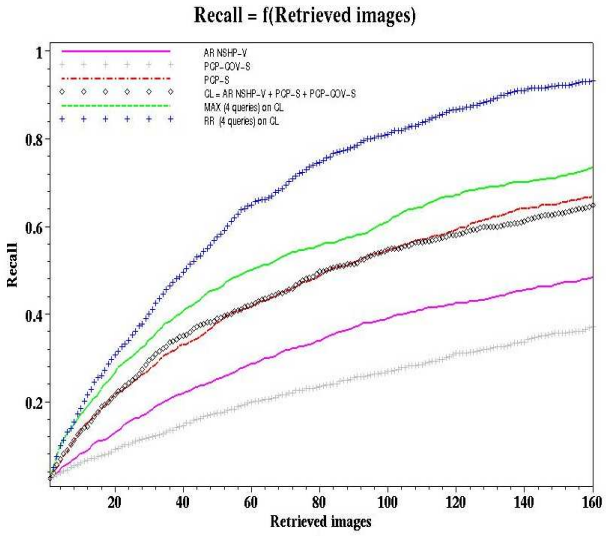
Benchmarking we have done, based on the precision-recall measures, concerns both the use of one query as well as the use of multiple queries (see figure 7):

- When considering one query, we have considered the best version of each separated model/viewpoint. Then we have fused these best multiple models/viewpoints, using the CL results fusion model, for the same query.
- When considering multiple queries, based on the fused models/viewpoints for each query, we have fused the results of 4 queries using the MAX model and the RR model as described earlier in this chapter. Query images were selected randomly and correspond to images 1, 11, 21 and 31 from each class.

Figure 7 shows precision and recall graphs for different models, both separated and fused, by considering one query and multiple queries. When examining this figure, we can see that the PCP model weighted with Spearman coefficients performs better than the other separated models. Fusing different models/viewpoints for the same query did not achieve an important improvement in performance since two of the three models/viewpoints considered, namely the $ARNSHP-V$ and the $PCP-COV$ model, have rather poor performance in the case of invariance as explained earlier in this chapter. Fusing multiple queries using both the MAX and RR models allow improvement in performance. While the MAX model allows an average improvement in performance, the RR model allows a significant improvement in performance measured in terms of precision and recall. Remember that both the MAX and the RR models exploit the skimming effect while the CL model exploits both the chorus and dark horse effects. Thus, in the case of invariance, the skimming effect, in a multiple queries approach, is more important than both the chorus and dark horse effect. These conclusions can be also drawn by examining tables 1, 2, 3, 4, 5 and 6 in the next subsection.



(a)



(b)

Fig. 7. Precision and recall graphs for different separated models (1 query) and fused model (4 queries): (a) Precision as a function of the number of retrieved images and (b) Recall as a function of the number of retrieved images. The fused *RR* model is clearly the best model.

6.6 Retrieval rate per class and average retrieval rate

Tables 1, 2, 3, 4 and 5 give the retrieval rate (recall) per class at positions 40, 80, 120 and 160 according to the PCP, the $PCP - COV$, the $ARNSHP - V$, the MAX and the RR models respectively. Table 6 gives the average retrieval rate at positions 40, 80, 120 and 160 across all the database according to a selection of separated and fused models. Examination of these different tables allows to draw the same conclusions as in the precedent subsection in which we showed precision and recall graphs for the different separated and fused models. We can particularly notice, again, the good performance of the multiple queries approach using the RR fusion model.

Class	P40	P80	P120	P160
T01	.325	.45	.575	.65
T02	.225	.4	.5	.625
T03	.1	.225	.3	.425
T04	.4	.525	.6	.65
T05	.525	.675	.7	.725
T06	.375	.6	.725	.8
T07	.475	.625	.75	.825
T08	.525	.675	.775	.9
T09	.225	.45	.575	.6
T10	.4	.65	.825	.875
T11	.2	.425	.55	.625
T12	.2	.375	.475	.55
T13	.475	.7	.825	.925
T14	.275	.375	.45	.55
T15	.15	.275	.425	.55
T16	.425	.55	.65	.725
T19	.075	.175	.375	.5
T20	.65	.875	.975	1
T21	.35	.45	.5	.55
T22	.175	.275	.4	.475
T24	.625	.85	.925	.925
T25	.1	.15	.175	.3

Table 1. Retrieval rate for each class at positions 40, 80, 120, and 160 according to the $PCP - S$ model (1 query). With the $PCP - S$ model, results are acceptable for some classes and not acceptable for others. For example, for class T24, at position 40, we have retrieval rate of 0.625 while for class T03, the retrieval rate is 0.1 at position 40. For positions 80, 120 and 160, results are improved in an important way for some classes.

Class	P40	P80	P120	P160
T01	.075	.2	.225	.3
T02	.15	.2	.25	.3
T03	.1	.225	.3	.35
T04	.05	.1	.125	.15
T05	.1	.175	.2	.3
T06	.15	.25	.325	.4
T07	.325	.475	.65	.725
T08	.35	.55	.7	.725
T09	.05	.1	.125	.15
T10	.125	.2	.3	.4
T11	.075	.1	.2	.275
T12	.15	.25	.45	.6
T13	.15	.25	.35	.4
T14	.075	.075	.125	.125
T15	.15	.2	.275	.35
T16	.1	.15	.15	.175
T19	.025	.1	.175	.2
T20	.5	.75	.8	.875
T21	.25	.35	.5	.5
T22	.05	.175	.225	.275
T24	.05	.1	.175	.3
T25	.15	.175	.2	.275

Table 2. Retrieval rate for each class at positions 40, 80, 120, and 160 according to the $PCP - COV - S$ model (1 query). With the $PCP - COV - S$ model, results are not acceptable for most of classes. For example, for class T20, at position 40, we have retrieval rate of 0.5 while for class T04, the retrieval rate is 0.05 at position 40. For positions 80, 120 and 160, results are improved for some classes only. Note that the results of the $PCP - COV - S$ model are clearly less good than the results of the $PCP - S$ model.

Class	P40	P80	P120	P160
T01	.175	.425	.5	.525
T02	.375	.425	.475	.475
T03	.05	.1	.1	.15
T04	.4	.475	.625	.625
T05	.225	.35	.5	.6
T06	.175	.2	.25	.25
T07	.05	.1	.15	.175
T08	.2	.375	.4	.45
T09	.275	.525	.775	.875
T10	.325	.45	.55	.7
T11	.325	.5	.675	.8
T12	.275	.3	.325	.4
T13	.3	.375	.5	.6
T14	.125	.225	.3	.375
T15	.1	.275	.4	.45
T16	.4	.65	.75	.775
T19	.125	.25	.325	.375
T20	.275	.475	.6	.7
T21	.025	.025	.025	.025
T22	.125	.225	.25	.35
T24	.35	.55	.625	.675
T25	.15	.2	.275	.325

Table 3. Retrieval rate for each class at positions 40, 80, 120, and 160 according to the $ARNSHP - V$ model (1 query). With the AR-NSHP-V model, results are not acceptable for most of classes. For example, for class T04, at position 40, we have retrieval rate of 0.4 while for class T03, the retrieval rate is 0.05 at position 40. For positions 80, 120 and 160, results are improved for some classes only. Note that the results of the AR-NSHP-V model are clearly less good than the results of the $PCP - S$ model.

Class	P40	P80	P120	P160
T01	.375	.575	.675	.75
T02	.375	.55	.65	.725
T03	.35	.475	.6	.6
T04	.25	.325	.425	.525
T05	.425	.6	.675	.725
T06	.325	.425	.55	.675
T07	.275	.375	.6	.7
T08	.4	.6	.65	.7
T09	.5	.6	.775	.85
T10	.55	.7	.8	.85
T11	.675	.825	.95	.95
T12	.575	.8	.95	.95
T13	.5	.675	.825	.875
T14	.35	.525	.725	.825
T15	.4	.55	.625	.725
T16	.55	.725	.85	.925
T19	.35	.5	.6	.7
T20	.725	.85	.95	.95
T21	.2	.325	.45	.525
T22	.2	.475	.55	.575
T24	.35	.4	.475	.575
T25	.275	.375	.475	.5

Table 4. Retrieval rate for each class at positions 40, 80, 120, and 160 according to the MAX model (4 queries) applied on the fused CL model (1 query). With the fused model MAX model, results are clearly much better than the cases where we used one model/query. For most classes, the retrieval rate, at positions 80, 120 and 160, is improved in an important way.

Class	P40	P80	P120	P160
T01	.425	.675	.95	1
T02	.4	.7	.9	1
T03	.425	.675	.9	1
T04	.275	.525	.7	.9
T05	.55	.9	1	1
T06	.275	.5	.65	.8
T07	.425	.725	.9	1
T08	.55	.8	1	1
T09	.725	1	1	1
T10	.625	.9	1	1
T11	.675	1	1	1
T12	.675	.95	1	1
T13	.625	1	1	1
T14	.35	.6	.8	1
T15	.35	.625	.75	.825
T16	.65	.95	1	1
T19	.4	.55	.65	.775
T20	.875	1	1	1
T21	.375	.475	.525	.625
T22	.375	.5	.6	.7
T24	.45	.775	1	1
T25	.425	.6	.75	.875

Table 5. Retrieval rate for each class at positions 40, 80, 120, and 160 according to the RR model (4 queries) applied on the fused CL model(1 query). With the fused model RR model, results are clearly much better than the cases where we used one model/query. For most classes, the retrieval rate, at positions 80, 120 and 160, is improved in an important way and is close to 1. Note that the results of the fused RR model are clearly much better than the fused MAX model.

Model	P40	P80	P120	P160
$ARNSHP - V$	.219	.34	.426	.485
$PCP - S$	.331	.489	.593	.67
$PCP - COV - S$	.145	.234	.31	.37
MAX (4 Queries)	.408	.557	.674	.735
RR (4 Queries)	.495	.747	.867	.932

Table 6. Average retrieval rate at positions 40, 80, 120, and 160 according to different separated and fused models. The fused RR model is clearly the best model.

7. Conclusion

In this chapter, we presented a multiple models/viewpoints and multiple queries approach to tackle the problem of invariant image retrieval, in the case of textures. Three content representation models/viewpoints were used: the autoregressive model and a perceptual model, based on a set of perceptual features such as coarseness and directionality, used with two viewpoints: the original image's viewpoint and the autocovariance function viewpoint. Each of these models/viewpoints results in a parameters vector of size four. These models/viewpoints were not built to be invariant with respect to geometric and photometric transformations.

Experimental results and benchmarking conducted on a large database, using precision and recall measures, show that, when fusing the results returned with each model/viewpoint for the same query, using the CL fusion model, retrieval effectiveness are not significantly improved and are quite similar to the case when only the perceptual model based on the original image's viewpoint was used. When fusing results returned by multiple queries, using both the MAX and the RR fusion models, retrieval effectiveness is significantly improved especially with the RR fusion model.
The choice of appropriate queries is an open question. In this work, we have chosen multiple queries in a random way. We think that, if this choice can be done using some procedure taking into account users' needs, search effectiveness can be further improved.

Acknowledgments

The author acknowledge the financial support of the research center at the college of computer and information sciences (CCIS) at King Saud University(KSU), Riyadh, KSA.

8. References

[1] Datta R., Joshi D., Li J., and Wang J.Z. Image Retrieval: Ideas, Influences, and Trends of the New Age. ACM Transactions on Computing Surveys, 40(2), 60 pages, 2008.
[2] Abbadeni, N. and Alhichri, H. Low-level invariant image retrieval based on results fusion. Proceedings of the IEEE ICME, Hannover-Germany, June 2008.
[3] Abbadeni N. An Approach Based on Multiple Representations and Multiple Queries for Invariant Image Retrieval. Proceedings of the International Conference on Visual Information Systems, VISUAL' 2007: 570-579, Shanghai, 2007.
[4] Lew M., Sebe N., Djeraba C., and Jain R. Content-Based Multimedia Information Retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications, 26 pages, 2006.
[5] Abbadeni N. Perceptual interpretation of the estimated parameters of the autoregressive model. Proceedings of the IEEE ICIP, Genoa, Italy, 2005.
[6] Abbadeni N. Perceptual image retrieval. Proceedings of the International conference on visual information systems (VISUAL'05), Amsterdam, Netherlands, 2005.
[7] Abbadeni N. A new similarity matching measure: application to texture-based image retrieval. In: Proceedings of the 3^{rd} International Workshop on Texture Analysis and Synthesis (held in conjunction with IEEE ICCV), Nice, France, October 2003.
[8] Abbadeni N, Ziou D, Wang S. Computational measures corresponding to perceptual textural features. In: Proceedings of the 7^{th} IEEE International Conference on Image Processing, Vancouver, BC, September 10-13 2000.
[9] Abbadeni N, Ziou D, Wang S. Autocovariance-based perceptual textural features corresponding to human visual perception. In: Proceedings of the 15^{th} IAPR/IEEE International Conference on Pattern Recognition, Barcelona, Spain, September 3-8 2000.
[10] Abbadeni N. Recherche d'images basée sur leur contenu. Représentation de la texture par le modèle autorégressif. Research report (in French), NO. 216, 50 p., University of Sherbrooke, 1998.
[11] Antani S., Kasturi R., Jain R. A Survey on the Use of Pattern Recognition Methods for Abstraction, Indexing and Retrieval of Images and Video. *Pattern Recognition*, 35(2002):945-965. 2002.

[12] Belkin NJ, Cool C, Croft WB, Callan JP. The effect of multiple query representation on information retrieval performance. In: Proceedings of the 16th International ACM SIGIR Conference, pp. 339-346, 1993.

[13] Berretti S, Del Bimbo A, Pala P. Merging results for distributed content-based image retrieval, Multimedia Tools and Applications, 24:215-232, 2004.

[14] Del Bimbo A. Visual information retrieval. Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[15] Dunlop MD. Time, relevance and interaction modeling for information retrieval. In: Proceedings of the International ACM SIGIR Conference, pp. 206-213, Philadelphia, USA, 1997.

[16] Flickner M., Sawhney H., Niblack W. et al. Query by image and video content: the QBIC system. IEEE Computer, 28(9):23-32, 1995.

[17] French JC, Chapin AC, Martin WN. An application of multiple viewpoints to content-based image retrieval, In: Proceeding of the ACM/IEEE Joint Conference on Digital Libraries, pp. 128-130, May 2003.

[18] Gower JC. A general coefficient of similarity and some of its properties. Biometrics Journal, 27:857-874, December 1971.

[19] Jin X. and French JC. Improving Image Retrieval Effectiveness via Multiple Queries. *ACM MMDB Workshop*, pp. 86-93, New Orleans, Louisiane, USA, novembre 2003.

[20] Lee JH. Analysis of multiple evidence combination, In: Proceedings of the ACM SIGIR Conference, pp. 267-276, Philadelphia, PA, USA, 1997.

[21] S. Lazebnik, C. Schmid and J. Ponce. A Sparse Texture Representation Using Local Affine Regions. *Beckman CVR Technical Report*, NO. 2004-01, University of Illinois at Urbana Champaign (UIUC), 2004.

[22] Lazebnik, S., Schmid, C. and Ponce, J. Sparse Texture Representation Using Local Affine Regions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8):1265-1278, 2005.

[23] Liu F, Picard RW. Periodicity, directionality and randomness: Wold features for image modeling and retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(7):722-733, July 1996.

[24] Lu Y, Hu C, Zhu X, Zhang H, Yang Q. A unified framework for semantics and feature based relevance feedback in image retrieval systems. In: Proceedings of the 8^{th} ACM International Conference on Multimedia, pp. 31-37, Marina Del Rey, CA, 2000.

[25] Mao, J. and Jain, AK. Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models. *Pattern Recognition*, 25(2):173-188, 1992.

[26] Rui Y, Huang TS., Chang S-F. Image retreival: current techniques, promising directions and open issues. Journal of Visual Communication and Image Representation, 10:39-62 1999.

[27] Smeulders A. W. M., Worring M., Santini S., Gupta A., Jain R. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349-1380, 2000.

[28] Smith MA. and Chen T. Image and Video Indexing and Retrieval Handbook of Image and Video Processing (Second Edition), Pages 993-1012, 2005.

[29] Vogt CC, Cottrell GW. Fusion via a linear combination of scores. Information Retrieval Journal, 1:151-173, 1999.

[30] J. Zhang and T. Tan. Brief Review of Invariant Texture Analysis Methods. *Pattern Recognition*, 35:735-747, 2002.

8

Software applications for visualization territory with Web3D-VRML and graphic libraries

Eduardo Martínez Cámara, Emilio Jiménez Macías, Julio Blanco Fernández, Félix Sanz Adán, Mercedes Pérez de la Parte and Jacinto Santamaría
University of La Rioja
Spain

1. Introduction

In the last ten years, the increasing power of computers and graphics cards has stimulated developers and users to deepen Virtual Reality (Huang & Lin, 1999; Kreuseler, 2000; Bernhardt et al., 2002). One of its natural applications of interest for the academic and the industrial communities is the Terrain Visualization Systems (TVS) (Lin et al., 1999; Luebke et al., 2003; Ellul & Haklay, 2006). In this field, two-dimensional representations have been completely superseded since three-dimensional (3D) visualization is closer to reality as well as easier to interpret. Furthermore, it allows the user to interact realistically with the environment (Hirtz et al., 1999; Kersting & Döllner, 2002).

Nowadays, there exist lots of issues to bear in mind on designing a TVS: various Web3D technologies to represent the Digital Terrain Elevation Models (DTEM), several ways to provide the TVS with sufficient realism, many graphic libraries both private and open-source projects, different applications to enable the user to interact with the system, etc. For example, some recent works can been reviewed about the use of non-proprietary available Web3D technologies (Web3D Consortium, 2006), specifically VRML, X3D and Java3D (Fairbairn & Parsley, 1997; Huang & Lin, 2002; Hay, 2003; Geroimenko & Chen, 2005; Hirtz et al., 2006). The creation of a DTEM can be referred for instance in (Ayeni, 1982; Longley et al., 2001; Fencík et al., 2005). There exist also several databases that can be used as starting point (GTOPO30, 2006; ETOPO2v2, 2006; Gittings, 1996). It is necessary to operate with these databases to extract a grid according to the necessities of realism and fast screen refresh required for this type of application. Furthermore, a correct structure and nomenclature for this grid must be carried out, in order to facilitate and to expedite its management. Another important aspect is the inclusion of texture-mapping in the model to give realism to the visualization (Heckberts, 1986; Guedes et al., 1997; Döllner et al., 2000). The applied textures are the terrain orthophotographs, which are previously treated -to readjust them according to the coordinates of the VRML environment-, partitioned -with the appropriate size for the different elements that constitute the DTEM- or properly structured in order to improve the interactive visualization of massive textured terrain datasets if needed. Regarding the VRML viewers that can be used to represent the DTEM, some well-

known samples are DeepView™ , CASUSPresenter™ , WorldView™ , BS Contact, Viewpoint Media Player, Emma 3D (Emma3D, 2006; g3DGMV, 2006; Universal 3D Format, 2006). Once 3D navigation system is developed, some interaction tools can be added using VRML Script and Java (Moore et al., 1999).

As it can be seen, there are several developments all over the world and very recent semantics in 3D visualization, so it is necessary to make an special effort in generating surveys and standards (Duke et al., 2005). In this chapter, we wish to contribute to clarify the process of development of a TVS in real time by providing a guide through the issues previously commented and illustrating the stages with practical examples. We explain the pros and cons of some of the different currently available options, offering criteria for an appropriate development. We come out of our experience in the Renewable Energy Research Group of La Rioja (Spain) to illustrate this guide. In order to overcome the limitations given by Web3D technologies in general, and VRML in particular, a specific graphic engine developed with open source graphic libraries is shown. Some programs - used to rename the terrain textures according to general VRML structures - and small applets, as interaction tools between the user and the 3D scene, have been implemented in a virtual TVS of La Rioja (one of the 17 autonomous regions in Spain, with a surface of about 5000 Km^2). They are used to clarify and exemplify some issues throughout the chapter.

The chapter is organized as follows. First, the basic characteristics of a TVS will be briefly commented in Section 2. The Web3D-VRML technologies are introduced in Section 3 where their strong and weak points are shown. Section 4 is devoted to explore the VRML viewers and some tips to create the DTEM and to endow the TVS with interactivity are provided. The development of the graphic engine, and its libraries, which present the 3D geometry of the scene, are discussed in Section 5. Finally, Section 6 concludes the chapter and refers to future work.

2. Terrain Visualization

The spatial distribution of the terrestrial surface is a continuous function, but for a digital storage and representation of these values it is necessary to reduce the infinite number of points to a finite and manageable number, so that the surface can be represented by a series of discrete values (GeoInformation, 2002) (surface discretization). For this purpose, Digital Terrain Models (DTM) and DTEM are used.

A DTM is a numeric data structure that represents the spatial distribution of a quantitative and continuous variable. These variables may be height, slope, contour and orientation, as well as any other data applicable specifically to the terrain and the characteristics of any given point. A DTEM is a numeric data structure that represents the height of the surface of the terrain. By definition it can be seen that a DTEM is a particular type of DTM (Kumler, 1994).

DTEM are usually stored in two digital formats: a) as a map of heights, that is, a two-dimensional matrix in which each quadrant represents the corresponding height of each point; b) by means of chromatic representation of the heights, that is, an image either in

shades of grey or in colours, where they depend on the specific height of each point defined. In general, dark colours are assigned to areas with low heights, and light colours are assigned to areas of high heights.

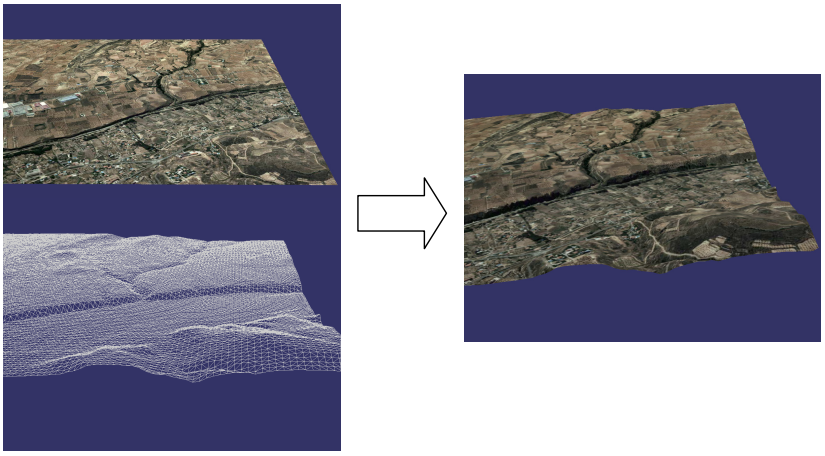


Fig. 1. Creation of a 3D terrain model from a DTEM and an orthophotograph

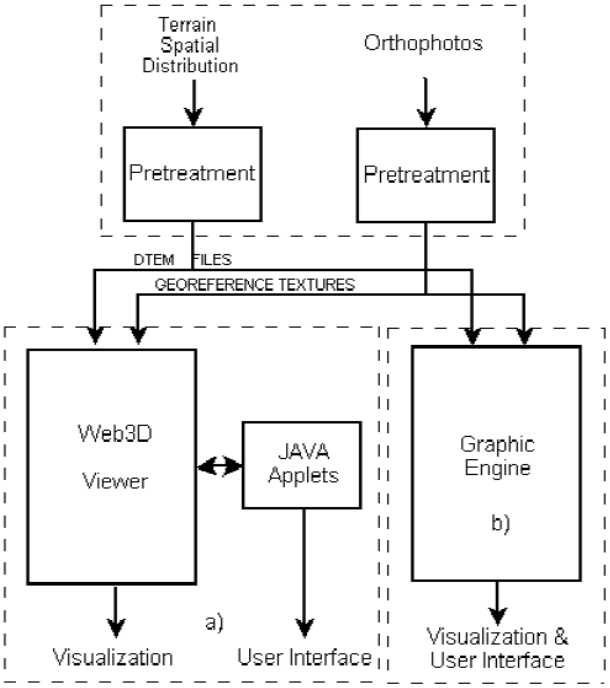


Fig. 2. Virtual Terrain Visualization Systems: a) with Web3D viewers b) with graphic engine

Obviously both formats are similar, but the main problem with this second storing method is the colour scaling assigned to the true terrain height. Starting from this point and taking any of the DTEM available on the market as a reference, it is possible to develop an own 3D DTEM. For this purpose, a polygonal surface can be created, where the vertexes agree with the coordinates taken from the appropriate DTEM (Lindstrom & Pascucci, 2001).

The next step is to achieve that this 3D model has a realistic appearance; that is to say, that it allows the user to perceive more details of any height of one given point with respect to any other one. To achieve this objective, we can think about the possibility of incorporating a specific model texture. The quickest method for achieving this realistic aspect is by using orthophotographs of the terrain.

An orthophotograph is a digitally corrected photographic presentation that represents an orthogonal projection of an object, generated from real oblique photographs. Thus, in an orthophotograph, actual horizontal measurements can be taken, the same way in which they can be taken from a map. Incorporating these corrected photographs to a DTM, an acceptable realistic representation can be obtained (see Figure 1). Figure 2 shows a diagram with the elements involved in a TVS development.

First, the orthophotos and the terrain spatial distribution data must be pre-treated, and the results are the data input to the graphic engine (case a) or to the Web3D viewer (case b). In this latter case, it can be seen that it is necessary to use Java applets, independent of the visualization, which provide the system with the capability of interaction with the user. In this chapter, Section 4 is devoted to the basic steps for a TVS implementation based on VRML, whereas Section 5 deals with a development based on a graphic engine.

3. Web3D Technologies

One of the possible ways to implement a TVS consists in using Web3D technologies. The use of any of the available Web3D technologies permits to develop a 3D environment that can be shared on Internet.

The term 'Web3D' (Web3D Consortium, 2006) refers to any programming language, protocol, archive format or technology that may be used for creating or presenting interactive 3D environments on Internet. Among these languages, used to program virtual reality, VRML (Virtual Reality Modelling Language), Java3D and X3D (Extensible 3D) are open standards.

There are also a large number of proprietary-level solutions that satisfy the specific needs of the customers, generally aimed at electronic trade and entertainment purposes, such as Cult 3D, Pulse 3D, ViewPoint, etc.

However, to use an open standard presents important advantages. For instance, the specifications and documentation are well known, and there are all kinds of applications that support these standards. Next paragraphs briefly analyse the available standards, their main characteristics, and their advantages and disadvantages.

3.1 VRML

VRML is an archive format that permits the creation of interactive 3D objects and environments. The standard VRML was created and developed by the VRML Consortium, in principle a non-profit-making organization exclusively aimed at the development and promotion of VRML as a standard 3D system on Internet. VRML appeared in 1994 as the first officially recognized technology for the creation, distribution and representation of 3D elements on Internet by the ISO (International Standards Organization). VRML was designed to cover the following basic requirements (Bell et al., 1995):

(i) To make possible the development of programs to create, publish, and maintain VRML archives, as well as programs that can import and export VRML and other 3D graphic formats.
(ii) To provide the capacity to use, combine, and re-use 3D objects in the same VRML environment.
(iii) To incorporate the capacity to create new types of objects that has not been defined as part of VRML.
(iv) To provide the possibility of being used in a wide variety of systems available on the market.
(v) To emphasize the importance of interactive functioning in a wide variety of existing applications.
(vi) To allow the creation of 3D environments at any scale or size.

VRML is a hierarchic language of marks that uses *nodes, events* and *fields* to model static or dynamic virtual realities:

- *Nodes* are used to represent particular instances of the 54 primitives of the language. Instances are defined with a collection of *fields* that contain values of the basic attributes of the primitive form.
- *Fields* are attributes that define the behaviour of the primitive forms. There are special *fields* (EventIn and EventOut) that allow the sending and reception of events to other *fields*. With these special *fields* and the command ROUTE, the flow of *events* can be controlled, directing the effect of one action among other multiple objects in order to animate a scene or simply to pass information to any of these objects.

3.2 X3D

X3D is an open standard XML (eXtensible Markup Language), a 3D archive format that permits the creation and transmission of 3D data between different applications, especially web applications.

Its principal characteristics are:

(i) X3D is integrated in XML; this represents a basic step to achieve a correct integration in:
  - Web services.
  - Distributed networks.
  - Multiplatform systems and transference of archive and data among applications.

(ii) X3D is modular; this allows the creation of a lighter 3D kernel, adjusted to the needs of developers.
(iii) X3D is extensible; this allows adding components to provide more functions, in order to satisfy the market demands.
(iv) X3D is shaped; this means that different appropriate extension groups can be chosen according to the specific needs of each application.
(v) X3D is compatible with VRML; this implies that the development, the content and the base of VRML97 is maintained.

X3D, instead of being limited to a single static wide specification - as in VRML that requires total adoption to achieve compatibility with X3D - has been designed with an structure based on components that give support for the creation of different profiles, which can be individually used. These components can be independently extended or modified, adding new levels or new components with new characteristics.

Thanks to this structure, the advances in X3D specification are faster and the development of one area does not delay the evolution of the global specification.

3.3 Java3D

Java3D™ API is a set of classes to create applications and applets with 3D elements (Sowizral et al., 1998). It offers to developers the possibility of managing 3D complex geometries. The main advantage that this application programming interface (API) presents against other 3D programming environments is that it allows the creation of 3D graphic applications, independently of the type of system. It forms part of API JavaMedia. Therefore, it can make use of the versatility of Java language, and it can support a great number of formats, including VRML, CAD, etc.

Java3D has a set of high class interfaces and libraries, which make good use of the high speed on graphic loading of many graphic cards. The calls to Java3D methods are converted into Open GL or Direct 3D functions. Though either conceptually or officially Java3D form part of API JMF, it has libraries that are installed independently of JMF. Despite Java3D does not directly support each possible 3D necessity, it permits a compatible implementation with Java code; in other cases, VRML loaders are available. They translate files from this format to appropriate objects of Java3D, which are visualized by means of an applet.

Java3D provides a high-level programming interface based on the object-oriented paradigm. This fact implies some advantages such as to obtain a more powerful, faster, and simpler development of applications.

The programming of 3D applications is based on 'scene graph models', which connect separated models with a tree-like structure, including geometric data, attributes, and visualization information. These graphs give a global description of the scene, also known as 'virtual universe' (Sowizral & Deering, 1999). This permits us to focus on geometric objects instead of on the low level triangles existing in the scene.

3.4 Comparisons

X3D takes the work carried out by VRML97 and it tackles matters that have not been specifically treated so far. From the VRML basis taken as premise, X3D provides more flexibility than VRML. The main change is the total rewriting of the specifications in three different chapters, regarding: abstract concepts, file formats, and ways to access to the programming language. Other modifications provide a greater precision in illumination and event models and they also rename some fields in order to constitute a solider standard.

The most important changes are:

(i) Expansion of the graphic capacities.
(ii) A revised and unified model of programming of applications.
(iii) Multiple files coding to describe the same abstract model, including XML.
(iv) Modular structure that permits ranges of adoption levels and support for the different kinds of market.
(v) Expansion of the specification structure.

The X3D scene graphics, the core of any X3D application, are identical to the VRML97 scene graphics. The original design of VRML graphic structure and its node types are based on already existing technology for interactive graphics. The changes initially introduced in X3D aimed at incorporating the advances in commercial hardware by means of the introduction of new nodes and types of fields for data. Later, a single unified API was developed for X3D; this means an important difference from VRML97, which had a scripting internal API apart from the external API. The X3D unified API simplifies and solves many of the problems found in VRML97, as the result of a more robust implementation.

X3D supports multiple codification archives, such as VRML97 and XML, or compressed binary (being developed at present). It uses a modular structure that provides greater extensibility and flexibility. The great majority of these applications neither need the full power of X3D nor the support for all its platforms and its functionalities defined in the specification. One of the ad- vantages of X3D is that it is organized in components that can be used for the implementation of a defined platform or specific market. For that purpose, X3D includes the concept of profiles. They are a predefined collection of components generally used in certain applications and platforms, or in scenarios like the geometric interchange between design tools. Unlike VRML97, which requires total support from the implementation, X3D allows a support for each particular need. The mechanism of X3D components also permits the companies to implement their own extensions following a rigorous set of rules.

Furthermore, X3D specification has been restructured in order to allow a greater flexibility in the life cycle of this standard, which is adjusted to its own evolution. The standard X3D is divided into three different specifications that permit ISO to achieve the adaptations of the concrete parts of the specification and their distribution in time.

One of the main differences between VRML/X3D and Java3D, at a conceptual level, is that Java3D is defined as a low-level 3D-scene programming language. This means that the creation of 3D objects in Java3D does not only require the 3D element building, but also the

definition of all the aspects related to the visualization and control of the environment capabilities. For example, for the creation of the simplest scenes, the Java3D code is notably larger than the necessary code in VRML/X3D. On the other hand, the control of the different elements in the system is more powerful and natural in Java3D. This does not mean that it is not possible to control a 'virtual universe' in VRML to include user interaction, but that it is more complex. VRML has been the favourite of most Web 3D GIS researchers for over fifteen years because it is cheap, it can provide middle-quality interactive visualization, and it has high compatibility with Java applets (Zhu et al., 2003). However, recently a growing number of engineers in the graphics and design communities are using Java3D technology (Huang, 2003; Java 3D, 2007). Because of its control power, it may be interesting to use Java3D as a VRML/X3D viewer in a TVS (Lukas & Bailey, 2002; Jin et al., 2005). It is only necessary to use some of the VRML/X3D loaders developed for Java3D. At present, the Web3D Consortium is developing under GNU LGPL (Lesser General Public License), Xj3D as a tool - completely written in Java - to show VRML and X3D contents.

The main advantages of using Java3D as a VRML/X3D viewer is its execution capability in different platforms and the fact that the final user is released of installing specific VRML/X3D plug-ins for the browser. In contrast, it must be considered the loss of speed and performance when using by Java3D vs. other VRML/X3D viewers developed in C/C++ (Burns & Wellings, 2001) and vs. viewers that directly use Direct 3D or OpenGL (Mason et al., 1999). Whichever viewer used, it must be taken into account that it is necessary to choose between internal or external programming within the VRML/X3D code. This choice is subject to the VRML/X3D implementation specification chosen by the programmer of the VRML/X3D viewer. For instance, at the level of External Authoring Interface (EAI) implementation, some VRML viewers, like CosmoPlayerTM, are based on Sun's Microsystem Java Virtual Machine, and others, like BS Contact, are based on the version of Microsoft.

4. Use of VRML for the Implementation of a TVS

4.1 Selection of the VRML Viewer

The first decision to make in a TVS implementation is to choose a VRML viewer, which must be capable of supporting and managing the great amount of data to visualize with a satisfactory performance. It is important to be very clear about this issue in order to achieve the best possible results according to the needs of the specific application to develop. In the list of Web3D viewers available on the market at present, we can specially remark, among others, those represented in Table 1.

Regarding their technological characteristics, we can distinguish between those viewers based on the use of a VRML/X3D plug-in in the browser - first column in Table 1 - and those viewers that use Java applets - second column in Table 1 -. Furthermore, we can find different non-standard format solutions that use their own technologies and file formats to store virtual environments. Although these non-standard format solutions can be better adjusted to the specific needs of a particular application at a given moment, they lack the advantage to work on an open standard that is universally recognized. So, they are subject to the decisions of the company proprietary of that format and solution. However, the use of

a system based on an open standard allows us to port our own virtual environment to other different developments that also support the standard.

VRML/X3D		Non-Standard Format
Plug-in	Java	
CosmoPlayer	AppletXj3D	Exel
Cortona	WireFusion	ViewPoint Media Player
BS Contact	3DzzD	Adobe Atmosphere
Octaga	BS Contact J	Deep View
Flux	Shout 3D	Emma 3D
FreeWRL	Blaxxun Contact3D	Cult 3D
OpenVRML		
Venues		
OpenWorlds		

Table 1. Web3D viewers

For instance, Viewpoint Media Player uses a file format based on XML, and includes the interaction capability through the use of scripting - continuous lines of interpreted commands -. Scripting vs. VRML presents a similar capability to interact directly with the environment in terms of execution time. On communicating with the Viewpoint Media Player plug-in from an HTML page, the possibility of using either JavaScript or Flash may be taken into consideration. Adobe Atmosphere and Deep ViewT M (Deep View, 2006) are different applications mainly used by Adobe to give to its PDF documents the possibility to include 3D contents.

Adobe stopped the development of Adobe Atmosphere in December 2004, and presently it uses the Deep View technology developed by HighHemisphere. In this case, Universal 3D (U3D) file format is used (Universal 3D Format, 2006). Emma 3D (Emma3D, 2006) is an open source development based on Ogre3D graphic engine and uses an archive format similar to VRML. Cult3D allows the visualization of models imported directly from 3D Studio and other formats, as well as basic animation and interaction with the scene. For example, if we use CosmoPlayer™ as a viewer, we must take into account that it is old software; that implies that it cannot make good use of the graphic capabilities of new 3D graphic cards, and it make mainly the rendering by using software instead of hardware.

On the other hand, if we use Xj3D, as well as any other viewer based on applets, we must remember that we use a viewer running in Java. Therefore, we must have the Java Virtual Machine (JVM) from Sun Microsystem installed and, according to the particular viewer, we may also need the Java3D library. In this case, to use the Java3D library allows us to accede to the graphic capabilities of 3D graphics cards available nowadays. However, using JVM involves certain declines in the performance of the application, since it is an interpreted

programming language (Barr et al., 2005) - or semi-interpreted language because a pre-compilation is carried out at bytecode level -. In the Table 2, a comparison can be observed about the loss of performance and speed of Java3D - pure Java (with 3D) in Table 2 - against other VRML/X3D developed in C/C++ and directly using Direct 3D or OpenGL - C++ (with 3D) in Table 2 -.

Elements Used	Comparison with C++
C++ (no 3D)	0%
Pure Java (no 3D)	54%
Mixed Java/C++ (no 3D)	22.5%
C++ (with 3D)	0%
Pure Java (with 3D)	92.4%
Mixed Java/C++ (with 3D)	32%

Table 2. Comparison of performances between Java and C++ (Marner, 2002)

Another important aspect on choosing a viewer is to know on which platforms it can run (see Table 3), and then, which is the possible range of users having access to the application. Let us recall here that a viewer developed in Java is multiplatform and requires the JVM of Sun Microsystem.

As commented in the first Section of this chapter, our research group has developed a virtual TVS of La Rioja that will be used to illustrate the different stages of the process of development. In the decision-making process of choosing a viewer, the use of proprietary solutions was discarded in order to make good use of the advantages of an open standard such as VRML.

As previously shown, VRML viewers can be classified into two main groups according to the technology employed: those that make use of JVM and those that incorporate plug-ins for the browser by means of ActiveX. The principal disadvantage that the former group presents against the latter one -applications compiled at machine-code level- is the loss of performance and speed (Wellings, 2004). This is the reason for discarding the use of any VRML viewer developed in Java; in general the specific needs of a TVS demands mainly high performance in refresh rates of the visualization (frames per second - FPS) and in memory use.

Finally, once the capability of the remaining viewers to execute our developed particular application was tested, we decided to use the Bitmanagement Software viewer (Bitmanagement, 2006) (BS Contact). At present, Bitmanagement Software and Octaga develop the leading viewers for the visualization of Web3D VRML/X3D technologies. The other viewers are a step behind as regards performance and updating on the development of new standards such as X3D (Bitmanagement, 2006).

VRML Viewer	Windows	Pocket PC	Linux	Mac OS
BS Contact	X	X	-	-
Cortona	X	-	-	X
Octaga	X	-	X	-
Flux	X	-	-	-
FreeWRL	-	-	X	X
CosmoPlayer	X	-	-	-
OpenVRML	X	-	X	X
Venues	X	-	-	-
OpenWorlds	X	-	-	-

Table 3. Summary of the running capability in different platforms

4.2 3D Model Creation

Once the different available Web3D technologies have been analysed and one has been selected, as well as the necessary Web3D viewer, we have to determine the specific needs of the system that we want to implement. In this stage, the first step to develop the 3D TVS is to prepare a DTEM. In order to use this model, it is necessary to obtain the terrain heights corresponding to each pair of coordinates (X, Y) in the specific area that is to be visualized.

Nowadays, there is the possibility of knowing free the height of any point on Earth with a resolution of approximately 1 kilometre. This is possible thanks to files such as GTOPO30 (Global Topographic Data horizontal grid spacing of 30 arc seconds) which provides a global digital elevation model of the U.S.

Geological Survey's Center (USGS) for Earth Resources Observation and Science (EROS) (GTOPO30, 2006). Without any doubt this is a highly useful tool, but they do not reach the desired precision for our TVS of La Rioja; so it was necessary to resort to other local databases with higher resolution. In this case, as a starting point for the generation of the DTEM, a database with a 5-metre spatial resolution was used. This database is in dBASE format and occupies several Gigabytes. In order to work with it, it was necessary to create a program that allowed consulting automatically through coordinates X and Y, represented in Universal Transverse Mercator projection (UTM) (Longley et al., 2001).

So, it was possible to sequence the process of generation of the grid on the terrain. For this purpose, a program using PERL was created, which permitted covering the whole area, a total surface of more than 5.000 Km^2, and extracting the corresponding height coordinates. Although the program worked correctly, the consulting process was too slow since the different databases used were not correctly indexed. In order to solve this problem we had to resort to a program in C++ Builder that makes the automation of the indexation of the different databases. Subsequently, another specific application was necessary to choose the step of the grid and the area from where the data would be extracted. In the code in Figure

3, it can be seen how the Borland Data Base Engine (BDBE) is used to accede to the databases by means of the use of TTable (named as Table1 in the code).

Thus, a dynamic access to the databases and to their different index archives -previously generated is achieved, which accelerates the search of heights in UTM coordinates. This piece of program shows a nested loop that accedes to the databases, which contain the terrain elevation data. Inside this loop, another nested loop allows obtaining the grid desired. The horizontal and vertical grid steps are defined by the user as input parameters of a C++ Graphical User Interface (GUI).

```
for(int j=initrow;j<=endrow;j++){
  ...
    for(int i=initcolumn;i<=endcolumn;i++){
    ---
         DB="Xy"+IntToStr(numdb); //Name of the Database
    ...
         for (int h=0;h<ypoints;h++){
        ...
              for (int k=0;k<xpoints;k++){
                   Table1->TableName=DB;
                   Table1->IndexName="UTMXY";  //Name of the file
        //of indexation
                   Table1->Open();  //Database open
                   ...
                   Table1->SetKey();  //Activation of the search
        //by key
                   Table1->FieldByName("UTMX")->AsString = utmx;
        //Set of the UTMX parameter for the search
                   Table1->FieldByName("UTMY")->AsString = utmy;
        //Set of the UTMY parameter for the search
                   if(Table1->GotoKey()){ //Test of existence of
        // some element with these UTM coordinates
                        utmz=Table1->FieldByName("UTMZ")-
        >Text+"\n"; //Obtaining of the corresponding height
                   }else{
                        utmz="0\n"; //If there is no existence then
        //set 0
                   }
                   Table1->Close(); //Database close
              ...
              }
        ...
        }
  ...
  }
}
```

Fig. 3. DTEM Creation code

Once we obtained the height of each one of the desired coordinates, the following step is to classify the resulting information into an appropriate structure for its subsequent treating and processing. In our exemplary application, it can be noted in Figure 6 that the region of La Rioja (Spain) was divided into different Zones with the aim of facilitating dynamic up-loading and down-loading according to the relative position of the observer/user. It was decided to build a dot-matrix structure totally adaptable to the step of the grid terrain at any moment for each zone.

From this point, different tests must be carried out to adjust the size of the grid step to the specific needs of each application. In our TVS of La Rioja, we are concerned with a terrain

visualization application in which a flight at a determined height is simulated; then an excessive precision is not required, as it is not going to provide anything visually important for the observer (Ayeni, 1982).

So finally, a compromise was reached for obtaining satisfactory resolution fidelity, a required level of detail, and an admissible visualization refresh rate (the previously mentioned FPS): the value of 100 meters grid step was selected. This value is more than sufficient to maintain an acceptable realism in a topographic environment, and even provides a fluid navigation.

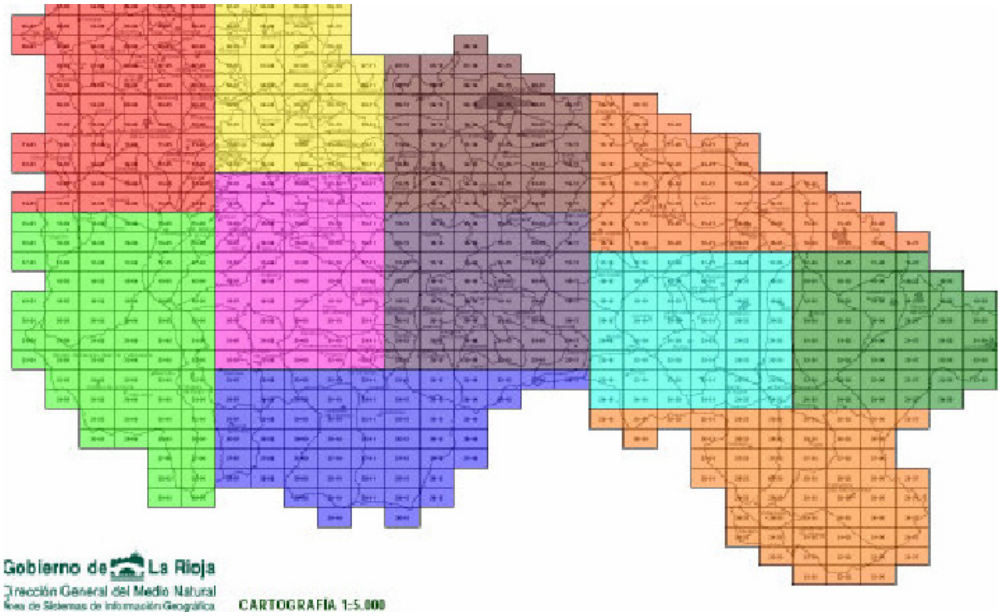


Fig. 4. Division in zones of the surface to be visualized (116x75 Km). Each colour represents a different zone

4.3 Texture Treatment

The next step to achieve a realistic TVS is to map textures on the DTEM by means of orthophotographs obtained from aerial photographs (Höhle, 1996; Ewiak & Kaczynski, 2006). Moreover, some treatments must be applied to the orthophotographs before mapping them, in order to optimise the implementation. For example, in our TVS of La Rioja, the orthophotos were obtained from the Autonomous Government of La Rioja, in JPG format, covering 10 Km^2 of surface everyone. When carrying out a simulation of a VRML environment, it is necessary that the model files and texture files are not excessively large.

That is the reason why each scene is stored in dot-matrix file, which cover an extension of only 1 Km^2. Therefore, the texture/orthophotographs have to be divided into partitions. A

specific C++ program was employed to obtain these partitions. This application allows the user to define on the orthophotos some parameters, such as their origin and destination directories, their new names once they are partitioned and new size. It considers their original size in case of being necessary to use data from some of them.

Another treatment required is to adjust the optimum resolution of the textures for its further visualization during the simulation. In order to find this optimum resolution, some tests were carried out from 0.5 m/pixel up to 4.0 m/pixel. Finally, it was proved that using resolutions lower than 2 m/pixel does not really provide a more realistic scene, due to the texture treatment that the different viewers carry out. Furthermore, to use high resolution textures implies that the files are heavier, more work for the viewers, and a lower visualization refresh rate (FPS). With all these issues in mind, we opted for 2 m/pixel as an appropriate texture resolution for our particular application.

Another important aspect in order to accelerate the simulation is the inclusion of levels of detail (Blow, 2000; Luebke et al., 2003), so that it is possible to lighten the viewer load without losing realism or quality for the observer user. The level of detail was established with the use of different resolution textures depending on the distance from the observer to them.

(i) From 0 to 1,500 metres: 2 m/pixel resolution.

(ii) From 1,500 to 5,000 metres: 4 m/pixel resolution.

(iii) From 5,000 metres to eye-sight reach: without texture and only the net-meshing is observed.

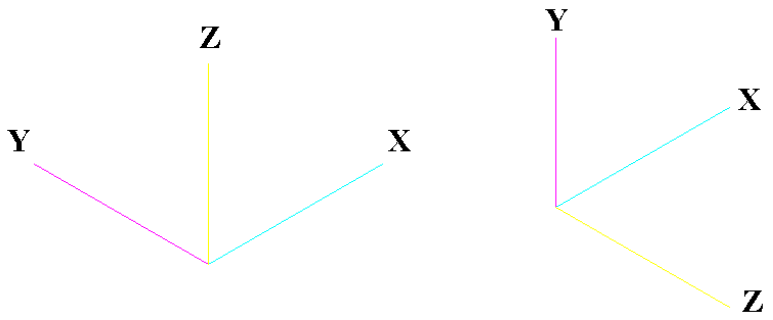


Fig. 5. Orthophotograph reference system

Another noteworthy treatment of the orthophotographs is their reference coordinates. The orthophotographs are represented in UTM (see Figure 5a) and they have to be adjusted to the coordinate system used in the VRML simulation (see Figure 5b). For that reason, it is necessary to make a 180º clockwise turn of the orthophotographs with a software application (for instance, it is sufficient to use any photo-editor application). In this way, orthophotograph UTM coordinate system and VRML environment system fit one another perfectly. Once the texture files are prepared, the graphic engine or the web3D viewer can read them for a realistic visualization.

4.4 Tools for implementing the user interaction

What provide VRML with the capability of implementing the user interaction is the JavaScript Code (usually denominated VRMLScript) and the Java code. On the one hand writing VRMLScript code requires the incorporation of different interaction elements within the VRML virtual scene; on the other hand it is possible to interact with the scene from some external applets with Java code.

Thus, the programmer is completely free to create his own user interface with Java libraries and to connect with the virtual scene with the EAI library (Gosling et al., 1996; Yu et al., 2005; Sowizral & Deering, 2005).

For the use of VRMLScript or Java, it is necessary to resort to VRML package libraries (see Figure 7): whilst VRMLScript needs the vrml.node and vrml.field packages, when using Java applets, vrml.external library is required. Let us focus on the use of applets to connect with VRML scenes by means of EAI library. The following elements are examples that may be taken into account in order to pro-vide a TVS with interactivity; in fact, they were used in the TVS of La Rioja (see Figure 6):

(i) HTML file: in the own HTML page, the reference to the VRML file and to the applet must be included. The reference to the VRML files is made with the label '<embed src="scene.wrl"...>' and the applets are usually inserted using the label '<applet width="150" height="40" code="joystick.class">'.

(ii) Applets: they present their usual generic code, but they must also include the necessary code to communicate with the VRML scene. This communication is achieved with the creation of an instance in the Browser class. By means of these instances, we can accede to the VRML scene and control it.

(iii) References to the nodes: in order to read certain information from the VRML scene or to control certain parameters, it is necessary to make reference to a specific node that contains this information or parameters. To achieve this, the getNode 'Node node1 = browser.getNode ("Control-Position")' is used. This way, we can make reference to a specific node that has been previously defined in the scene by means of the key-word 'DEF'.

(iv) Reading/Writing of the VRML scene: Once a specific node is referenced, their fields can be acceded with the functions getEventIn(String) and getEventOut(String): 'Orientation = (EventOutSFRotation) node1.getEventOut ("orientation changed");'. The use of these functions is limited to access to the fields defined as eventOut and exposedField. Once the reference to the field is created, its value can be read by means of getValue(): 'orientation = Orientation.getValue();' or it can be written with the function SetValue(): 'avatarSize.setValue(size);'.

(v) Receiving VRML scene events: In the case of needing to receive events produced by the scene, we must implement in our applet the interface EventOutObserver: 'public class compass extends Applet implements EventOutObserver'. The next step in the definition of the applet is to overwrite the callback method, which will reply each event produced in the scene: 'public void callback(EventOut eventout, double d, Object obj)'.

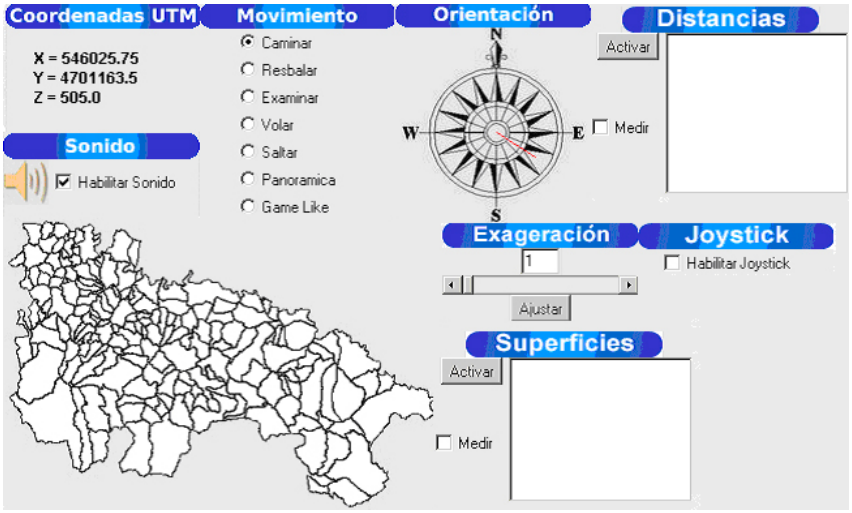


Fig. 6. External applets for interaction with the user

In Figure 6, the different applets that have been developed for our exemplary application are shown:

(i) UTM Coordinates: this applet shows at any time the position of the user in UTM coordinates.

(ii) Orientation: with this element, the user can know the angle from where he is watching.

(iii) Movement: this panel permits to change rapidly the way we move in the 3D scene (walking, sliding, inspecting, flying, jumping, panoramic view, Game-like view).

(iv) Sound: this application allows to activate or deactivate the background sound.

(v) Distances: with this applet, the zenith view and measure distances in two or three dimensions can be activated.

(vi) Surfaces: the option to measure the surface defined with a series of points pointed out in the scene can be used.

(vii) Exaggeration: this applet permits to increase the height, exaggerating it in order to watch the size of the different heights increased.

(viii) Joystick: it allows activating or deactivating the navigation or the scene control by means of a Joystick added to the computer.

(ix) Map: this applet includes a map of the terrain and shows us at any moment the position and orientation of the user. Besides, it permits us to move directly to any point on the map simply by clicking on it.

```
import vrml.external.Browser; //Importation of VRML libraries
import vrml.external.Node;
...
public class brujula extends Applet
    implements EventOutObserver, Runnable
{
    ...
    //Applet initialisation
    public void run()
    {
        if(browser == null)
        {
            //Reference to the VRML viewer
            for(; browser == null; browser = Browser.getBrowser(this))
                try
                {
                    Thread.currentThread();
                    Thread.sleep(5L);
                }
                catch(InterruptedException ex) { }

            ...
            //Reference to the node of position control
            Node node1 = browser.getNode("PositioonControl");
            //Reading of the output events of the orientation changes
            Orientation =
(EventOutSFRotation)node1.getEventOut("orientation_changed");
            //Definition of this applet to answer to the events
            Orientation.advise(this, null);
            ...
        }
    }
...
    //Answer of the output events generated by the VRML viewer and
    //associated to this applet
    public void callback(EventOut eventout, double d, Object obj)
    {
      //Reading of the present orientation value
      orientation = Orientation.getValue();
      //Orientation treatment
      ...
    }
```

Fig. 7. Basic structure for interconnection between an applet and the VRML scene

5. Graphic Engine for a TVS

5.1 Open Source Libraries for 3D Visualization

An alternative solution to the Web3D viewers for the TVS performance is to develop a program known as graphic engine. It executes graphic routines by calling methods from specialized libraries, like Open inventor TM, Coin3D, or OpenSceneGraph, among others. Let us focus on this latter mentioned library to explain some of their common characteristics. OpenSceneGraph (OSG) (OpenSceneGraph, 2006) is a recently-developed graphic library which incorporates the different primitive basic concepts of OpenGL (Mason et al., 1999). This library uses the programming language C++, because it is independent of the platform

and open source. Among the possible uses of this library we find simulations, scientific visualizations, virtual engineering and game development.

OpenSceneGraph uses scene graph techniques to contain all the information regarding the scene generated. A scene graph is a data structure that permits the creation of a scene hierarchic structure, keeping the father-son connections among the different elements. For example, variations of position and orientation in the father node affect son nodes; thus, an arm robot with several joints can be created, with each piece dependent on the previous one, and simply applying movement to the initial piece; the rest of the dependent pieces will automatically move according to the defined structure.

Another important father-son relationship exploited by the scene graph techniques is the possibility of defining enveloping volumes, which gather close elements. Thus, during the process of rejection of the elements that will be represented on screen, it is not necessary to analyze the children of a node father already rejected.

Although the library is still under development at present, it is being employed by different users in research and commercial applications. There are even other open source developments, which extend and give support to the capabilities of OpenSceneGraph, such as OpenProducer (OpenProducer, 2006) or VT Jugger (VR Juggler, 2006), for instance. OpenProducer is a library that permits the treatment of different current representation systems and interaction devices.

Although there is other libraries providing this support, OpenProducer offer the advantage of being an open source development and it allows working with the OpenScreneGraph library. VR Juggler is a technology that supplies necessary tools for the development of virtual reality applications. It constitutes a virtual platform for the development of applications that are applicable to nearly all the virtual reality systems.

5.2 Graphic Engine Development

Once the graphic library is selected, the graphic engine consults the 3D model; so, it must be optimized for satisfactorily fast terrain visualization. At software level, the way to deal with this matter is to maintain at any time a similar amount of information (textures and DTM), which allows a fluid data management.

For this purpose, let us resort to a dynamic up-loading and down-loading of the scene according to the position and direction of the user at each moment. This process is carried out with a database that paginates the different scene areas and allows us to decide which parts are necessary to bear in mind at each moment (see Figure 8). At a theoretical level, the database limit is not defined, but in practice, the larger the size, the less performance, and the dynamic up-loading and down-loading are affected. That is the reason why a database restructuring was carried out in our TVS of La Rioja, regarding communities and provinces.

The restructuring consists in searching for a way to limit the database size to the specific needs at each moment according to the zone where the user goes. In this way, it was

possible to maintain a more or less constant loading process with a reasonable order for a desktop PC with the following minimum characteristics: Pentium 4 1,7 GHz, 80 GB ATA 100 Hard-Drive, 512 Mb DDRAM, Ati Rage 128 Pro II. In order to achieve this, we can set several levels of texture resolution with a PagedLOD node. So, according to the distance between the observer view-point and the model, the appropriate level will be visualized. The PagedLOD node (see Figure 9) is a variant of the LOD node, but it also allows the up-loading and down-loading of the memory of the scene elements. Then, the scene graph that the system has to manage during the rendering process is smaller.

```
osgDB::DatabasePager* databasePager = osgDB::Registry::instance()-
>getOrCreateDatabasePager();
databasePager->registerPagedLODs(root);
sceneView->getCullVisitor()->setDatabaseRequestHandler(databasePager);
```

Fig. 8. Activation of the dynamic up-load and down-load system

```
PagedLOD {
      DataVariance DYNAMIC
      nodeMask 0xffffffff
      cullingActive TRUE
      Center 488500 4.694e+006 0
      Radius -1
      RangeMode DISTANCE_FROM_EYE_POINT
      RangeList 3 {
        5000 15000
        1000 5000
        0 1000
      }
      NumChildrenThatCannotBeExpired 0
      FileNameList 3 {
        f29c1_C.ive
        f29c1_B.ive
        f29c1_A.ive
      }
      num_children 0
    }
```

Fig. 9. PagedLOD node of a scene element

The rendering in OpenSceneGraph is divided into three stages. The first stage is the Update Process, in which changes in the scene graph regarding execution time are made. The second stage is the Cull Process, in which the list of scene elements that will be rendered in the next stage is set. Finally, the Draw Process is the last stage. With this structure, and the ranges for rendering appropriately chosen for each level, constant refresh rates from 20 to 30 FPS were achieved, even during the loading of the application (see Figure 10).

On the other hand it is important, although not strictly necessary, to generate the whole geometry of the scene in the binary format of OpenSceneGraph. This binary format (IVE) facilitates the initial process of scene loading, so the waiting time that users spend in loading the application is reduced.

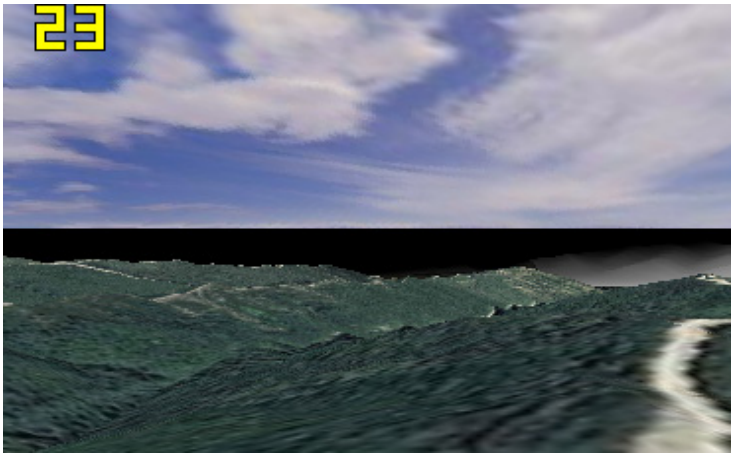


Fig. 10. Screen capture of the application at an initial moment and FPS

```
class MyKeyboardMouseCallback : public Producer::KeyboardMouseCallback
{
public:
    MyKeyboardMouseCallback(osgUtil::SceneView* sceneView) :
        Producer::KeyboardMouseCallback(),
        _mx(0.0f),_my(0.0f),_mbutton(0),
        _done(false),
        _trackBall(new Producer::Trackball),
        _sceneView(sceneView)

    {
      ...
    }
      ...

    osg::Matrixd getViewMatrix()
    {
      ...
      // Utilization of the IntersectVisitor
      osgUtil::IntersectVisitor iv;
      // Creation and initialisation of the row with present and next
      // positions
      osg::ref_ptr<osg::LineSegment> segNormal = new osg::LineSegment;
      segNormal->set(fp,lfp);
      // Addition of the row to the IntersectVisitor
      iv.addLineSegment(segNormal.get());
      // Launch of the Visitor on the scene
       sceneView->getSceneData()->accept(iv);
      // Verification of collision detection and consequent actuation
        if (iv.hits())
        {
        ...
        }
      ...
      return osg::Matrixd( trackBall->getMatrix().ptr());
    }
...
};
```

Fig. 11. Terrain collision detection

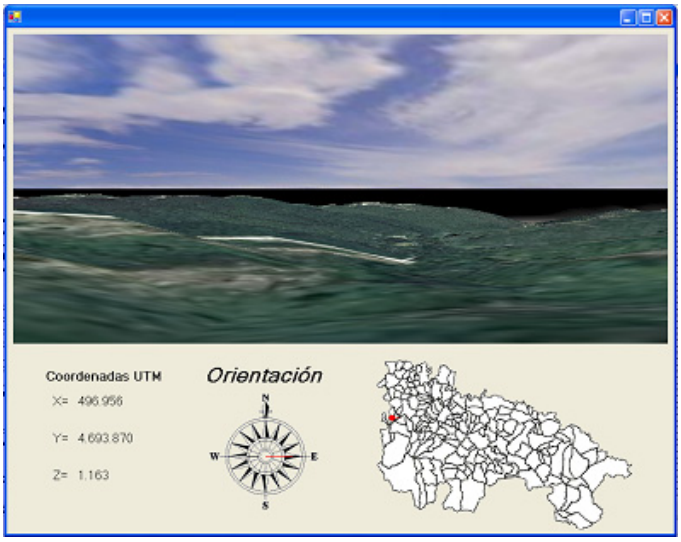


Fig. 12. User interface

5.3 User Interaction Tools

One of the most important aspects in a graphic application of a TVS is to provide the user with an easy and friendly navigation. For this purpose, user interaction is usually implemented by means of the mouse, so the cursor can be moved over the scene on screen.

Therefore, it is required to endow the application with the capability of receiving and responding to the mouse events happened on the scene. This can be achieved by creating a new class from the class 'Producer::KeyboardMouseCallback' and implementing the appropriate methods to define the interaction between this object and the rest of the objects in the environment.

For example, in a TVS, the user must always be above ground level, and, generally, at a determined height. Then, it is necessary to use a collision detection system, which prevents the user from going through the ground as he moves over the scene. For this purpose we used a specifically designed visitor in OpenSceneGraph: 'IntersectVisitor'. The visitor is a design patter of performance, which permits to define the operations that will be applied to the elements of a heterogeneous structure of objects.

A design pattern in programming is just the structure or the core of the solution of a common problem of software development. In this case it is an algorithm that allows determining whether a certain segment/line intersects any point of the geometry of the object that accepts the visitor. In the code in Figure 11, the results of the operations executed by IntersectVisitor are sent and accepted by the object that stores the geometries of the scene 'sceneView → getSceneData()→ accept(iv)'. In the code in Figure 11, the results of the operations executed by IntersectVisitor are sent and accepted by the object that stores the geometries of the scene 'sceneView→ getSceneData()→ accept(iv)'.

Apart from facilitating 3D visualization of the scene, it is also necessary to create a user-friendly interface (see Figure 12). With this interface, the user can interact in a simple way, and it provides him with the capability to control or to obtain information from the scene. In Figure 13, the screen refresh loop in C++ that has been used in our TVS is shown. The observer position can be obtained and presented on screen at any moment by means of the last two sentences in C++ code shown in Figure 13.

6. CONCLUSIONS AND FUTURE WORK

In this chapter we have set out the main steps to be followed for the development of an application for terrain visualization.

```
    while( renderSurface->isRealized()&& !kbmcb->done() )
    {
     // set up the frame stamp for current frame to record the current
     // time and frame
     // number so that animtion code can advance correctly
        osg::ref_ptr<osg::FrameStamp> frameStamp = new
osg::FrameStamp;
        frameStamp->setReferenceTime(osg::Timer::instance()-
>delta_s(start_tick,osg::Timer::instance()->tick()));
        frameStamp->setFrameNumber(frameNum++);
     // pass frame stamp to the SceneView so that the update, cull and
     // draw traversals all use the same FrameStamp
        sceneView->setFrameStamp(frameStamp.get());
     // pass any keyboard mouse events onto the local keyboard mouse
     // callback.
       kbm->update( *kbmcb );
     // set the view
        sceneView->setViewMatrix(kbmcb->getViewMatrix());
     // update the viewport dimensions, incase the window has been
     // resized.
        sceneView->setViewport(0,0,renderSurface-
>getWindowWidth(),renderSurface->getWindowHeight());
      // do the update traversal the scene graph - such as updating
      // animations
        sceneView->update();
      // do the cull traversal, collect all objects in the view
      // frustum into a sorted
      // set of rendering bins
        sceneView->cull();
      // draw the rendering bins.
        sceneView->draw();
      // Swap Buffers
            renderSurface->swapBuffers();
      // Reading of the present position value
       sceneView->getViewMatrixAsLookAt(eye, center, up, 0.0f);
      // Observer position updating
       refresCoord(eye, center, up);
      }
```

Fig. 13. C++ code for the screen refresh loop

First, the basic characteristics of a TVS and a diagram of the development process have been presented. Then, a brief study of the different applicable Web3D technologies (VRML, X3D, or Java3D) has been included. With this base, the different necessary steps to realize a TVS

based on VRML have been detailed, from the choice of the viewer to the incorporation of external tools in Java for the interaction with the user. Aspects such as DTEM creation and the inclusion of photorealistic textures to the model have been also explained.

From the aforementioned, it can be deduced that the use of VRML for the creation of terrain visualization is viable, but always at the expense of depending on an external element that takes over the scene visualisation, on which neither real control exists, nor its code is known, nor its program can be modified. In order to overcome these limitations, the development of a specific graphic engine by means of the use of open source libraries has been proposed. To achieve this aim, several available open source graphic libraries and their basic functioning characteristics have been analysed.

Finally, the steps followed in the implementation of a TVS by means of OSG library have been detailed. Several tips and codes are also involved in this chapter to illustrate some stages in the development process. One of the potential capabilities of the developed system, which may be implemented in the future, is to include different 3D geometries, with the final purpose of facilitating aspects such as management, distribution and planning of the terrain, as a further onward step in geographical information systems (Longley et al., 2001). One practical example may be the inclusion of tracing of roads in the design stage (see Figure 14).

The starting point would be the road design maps, in this particular case, or any other element on which one may be working (canalisation, electricity posts and lines, railways and other forms of communication networks, etc.). From these graphic maps a previous treatment should be carried out, with any of the 3D design tools that exist on the market (3D Studio, Mayer, Blender, etc).

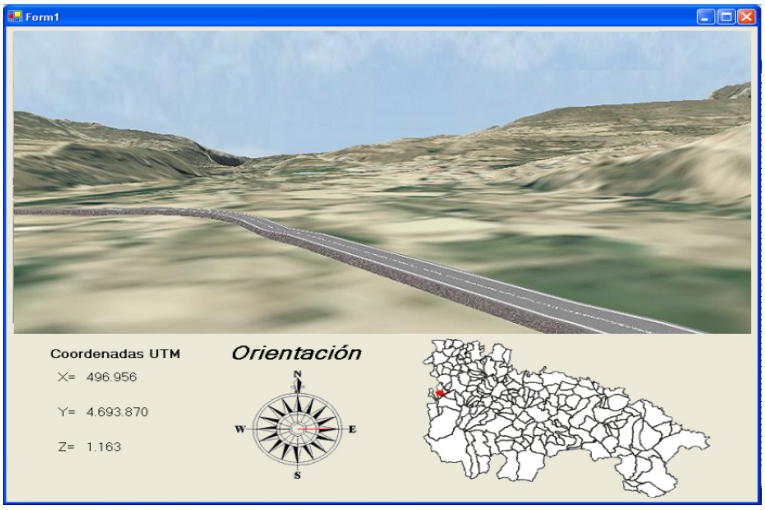


Fig. 14. Example of the incorporation of external geometries

7. Acknowledgements

To the Autonomous Government of La Rioja, Ministry of Tourism, Natural Environment and Terrain Policy, for making available to us digital orthophotographs and terrain digital models, which enabled us to develop this simulation.

To GER (Grupo Eólicas Riojanas) for the partial financing of this project and for the interest shown in applying simulation in its own specific field of activity.

8. References

Ayeni, O.O. (1982). Optimum sampling for digital terrain models: a trend towards automation, *Photogrammetric Engineering & Remote Sensing*, 48, 11, 1687–1694.

Barr R.; Haas Z.J. & Van Renesse R. (2005). JiST: an efficient approach to simulation using virtual machines. *Software – Practice and Experience*, 35, 6, 540–576.

Bell G.; Parisi A. & Pesce M. (1995). *The Virtual Reality Modeling Language: Version 1.0 Specification,* Technical Report.

Bernardin T.; Cowgill E.; Gold R.; Hamann B.; Kreylos O. & Schmitt A. (2006). Interactive mapping on 3-D terrain models. *Geomistry Geophysics Geosystems*, 7, 10.

Bernhardt Saini-Eidukata B.; Schwerta D.P. & Slator B.M. (2002). Geology explorer: virtual geologic mapping and interpretation. *Computers & Geoscience*, 28, 10, 1167–1176.

Bishop J. & Horspool N. (2004). Developing Principles of GUI Programming Using Views, *In Proceedings of SIGCSE'04*, pp. 373–377, Norfolk (VA-USA), March.

Bitmanagement Software. http://www.bitmanagement.de/ [26 July 2006].

Blow J. (2000). Terrain rendering at high levels of detail, *In Proceedings of Games Developers Conferences*, San Jose (CA-USA).

Bradley J. (1999). An Efficient Modularized Database Structure for a High-resolution Column-gridded Mars Global Terrain Database. *Software- Practice and Experience*, 29, 5, 437–456.

Burns A. & Wellings A.J. (2003). *Real-time systems and programming languages*, Addison-Wesley, isbn 0201729881.

Burrows A.L. & England D. (2002). Java 3D, 3D graphical environments and behaviour. *Software-Practice and Experience*, 32, 4, 359–376.

Deep View. http://www.righthemisphere.com/company/links/solutions/Adobe/index pdf.htm [25 July 2006].

Döllner J.; Baumman K. & Hinrichs K. (2000). Texturing techniques for terrain visualization, *In Proceedings of Conference on Visualization'00*, pp. 227–234, Los Alamitos (CA-USA.

Duke D.J.; Brodlie K.W.; Duce D.A. & Herman I. (2005). Do You See What I Mean?. *IEEE Computer Graphics and Applications*, 25, 3, 6–9.

Ellul C. & Haklay M. (2006). Requirements for Topology in 3D GIS. *Transactions in GIS*, 10, 2, 157–175.

Emma3D. http://www.emma3d.org/ [25 July 2006].

ETOPO2v2 DEM. http://www.ngdc.noaa.gov/mgg/fliers/06mgg01.html [26 August 2006].

Ewiak I. & Kaczynski R. (2006). True ortho-sat vs. aerial. *GEO: connexion*, 5, 5.

Fairbairn D. & Parsley S. (1997). The Use of VRML for cartographic presentation. *Computers & Geoscience*, 23, 4, 475–481.

Fencík R.; Vajsáblová M & Vaníková E. (2005). Comparison of interpolating methods of creation of DEM, *In Proceedings of 16th Cartographic conference*, pp. 77–87, Brno (Czech Republic).

g3DGMV. 3D Graphical Map Viewer. http://g3dgmv.sourceforge.net [13 July 2006].

GeoInformation Technologies Group, 2002. Digital elevation models and digital terrain models. Swiss Federal Institute of Technology, Zurich, Switzerland. http://www.geoit.ethz.ch/education/presentations/dem dtm/ [08 August 2006].

Geroimenko V. & Chen C. (2005). *Visualizing Information Using SVG and X3D, XML-based technologies for the XML-based Web*, Springer-Verlag, isbn 1852337907.

Gittings B.M.; 1996. Digital Elevation Data Catalogue. http://www.geo.ed.ac.uk/home/ded.html [04 January 2007].

Gosling J.; Joy B. & Steele G. (1996). *The Java Language Specification*, Addison-Wesley, isbn 0201634554.

GTOPO30 DEM. http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html [26 August 2006].

Guedes L.C.; Gattass M. & Carvalho P.C.P. (1997). Real-time rendering of phototextured terrain height fields, *In Proceedings of SIBGRAPI 97*, pp. 18-25, Campos de Jordao (SP-Brazil).

Guillen A.; Meunier C.H.; Renaud X. & Repusseau P.H. (2001). New Internet tools to manage geological and geophysical data. *Computers & Geoscience*, 27, 5, 563–575.

Hay R.J. (2003). Visualisation and Presentation of Three Dimensional Geoscience Information, *In Proceedings of 21st International Cartographic Conference,* Durban (South Africa).

Heckbert P.S. (1986). Survey of Texture Mapping. *IEEE Computer Graphics & Applications*, 6, 11, m-y, 56–67.

Hirtz P.; Hoffmann H. & Nuesch D. (1999). Interactive 3D landscape visualization: improved realism through the use of remote sensing data and geoinformation, *In Proceedings of Computer Graphics International*, pp. 101-108, Cannmore (Alberta-Canada).

Hirtz P.; Hoffmann H. & Nuesch D. (2006). Evaluating X3D for use in software visualization, *In Proceedings of ACM symposium on Software visualization*, pp. 161-162, Brighton (United Kingdom).

Höhle J. (1996). Experiences with the production of digital orthophotos. *Photogrammetric Engineering and Remote Sensing*, 62, 10, 1189–1194.

Huang B. (2003). Web-based dynamic and interactive environmental visualization. *Computers, Environment and Urban Systems*, 27, 6, 623–636.

Huang B. & Lin H. (1999). GeoVR: a web-based tool for virtual reality presentation from 2D GIS data. *Computers & Geoscience*, 25, 10, 1167–1175.

Huang B. & Lin H. (2002). A Java/CGI approach to developing a geographic virtual reality toolkit on the Internet. *Computers & Geoscience*, 28, 1, 13–19.

Java 3D Applications. http://www.j3d.org/sites.html [24 January 2007].

Jin B.; Bian F.; Zuo X.; Wang F., 2005 Study on visualization of virtual city model based on Internet. Geo-Spatial Information Science ; 8(2):115–121.

Kersting O. & Döllner J. (2002). Interactive 3D visualization of vector data in GIS, *In Proceedings of tenth ACM international symposium on Advances in geographic information systems*, pp. 107–112, McLean (VA-USA).

Kreuseler M., 2000. Visualization of geographically related multidimensional data in virtual 3D scenes. Computers & Geoscience; 26(1): 101–108.

Kumler M.P., 1994. An Intensive Comparison of Triangulated Irregular Networks (TINs) and Digital Elevation Models (DEMs). Cartographica; 31(2):1–99.

Lindstrom P. & Pascucci V. (2001). Visualization of large terrains made easy, In Proceedings of IEEE Visualization, pp. 363-371, Piscataway (NJ-USA).

Lin H.; Gong J.; Wang F., 1999. Web-based three-dimensional geo-referenced visualization. Computers & Geoscience; 25(10): 1177–1185.

Longley P.A.; Goodchild M.F.; Maguire D.J. & Rhind D.W. (2001). *Geographic Information Systems and Science*, Wiley, isbn 0471892750.

Luebke D.P. et al. (2002). *Level of Detail for 3D Graphics: Application and Theory. Terrain Level of detail*, Morgan Kaufmann, isbn 1-55860-838-9.

Lukas S. & Bailey M. (2002). FlySanDiego: A Web-aware 3D interactive regional information system, *In Proceedings of SPIE - The International Society for Optical Engineering*, pp. 368–378, San Diego (CA-USA).

Marner J. (2002). *Evaluating Java for Game Development*, Technical report.

Marschallingera R.; Johnson S.E., 2001. Presenting 3-D models of geological materials on the World Wide Web. Computers & Geoscience; 26(1): 467–476.

Mason W and others, 1999. OpenGL programming guide. Addison-Wesley. Moore K., Dykes J., Wood J., 1999. Using Java to interact with geo-referenced VRML within a virtual field course. Computers & Geosciences ; 25(10):1125–1136.

OpenProducer. Introducing OpenProducer. http://www.andesengineering.com/Producer [26 August 2006].

OpenSceneGraph. http://www.openscenegraph.org [26 August 2006].

Sowizral H.A.; Deering M.F., 1999. The Java 3D API and virtual reality. IEEE Computer Graphics and Applications ; 19(3):12–15.

Sowizral H.A.; Deering M.F., 2005. Virtual university three-dimension query system based on VRML and java technology. Computer Engineering; 31(6):173–175.

Sowizral H.; Rushforth K.; Deering M., 1998. *The Java 3D API Specification*. Addison-Wesley.

Universal 3D Format. http://www.intel.com/technology/systems/u3d/ [25 July 2006].

VR Juggler: About the Juggler Suite of Tools. http://www.vrjuggler.org/about.php [26 August 2006].

VR Juggler: The Programmer's Guide. http://www.vrjuggler.org [26 August 2006].

Web3D Consortium - Open standards for real-time 3D communication. http://www.web3d.org/ [22 July 2006].

Wellings A.J., 2004. Concurrent and real-time programming in Java. Wiley. Wu Q., Xu H., 2003. An approach to computer modeling and visualization of geological faults in 3D. Computers & Geoscience; 29(4): 503–509.

Yu C.; Wu M. & Wu H. (2005). Combining Java with VRML worlds for web- based collaborative virtual environment, *In Proceedings of IEEE Networking, Sensing and Control, Tucson*, pp. 299–304, Arizona (USA).

Zhu C.; Tan E.C. & Chan K.Y. (2003). 3D Terrain visualization for Web GIS, *In Proceedings of Map Asia*, Kuala Lumpur (Malaysia).

9

An Interactive Fire Animation on a Mobile Environment

DongGyu Park, SangHyuk Woo, MiRiNa Jo
Dept. of Information and Communications Engineering,
Changwon National University, Changwon City, 641-773, GyeongNam, Korea

DoHoon Lee
School of Computer Science and Engineering,
Pusan National University, 609-735, Pusan City, Korea

1. Introduction

One of the most difficult problems for computer graphics system is the physics based fluid simulations. Accurately rendering fire, smoke, and water is a challenging problem due to various subtle ways in which the fluid mechanics interact with this complex participating medium. Fluid mechanics is used as the standard mathematical framework on which these simulations are based. There is a consensus among scientists that the Navier-Stokes equations are a very good fluid flow model. Thousands of articles and books are published in many areas to compute Navier-Stokes equations with numerical methods.
Also, many fluid solvers have been proposed to compute the Navier-Stokes equation in realtime. However, the complexity of fluid formation, dynamics, and light interaction make fire and smoke simulation and rendering difficult in realtime. In an interactive fire and smoke simulation, users would like to look around and pass through them. Also, simulated fire and smoke should be realistically illuminated by light and in its own light.
The recent rapid increase in speed and programmability of graphic processors has enabled us to use graphic processing units (GPU) for more than just rendering fluids. In addition, the GPU implementation of variety for physically-based simulation outperforms implementations that perform all computations on the CPU.
In computer games, fluid-like animation, including fire and smoke animation, are commonly used for augmenting realistic scenes. As the gaming industry produces new games on mobile platforms, realistic-looking fluid simulation has become more and more important. Also, the development of mobile hardware enables us to run various application software, including interactive 2D or 3D games on mobile handsets. Although fire and smoke simulations are widely used on personal computer platforms, interactive physically-based simulation of fluids still has many problems on mobile platform. We have established effective physically-based fire and smoke simulation techniques in a mobile 3D environment.

Figure 1 shows some snapshots of our fire and smoke animation based on physically-based modeling and billboard layout in a mobile environment. The simulations are adopted on our mobile 3D game "Rupee Story," which is a first-person adventure game. The motion of the game characters is controlled by the arrow keypad, and we can shoot using the "OK" key on the keypad. Also we can control the camera while playing the game.

Fig. 1. Snapshots of fire and smoke animation in a mobile environment.

Our fire simulation and its applications are implemented on a WIPI (Wireless Internet Platform for Interoperability), which is the standard mobile platform in Korea. Based on the WIPI platform, there are several extended platforms for mobile 3D contents, such as NF3D, M3D and G3D. Most of the mobile 3D platforms are based on the OpenGL-ES (Embedded System) Engine. Therefore, all of our implementations are based on the NF3D platform.

2. Previous Works

In this section, we present an overview of the previous techniques for fluid simulation and rendering. Modeling and animating fluids have captured the attention of many graphics researchers. However, no one has created general fluid models that are physically realistic and computationally efficient for real-time animation. Fluid simulation is a useful building block that is the starting point to simulating a variety of natural phenomena. One of the most popularly reviewed examples of computer-generated fire appears in the movie Star Trek II : The Wrath of Khan(Paramount, 1982).

Fournier and Reeves first introduced particle systems to models and rendered fuzzy objects such as fire, where the particles are motion-blurred in order to avoid temporal aliasing or strobing(Reeves, 1983; Shinya & Fournier, 1992).

A number of simulation-based methods for generating realistic fluid animation have been proposed. These techniques use computational nodes that are either fixed in space (Eulerian) or that move with the fluid (Lagrangian). Stam introduced a semi-Lagrangian technique for velocity advection coupled with a projection method for enforcing mass conserving. While this approach is unconditionally stable, it suffers from mass dissipation and excessive numerical damping, especially of the vortices that are so interesting in fluid flows(Stam, 1999).
In "Visual Simulation of Smoke" Fedkiw et al. combat this using a relatively new technique from the Computational Fluid Dynamics literature called vorticity confinement and higher order interpolation. In Computational Fluid Dynamics, the vorticities are detected and receive augmenting forces(Fedkiew et al, 2001). Foster and Fedkiw interpolated this into a water solver and added a level set-augmented by using marker particles to counter mass loss-for high quality surface tracking(Foster & Fedkiew, 2001).
Selle et al. proposed a hybrid method between the grid based method and particle method for enhancing vorticity confinement, which produces a good turbulent flow(Selle et al, 2005). They implemented large rolling explosion effects and smoke explosion using a grid and large vortex particles.

3. Stam's Stable Navier-Stokes Solver

The incompressible Navier-Stokes equations can be written as

$$\frac{\partial \mathrm{u}}{\partial t} = -(\mathrm{u}\cdot\nabla)\mathrm{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\mathrm{u} + \mathrm{f}\,. \tag{1}$$

$$\nabla\cdot\mathrm{u} = 0\,. \tag{2}$$

with velocity u=(u, v, w), pressure p, constant fluid density ρ, kinematic viscosity, where f represents external forces that act on the fluid. Buoyancy, vorticity confinement, and gravity forces are good examples of external forces. Equation (1) is the momentum equation and equation (2) is the continuity equation. Equation (1) represents acceleration due to advection, pressure, diffusion, and external forces, respectively. Adding vorticity confinement and buoyancy as external forces produces a good appearance of turbulent flow Selle et al, 2005).
Stam described a stable numerical technique for interactive simulation of fluid motion(Stam 1999). Our fluid simulation solver is based on Stam's algorithm, which is stable and prevents numerical dissipation. Stam's stable fluid simulation solver is based on the Helmholtz-Hodge decomposition theorem(Stam 1999; Stam 2000, Stam 2002). The theorem states that any vector field w can uniquely be decomposed into the form:

$$w = u + \nabla q \tag{3}$$

where u has zero divergence. Here, u and q is a scalar field. Equation (3) means that any vector field is the sum of the mass conserving field and a gradient field. The equation can be reduced to a Poisson equation for scalar field q.
There are several kinds of fluids like water, smoke, cloud, fire, and mud. We implemented smoke and fire effects on our mobile games. The smoke and fire simulation can be easily

simulated on a billboard, because they can be easily simulated on regular grids. Also, they can be simulated using the same simulation equation, the fire equation.

3.1 Fire Equation

Generally, fire consists of the following components: fuel (F), heat (H), oxygen (O) and inert gas (G). If the fuel is hot enough and there is sufficient oxygen, then the fuel will react with oxygen. We assume that there always will be enough oxygen to react with the fuel gas, which simplifies the combustion computation. Using them up, they create heat and waste. These burning procedures are simplified when interacting with three arrays. The rate of reaction depends on the concentration of the reactants and temperature; in addition, the initial value of the fuel and heat are controlled by the game story. If we want to grow or shrink a fire, we can control it by providing much fuel and heat.

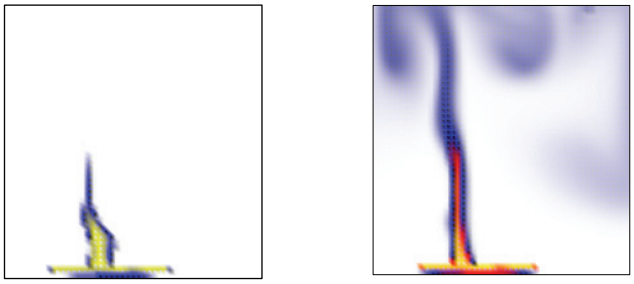
The next step is the convection step; here, we have some heat sources, which causes convection. The temperature can be converted into convection forces. There are several constants, such as an energy barrier, a rate constant, a max rate, and exothermicness, which controls various aspects of the reaction.

3.2 Coloring the Fire and Smoke

The last step is rendering. Fire has heat sources which emit light. The hotter the gas, the more energy is given off. Slightly hot gas gives off a mostly long wavelength light (infra red). As the temperature increases, it begins to give off light of shorter wavelengths: the colors are red, green (yellow), blue, and white.

The colors of the flame change, depending on its heat sources and chemical reactions. In our simulation, we implemented the gradient class to simulate various heat sources. The temperature of the fire determines the color of the flame. The fire color gradients are useful for determining various kinds of gaseous phenomena.

Figure 2 shows that various colors are implemented on our gradient class. Compared to Figure 2(a), we can see that Figure 2(b) creates smoke effects using gradient color and temperature.



(a) (b)

Fig. 2. Simulations of various chemical reaction of fire based on different gradient color. Here we can see that different colors are assigned to the core of the flame.

4. Stable Fluid Simulation on Billboards

A 3D volume rendering is widely used for realistic fire animation, but it requires much time and many resources. Therefore, the technique is not adequate for interactive games, including mobile games. Also, our technique is simulated on mobile platforms, which cannot support high resolution LCD screens. Currently, most mobile phones use 320x240 resolution LCD screens. One of the major constraints of a mobile handset is its small display device. But from the viewpoint of fluid simulation, fluid solvers do not need many grids. Hence we can save many system resources.
Two dimensional animated texture maps have been used to create the effect of the upward movement of burning gas, but such models are effective only when viewed from a specific direction. Orienting the polygon based on the view direction is called billboarding, and the polygon is called a billboard(Reeves 1983;Parent 2002)
As the view changes, the orientation of the polygon changes. Our billboard technique is combined with alpha texturing and fire animation. Figure 3 shows a billboard tree modelling on our Mobile 3D Environment.

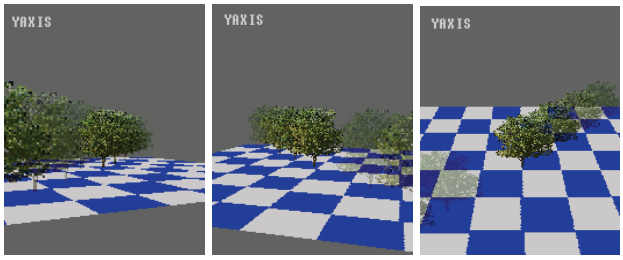


Fig. 3. Billboard Tree modeling on Mobile Environment

Figure 4 shows the fluid animation of our billboard technique in a mobile environment. The fluid is simulated on 64x64 grid billboard. The fluids are simulated on offscreen memory and the simulated images are mapped on a screen billboard.
The red color shows the velocity field and the black and white color shows the density field. As you can see, the velocity field and density fields are changed by external forces. The external forces are added by user interactions. Also, we can see that as the view changes, the orientation of the polygon changes.

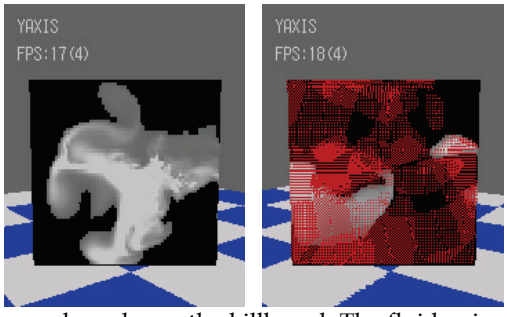
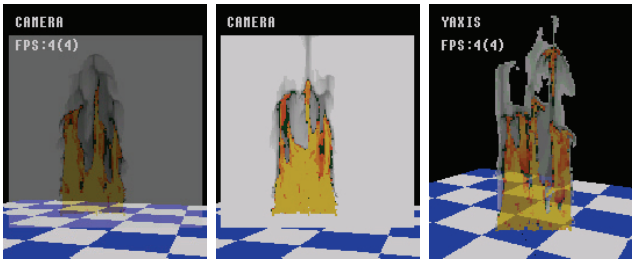


Fig. 4. Simulation of fire and smoke on the billboard. The fluid animation is simulated on a 64x64 grid billboard.

5. Alpha Blending with Background

In image based modeling, including in the billboard animation area, alpha blending is widely used for realistic looking effects with billboards and backgrounds. We used NF3D's sprite attributes for alpha blending.
Figure 5 shows the alpha blending effects of fire. Figure 5(a) is an alpha blended fire (alpha with 0.5), while Figure 5(b) does not have an alpha blended billboard (alpha with 0.0). If we apply the same alpha value on the billboard, we cannot get realistic-looking fire (see Fig 5(a)). Therefore, we can specify different alpha values to specific billboard colors (Fig 5(c)). In fire simulation, the core part of the flame emits strong lights and it has a small alpha value, while outer part the flame has a large alpha value.
We can get realistic-looking fire from billboarding and alpha blending.

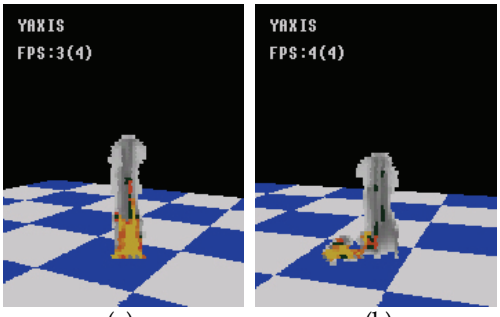


(a) (b) (c)

Fig. 5. (a) Alpha blended billboard with fire simulation, while (b) does not have an alpha blended fire with background. In figure (c), the original white background colors are erased by alpha blending.

6. Realtime Interaction with Game Characters

In computer games, interaction with characters in realtime is a very important issue. If fire and smoke animations are implemented without user interaction, they will look fake. Figure 6 shows fire animation under convection and buoyancy force. Notice that the fire's flames are moving only in an upward direction in Figure 6(a). Figure 6(b) shows the fire animation when additional external forces(like wind) are applied. In Fig 5(b), we applied external force on its left side.



(a) (b)

Fig. 6. (a) Fire animation under convection and buoyancy forces. (b) Fire animation with additional external force on its left side.

Figure 7 shows some snapshots of our mobile game "Rupee Story." As the game character approaches the fire, additional forces are applied on the fire. Therefore, it is influenced by the game character. The technique makes realistic looking fire animation in a mobile environment.

Fig. 7. The interactions are applied on our mobile 3D game "Rupee Story." As the game character is approaching the fire, the fire is influenced by the game character.

7. Performances

When simulating a fluid, the number of grid cell and the simulation time have a big trade-off. As can be seen from the results in Figure 8, the frame rate rapidly deteriorates when the simulation grid size increases.

Although a larger grid size would create a more detailed simulation, grid sizes larger than 96x96 are not adequate for interactive games on our 1024 Kb heap memory emulator. As you can see in Figure 8, less than 14 FPS animation is not a feasible frame rate for games.

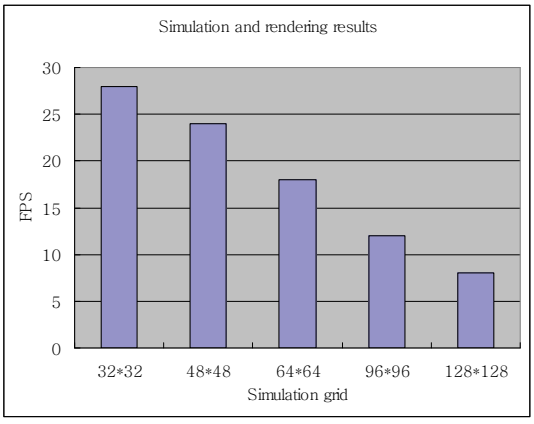


Fig. 8. Frame rates for a different implementation (on a 1024 Kb heap size).

8. Conclusions and Future Works

Physics based fluid simulations are very important but are difficult to implement for mobile platforms. We have presented a plausible physically based model for animating and

rendering fire and smoke on a mobile platform. Our fire and smoke animation was also implemented on the mobile 3D game, "Rupee Story." We demonstrated that this model could be used to produce realistic looking fire and smoke animation and plausible realtime interaction for mobile games.

Currently, Korean mobile standard forums are trying to create a mobile 3D standard API(Application Programming Interface). We are also attempting to develop more realistic fluid effects on new mobile 3D platforms.

We are also trying to implement water and ocean effects on mobile platforms, which are more difficult to render and must be implemented with level-set methods. The level-set methods are very effective methods for tracking fluid surfaces, but they require many system resources.

Also, we must implement more controls for fluid simulation, including defining the fire skeleton, and detaching flames.

9. References

Paramount: Star Trek II: The Wrath of Khan(film), June (1982)

Parent, R: *Computer Animation algorithms and technique*, Morgan Kaufmann, San Francisco (2002)

Stam. J.: Stable Fluids, *Proceedings of SIGGRAPH 99*, Los Angeles, (1999) 121-128

Harris, M. J.: Real-Time Cloud Simulation and Rendering, Ph. D Dissertation, University of North Carolina at Chapel Hill, (2003)

Reeves, W. T.: Particle Systems. A Technique for Modeling a Class of Fuzzy Objects, *Proceedings of SIGGRAPH 83*, (1983) 359-376

Shinya, M. and Fournier, M.: Stochastic Motion – Motion Under the Influence of Wind, *Proceedings of Eurographics 92*, (1992) 119-128.

Fedkiw, R., Stam, J., Jensen, H.: Visual Simulation of Smoke, *Proceedings of SIGGRAPH 2001*, (2001) 23-30.

Foster, N. and Fedkiw, R.: Practical Animation of Liquids, *Proceedings of SIGGRAPH 2001*, (2001) 15-22.

Selle, S., Rasmussen, N., Fedkiw, R.: A Vortex Particle Method for Smoke, Water and Explosions, *Proceedings of SIGGRAPH 2005*, (2005) 910-914.

Nguyen, D. Q., Fedkiw, R., Jensen, H. W.: Physically Based Modeling and Animation of Fire, *International Conf. on Computer Graphics and Interactive Techniques*, (2002) 721-728.

Melek, Z., Keyser, J.: Interactive Simulation of Fire, Technical Report 2002-7-1, Texas A&M University, (2002)

Carlson, M., Mucha P. J., Turk. G.,: Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid, *Procedings of SIGGRAPH 2004*, (2004) 377-384.

Stam, J.:Interacting with Smoke and Fire in Real-time, *Communications of the ACM*, Vol. 43, Issue 7, (2000) 76-83.

Stam, J.: A Simple Fluid Solver based on the FFT, *Journal of Graphics Tools*, Vol. 6, Number 2, (2002) 43-52.

Digital Camera Work for Soccer Video Production with Event Detection and Accurate Ball Tracking by Switching Search Method

Yasuo Ariki and Tetsuya Takiguchi
Kobe University
Japan

1. Introduction

Due to the increasing number of special channels, digital broadcasting requires a tremendous amount of video content and work associated with new interactive services. Sports video production for small spectators is a key issue associated with this problem, because such videos have not been produced to date due to the cost of producing such videos. From this viewpoint, an efficient and automatic low-cost production system for sports video content is needed even if it sacrifices a degree of professionalism.

When we watch a sports game on TV, the camera work helps us to understand the game progress owing to the panning and zooming carried out by the cameraman. This means that the camera work is strongly associated with the game events and the most suitable camera work is selected according to the events. Through camera work based on event recognition, more interesting and intelligible video content can be produced. Camera work can be classified into real camera work and virtual camera work. In real camera work, event recognition has to be carried out in real time. At present, it is difficult to do this, so virtual camera work is now the best way to produce dynamic video content with panning and zooming. This virtual camera work is sometimes called ``digital shooting,'' which is a composite of digital camera work and digital switching techniques.

We have been developing an automatic production system for commentary soccer video using digital shooting techniques based on the event (situation) recognition (Ariki et al., 2006). ``Commentary'' means intelligible and the system can produce comments on the game's progress or events because the events are recognized and digital shooting is carried out based on the events. The system is composed of image recognition techniques to track the soccer ball, event recognition, and finally digital camera work using panning and zooming. Event recognition is the key issue for digital camera work as well as for retrieving the events and summarizing the whole soccer game. However, event recognition mainly depends on the ball tracking accuracy, because events such as free kicks, goal kicks, throw ins, corner kicks and penalty kicks are strongly related to the ball. So far, ball tracking has been a difficult task, so complete event recognition and automatic video production have not been possible.

In this chapter, we describe a new technique to track the ball accurately on the Hi-vision video. Since the Hi-vision camera was located far from the soccer field in order to film half the field, the ball looks too small for tracking. At first, in the first frame, the ball is searched for using a global search with normalized cross-correlation. Although it is possible to search for the ball using global searching for all frames, the ball is lost sooner or later because of its small size and because of occlusion by the players. To solve this problem, we employed a local search with a particle filter, which can continue tracking around the area where the ball is ``lost.'' When the local search fails due to the disappearance of the ball (when the ball passes on and along the white line), the real ball is located far from the particles. In this case, a global search replaces the local search. After detecting the ball again, the local search continues tracking the ball. This switching of the search strategies is automatically carried out depending on the situation in which the ball is lost.
The organization of this chapter is as follows. In Section 2, the related works are described and in Section 3, the overview of the automatic production system is presented. A new ball tracking method is presented in Section 4. The camera work and the situation recognition are described in Section 5 and 6. In Section 7, the experiments of ball tracking and subjective evaluation of soccer video production with AHP are described.

2. Related Works

There are many aspects to a TV program sports video content production system, including generating highlights (Yow et al, 1995) (Assfalg et al, 2003), creating a summary (Ekin et al, 2003) (Nguyen et al, 2003), reconstruction (Bebie & Bieri, 1998), creating a mosaic (Kim & Hong, 2000) and other elemental technologies (Utsumi, et al, 2002). However, these produced videos are greatly subjected to limitations related to the video production staff (cameraman, editor, switcher, etc.). Also, the content inherits any mistakes in camera work and switching. Therefore, such video content has absolutely no degree of freedom (for example, adjusting the content to fit the preferences of the viewer).
Digital shooting has degrees of freedom higher than secondary usage of sports TV program content, because it is able to generate varied camera work and switching from the material video content, obtained by filming the whole soccer field using a HD (High-Definition) camera. One related topic of research in the field of automatic video shooting systems that we call a ``one-source multi-production system'' is Virtual Soccer Stadium (Koyama et al, 2003), which makes it possible to show viewers a free view of the soccer field in 3D. Using this system, we can watch the game from any viewpoint, but the camera work or shooting techniques used in TV broadcasts can help the viewer to understand the game's progress more clearly.
The camera work such as panning and zooming are frequently utilized in classroom lectures, and content production systems or content summarization systems are proposed, employing the detection of lecturer-movement and voice-detection algorithms (Yokoi & Fujiyoshi, 2005) (Yokoi & Fujiyoshi, 2006). However, the event detection needed to switch between various camera works in a soccer sports video production system is much more difficult.
Event recognition depends mainly on the ball-tracking accuracy, because events such as free kicks, goal kicks, throw ins, corner kicks, and penalty kicks are strongly related to the ball. There are two procedures in soccer ball tracking. One is ball detection and the other is ball

tracking. In paper (Huang et al, 2005), the ball candidates were first detected from several consecutive frames of broadcast TV video using color, shape and size information. Then a weighted graph was constructed with each node representing a candidate and each edge linking two candidates in adjacent frames. Then a Viterbi algorithm was carried out to extract the optimal path as the ball's locations. After ball was detected, it was tracked using Kalman filter-based template matching.
In paper (Tian et al, 2003), the ball candidates were also detected in each frame of the broadcast TV video by removing non-ball objects using color, shape and size information. After obtaining the ball candidates in all the frames, the Kalman filter was applied from the first frame to estimate the ball position on the succeeding frames. If the candidate was found in the next frame near the estimated point, it was used to update the Kalman filter. Otherwise (if no candidates were found), the estimated position was considered to be the ball position.
In paper (Liu et al, 2004), the soccer field was first extracted from the broadcast soccer video. Then several simple shape features and spatial context color were evaluated, and distinct non-ball regions were removed. Further examination determined the initial position of the ball in the first frame. The ball was then tracked using a CONDENSATION algorithm (particle filter) based on the similarity of histogram intersection of the object regions. This method hypothesizes the ball's occurrence on the game field and fails under some special conditions, such as occlusion of the ball by players or white field lines, too complex of a background, or the small size of the ball due to its distance from the camera. Although our approach also employs a particle filter, it is used for the local search only and switches to a global search when the ball is lost. This means that it is not necessary to hypothesize the ball's occurrence on the field.

3. Overview of the System

3.1 Digital Shooting

Digital shooting can be assumed to be an emulation of a virtual multiple-camera system that works by clipping a frame from HD material video content and mapping it roughly to frame at low resolution, such as SD (Standard Definition).
The digital shooting technique is composed of digital camera work and digital switching technique. The digital camera work is defined as virtual panning and virtual zooming. The virtual panning is a video production technique involving clipping a size-fixed frame by controlling frame location on HD video. The virtual zooming is a video production technique involving clipping a frame by controlling frame size. Also, digital switching is defined as the change of some virtual camera by controlling rapid change of frame location or size on HD video.
Although the camera work or switching by human beings during live sports events cannot retake the scene, digital shooting is able to repeatedly produce various camera work and switching from the recorded video material, which involved the filming of the entire soccer field with a HD camera.
In our experiments, we used a Victor GR-HD1 Hi-vision camera. Figure 1 shows half the field taken by Hi-vision camera and a SD image clipped from the fixed HD image. Digital camera work is produced by changing the coordinates of the clipping window on every HD

frame. For example, Figure 2 and Figure 3 show digital panning down from right to left down and digital zooming in, respectively.

Fig. 1. Clipping from the HD image to a SD image using digital camera work

Fig. 2. Panning by digital camera work

Fig. 3. Zooming by digital camera work

3.2 Processing Flow

Figure 4 shows the processing flow of the digital camera work system. At first, the entire image sequence is captured by the fixed Hi-vision camera.

In the image processing module, the players and the ball are tracked, and their coordinates are extracted. The image sequence is captured by the fixed Hi-vision camera so that no camera work is included in the original video content. Therefore, the background subtraction method can be applied to the material video to extract the players and ball. Background subtraction is a simple but effective method to detect moving objects in video images.

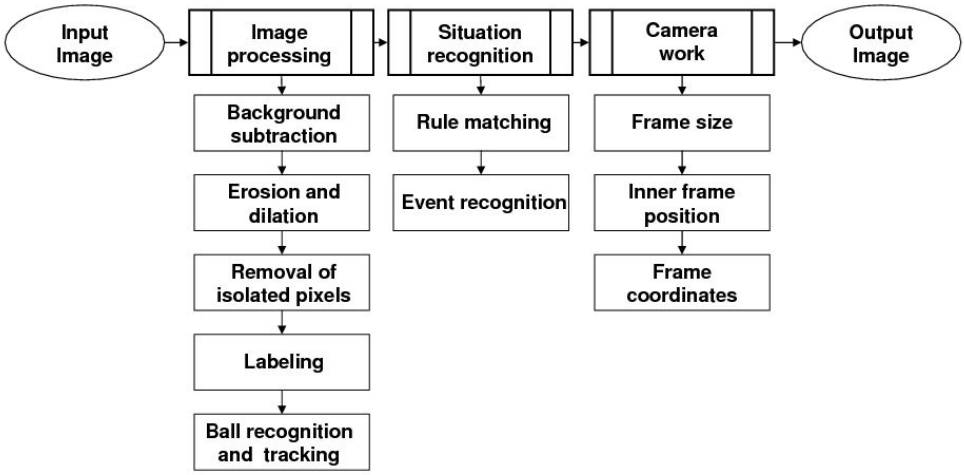


Fig. 4. Processing flow of the system

These background subtracted images do not depend on the image color. Hence, this method can be applied not only to the grass field but also to a dirt field. Furthermore, it is robust enough to handle the slow changes caused by sunshine or illumination if the background image to be subtracted is updated. The background image is updated by averaging the M frames at every N frames.

In the background subtraction process, each binary image is preprocessed by a morphological operator (erosion and dilation) to extract a player region, and noise reduction processing is applied. After region labeling, the ball is recognized and tracked. The players are extracted on every frame but not tracked or matched between consecutive frames.

In the situation recognition module, the events (such as throw ins, free kicks and goal kicks) are recognized based on the coordinates of the players and the ball, using the rules to recognize the event situation. In the camera work module, proper clipping size and frame coordinates are decided according to the recognized events.

The event recognition depends mainly on the ball tracking accuracy, because events (such as free kicks, goal kicks, throw ins, corner kicks and penalty kicks) are strongly related to the ball. So far, ball tracking was a difficult task so that complete event recognition and automatic video production were not successful.

In the following section, we propose an efficient and stable ball-tracking method by switching search methods back and forth between global search and local search. In the frames where the ball is lost as well as in the first frame, the global search with normalized cross-correlation is performed. Then the local search with particle filter continues the tracking. Once the system has recognized that the local search has failed by receiving a sequence of low probabilities among the ball-tracking results, the search method is switched to the global search.

4. Ball Tracking

In tracking a soccer ball on videos, there are several problems to be solved. The main problems are summarized as follows;

1. A ball is small and round, so sometimes it is confused with other similar objects due to its featureless properties.
2. In the case where the ball is kicked high and passes over the spectators, similar objects such as people's faces, caps or bags are confused with the ball.
3. The ball is sometimes occluded by players and confused with the players' white shoes.
4. In the case where the ball passes on the white line or white pole, it disappears for a short time.
5. The ball movement is sometimes irregular when it touches players.

To solve problems 1 and 2, background subtraction is effective because the stationary objects similar with the ball are removed. Then ball detection methods, such as normalized cross-correlation between the detected moving objects and a ball template, can be applied on the first frame. We call this the global search because the system does not know where the ball is on the first frame so that it searches the whole image.

This global search is time-consuming, so a more efficient search algorithm is required. One of the most plausible methods will be a local search, such as Kalman filter or particle filter. The Kalman tracker hypothesizes a unimodal Gaussian probability distribution function. On the other hand, the particle filter does not have such a hypothesis, so the particle filter is thought to be better than the Kalman filter. This ball search method is called a local search because the system predicts the ball's location area, and the ball is searched for within this area.

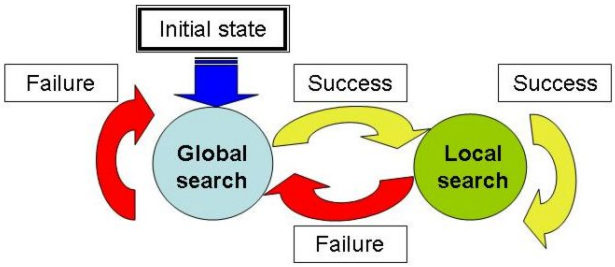


Fig. 5. Organization of a ball tracking system

A local search can solve problems 3, 4 and 5 to some degree, but it fails due to ball disappearance when the ball passes on and along the white line or a big white character

written on the wall in front of the spectators. In this case, after a few seconds, the global search should be applied to detect the ball position again.
Figure 5 shows the organization of the proposed ball-tracking system using global search and local search. In the frames where the ball is lost as well as in the first frame, the global search with normalized cross-correlation is performed. Then the local search with a particle filter continues the tracking. Once the system has recognized that the local search has failed by receiving a sequence of low probabilities of the ball tracking results, the search method is switched to the global search.

4.1 Global Search

In the global search for the ball, the normalized cross-correlation $R(x,y)$ is computed between the ball template $T(i,j)$ and the small region within the search area I based on the following equation:

$$R(x,y)=\frac{\sum_{i,j}\left\{T(i,j)-\overline{T}\right\}\cdot\left\{I(x+i,y+j)-\overline{I}\right\}}{\sqrt{\sum_{i,j}\left\{T(i,j)-\overline{T}\right\}^2\cdot\sum_{i,j}\left\{I(x+i,y+j)-\overline{I}\right\}^2}} \tag{1}$$

$$(\hat{x},\hat{y})=\arg\max_{x,y}R(x,y) \tag{2}$$

where $\overline{T}$ and $\overline{I}$ are the averaged intensity of the ball template image and the underlying small region, respectively. $(\hat{x},\hat{y})$ is the point where the ball is matched best within the search area I. In this study, the search area I is set to a whole image in the video. The ball template is prepared manually in advance.

4.2 Local Search with Particle Filter

4.2.1 Particle Filter

In the local search for the ball, a particle filter (Isard & Blake, 1998) is employed. The ball is tracked by estimating *a posterior* probability $p(x_t \mid Z_t)$ of the ball state x_t at time t after observing the image feature sequence $Z_t=(z_1,\ldots,z_t)$ up to current time t.
This *posterior* probability is computed based on Bayesian theory as follows, using *a prior* probability $p(x_t \mid Z_{t-1})$ and likelihood $p(z_t \mid x_t)$ of the image feature z_t at the state x_t.

$$p(x_t \mid Z_t)=k_t\,p(z_t \mid x_t)\,p(x_t \mid Z_{t-1}) \tag{3}$$

If the *posterior* probability $p(x_t \mid Z_t)$ is complicated, it can be computed by randomly sampling the states x_t as particles $s_t^{(n)}$ (n=1,…,N) according to the *prior* probability $p(x_t \mid Z_{t-1})$ and estimating the likelihood $p(z_t \mid x_t)$ of the image feature z_t at the state x_t as $\pi_t^{(n)}$. It is expressed as follows:

$$p(x_t \mid Z_t) \approx \sum_{n=1}^{N} \pi_t^{(n)} \delta(s_t^{(n)}) \tag{4}$$

Here, δ indicates the delta function.
The *prior* probability $p(x_t \mid Z_{t-1})$ is computed by transferring the *posterior* probability $p(x_{t-1} \mid Z_{t-1})$ at previous time *t-1*, using the state transition probability $p(x_t \mid x_{t-1})$ as follows:

$$p(x_t \mid Z_{t-1}) = \int_{x_{t-1}} p(x_t \mid x_{t-1}) p(x_{t-1} \mid Z_{t-1}) dx_{t-1} \tag{5}$$

In this study, we employ motion dynamics.

4.2.2 Motion Dynamics

The motion dynamics used to move the particles from the *posterior* probability at time *t-1* to the *prior* probability at time *t* is expressed by the following equation of motion:

$$x_t = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix} x_{t-1} + \omega \tag{6}$$

where ω is the Gaussian noise term, and x_t is the state of the particle and expressed as follows:

$$x_t = [p_{tx}, p_{ty}, v_{tx}, v_{ty}]^T \tag{7}$$

Here, p_{tx}, p_{ty}, v_{tx} and v_{ty} are the positions of x and y and the velocities in the x and y directions in the image at time t, respectively. The velocity is estimated using the past tracked positions.

4.2.3 Likelihood Estimation

The likelihood of the ball is estimated at each particle by the normalized cross-correlation between the image feature at the position of the particle and the ball template, expressed in Eq. (1). When the normalized cross-correlation is applied directly, many particles are confused with non-ball objects such as spectator faces, caps and bags. To avoid such a situation, moving objects are only extracted and tracked using the particle filter. Moving objects are extracted by subtracting the background image from the present image.

5. Camera Work Module

In the camera work module, the digital panning and zooming are controlled. Digital panning is performed on the HD image by moving the coordinates of the clipping window, and digital zooming is performed by changing the size of the clipping window.

5.1 Zooming and Clipping Size

After analyzing the professional camera work on a TV soccer game, it was found that during soccer games, three sizes of the clipping window are used in zooming in: tight shot, middle shot and loose shot size. These sizes of the clipping window are selected according to the game situation. For example, the tight shot is selected for the play near the goal when the ball movement is slight. If the tight shot is used frequently, the video becomes not intelligible so the tight shot is used only for important play with duration more than two seconds.

The loose shot or middle shot camera work is selected for normal play or situations, such as free kicks, where the ball is expected to move fast. Transition from the loose shot to middle shot or vice versa is performed according to the game situation. The transition is continuously done within 0.5 seconds. If the duration of the loose shot or middle shot after the transition is less than 0.5 seconds, then a transition does not take place.

Based on these parameters, in our experiment, three clipping sizes are prepared as shown in Table 1. Figure 6 shows examples of these shot sizes on the HD image. They are continuously or abruptly switched back and forth between each other according to the game situation, as shown in Figure 7.

Tight shot size	Middle shot size	Loose shot size
120 × 90	240 × 180	480 × 360

Table 1. Three sizes of the clipping window on the HD image (pixels)

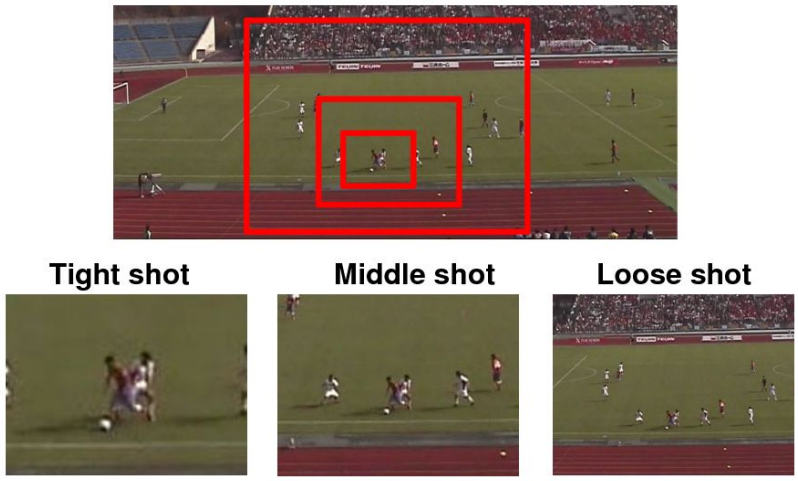


Fig. 6. Example of clipping window sizes

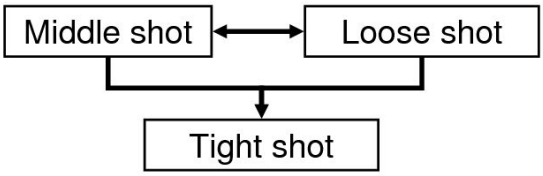


Fig. 7. Size switching of clipping window

5.2 Panning and Clipping Coordinates

The ball location is important because the soccer game progresses following the ball's location. However, ball trajectory cannot adequately be used directly to trigger a commencement of panning because smooth panning cannot be obtained due to the erratic movement of the ball.

In other words, the panning operation has to satisfy two contrary conditions, namely that the clipping window (frame) must follow the ball quickly if the ball moves fast, but if the ball moves erratically, the frame must not follow the ball.

To meet these two conditions, we set an inner frame within the clipping window as shown by the black frame in Figure 8. Even if the ball is moving erratically, the clipping window remains still if the ball exists inside the inner frame. If the ball moves out of the inner frame, the centroid of the clipping window moves toward the ball location as shown in Figure 9.

Fig. 8. Clipping window (white) and inner frame (black)

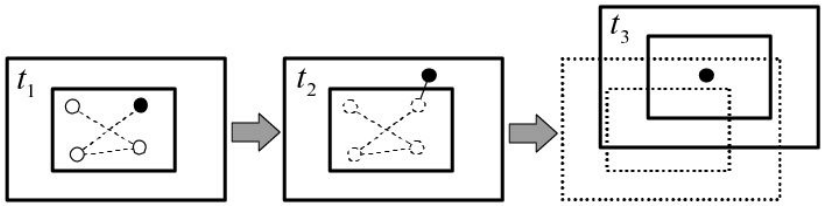


Fig. 9. Control method of clipping window

5.3 Inner Frame Position

The inner frame usually locates itself at the center of the clipping window. However, if the play is something like a free kick, long pass or dribbling etc., then the forward direction should be included in the clipping window to make the play formation or players remain visible.

Therefore, it is necessary to set the inner frame at the position opposite the center of the clipping window toward the ball movement direction. Figure 10 shows the inner frame in such a situation. The inner frame is shown as the small rectangle and the clipping window is shown as the large rectangle. Usually the left clipping window and the inner frame are employed. However, in the situation of a free kick by the goalkeeper, the right clipping window is selected to set the inner frame at the left position (opposite to the ball direction).

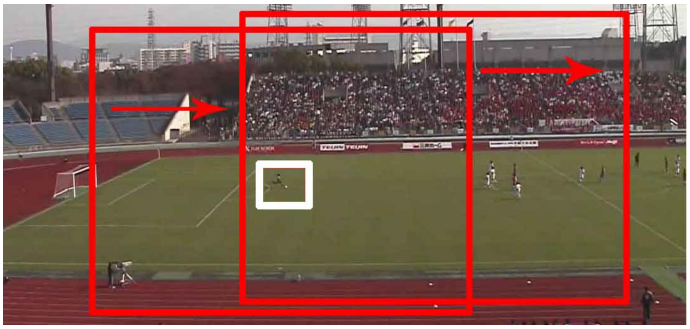


Fig. 10. Inner frame position control

6. Situation Recognition Module

In the situation recognition module, the game situation is recognized based on the ball and players extracted in the image-processing module and this game situation are forwarded to the camera work module. The game situations for controlling the camera work are classified into two groups. One changes the camera work among loose shots, middle shots and tight shots according to ball-player relationships. The other sets the typical camera work according to the events such as goal kicks, corner kicks and free kicks.
In order to clarify the game situation for changing the camera work, we analyzed the professional camera work of a soccer game broadcast on TV. From this analysis, rules were found for causing the camera work changes based on the game situation and the events. In this section, the rules for starting zooming and the rules for detecting events are described.

6.1 Zooming Rule

Rules for changing the camera work based on the game situation are described. The camera work change has already been described in Figure 7.

6.1.1 Rules for Switching from Loose Shot to Middle Shot

Changing the camera work from the loose shot to the middle shot occurs when a more detailed game situation is required in order to understand play. Through analyzing TV soccer games, two types of such situations are found. One is the case where the ball approaches the goal and the other is the case where the players crowd around the ball.
One of the most important scenes in a soccer game is a goal scene so that the camera must be zooming in at this time. When the ball passes over the penalty line toward the goal, a goal

scene may be expected, so the system changes the camera work to the middle shot from the loose shot. The situation is recognized by computing the horizontal coordinate of the ball.
When a lot of players are around the ball, it is effective to zoom in to see clearly. This situation is recognized by counting the number of players around the ball on middle shot resolution. If the number is over six or seven players, then the situation is interpreted as crowded.
The accuracy of the player detection is not so high due to the overlap of the players, however, so the crowded situation is determined by counting the number of players including the player overlap.

6.1.2 Rules for Switching from Middle Shot to Loose Shot

Changing the camera work from middle shot to loose shot occurs when the camera field does not include the ball or the minimum number of players. Through analyzing broadcast soccer games on TV, two such situations are found. One is the case where the ball moves too fast and the other is where the players scatter very fast.
A fast-moving ball situation is detected by computing the displacement of the ball coordinates at every frame. If it is over the threshold (six pixels), the fast movement of the ball is detected. This rule may be applied even when the ball moves forward and backward around the same place. In this case the camera work may change frequently. To avoid this situation, the coordinate of the ball is checked and if it is increasing or decreasing in a constant direction for more than two seconds, the rule is applied.
When players scatter, the camera work changes from the middle shot to loose shot to catch the ball and the minimum number of players. This situation is recognized by counting the number of players around the ball.
There are some conflicts between rules for changing camera work from the middle shot to loose shot and vice versa. One conflict is when a player kicks the ball toward the goal; i.e. when the ball moves very fast near the goal. In this case, two rules may be applicable. One is the rule to change the camera work from loose shot to middle shot because the ball is located near the goal as described in Section 6.1.1. The other rule is to change the camera work from middle shot to loose shot because of the very fast ball movement. In this case, the rule to change the camera work from the middle shot to the loose shot is inhibited.
The other conflict is when the ball moves very fast over a crowd of players. In this case, two rules may be applicable. One is the rule to change the camera work from middle shot to loose shot because the ball moves very fast. The other is the rule to change the camera work form loose shot to middle shot because the crowded players are detected. In this case, the rule to change the camera work from the loose shot to middle shot is inhibited.

6.1.3 Rules for Tight Shots

In the professional camera work on TV soccer games, tight shots, such as famous players or players scrambling for the ball and dribbling, are sometimes inserted in the videos. These tight shots make the game interesting. The proposed system does not recognize players' faces at present, so the scrambling situation is only zoomed in on as a tight shot in our system.

This situation is recognized when the ball movement is below some threshold and there is at least one player around the ball. The threshold is set to be 1.5 pixels through the preliminary experiment.

6.2 Event Detection Rule

The proposed system can recognize five events; free kicks, goal kicks, throw ins, corner kicks and penalty kicks. These events are detected using a feature that notices when the ball remains still for some duration under the tight shot camera work. The detection duration for such events is set at 6 seconds.

Once the events are detected, the system zooms out to view the entire situation and recognizes the event types according to rules using the ball position and the distance between the ball and the players' average position as shown in Table 2. For example, if the ball is on the corner arc and the distance between the ball and the players' average position is medium distance, the event is recognized as a corner kick.

After the event recognition, the clipping size is determined according to the rules and the center of the clipping window is set to the players' average position.

Events	Ball position	Distance between ball and players	Clipping size
Free kick	Field area	Far	LS or MS
Goal kick	Goal area	Far	LS
Throw in	Out of bounds	Middle	MS
Corner kick	Corner spot	Middle	MS
Penalty kick	Penalty spot	Middle	TS

Table 2. Event recognition rules

7. Evaluation Experiment

7.1 Ball Tracking Accuracy

We selected consecutive 800 consecutive frames (26.6 sec) from a soccer game that was played during the 38th National High School Soccer Championship (Kyoto area final) in Japan. The size of the image was 1,280 x 720 pixels with 24-bit color. The background image was produced in advance by averaging the randomly selected images in the video. The ball template image size was 15 x 15 pixels and the particle image size was 30 x 30 pixels, which is larger than the ball template image size, in order to search the best position within the particle image. The number of particles re-sampled was 100.

Three tracking methods (global search, local search and switching search) were evaluated. The results are shown in Table 3. In the table, ``Tracking rate'' is the ratio of the number of correctly tracked frames to the number of total frames (800). ``Restart'' indicates the necessity of manual restart when the tracking failed. The average ``processing time'' per image was calculated on a computer with a Xeon 3.06GHz processor and 1,024 MB of memory.

In Table 3, the local search achieved a very high tracking rate with low processing time owing to the particle filter. Two failures occurred when the ball passed on and along the

white line and it passed along the big white character written on the wall in front of the spectators. In these cases, a manual restart was required by specifying the true ball position. On the other hand, in the proposed method, the search was switched to the local search after the global search obtained a high cross-correlation value, and the search was switched to the global search when 40 frames continued with the low probability of the ball tracking results. The threshold of high cross-correlation and the low probability were empirically determined. It showed that the incorrect tracking caused by the particle filter was restored by the proposed switching search and the processing time was almost same as that of the particle filter.

Method	Tracking rate [%]	Restart	Processing time [sec]
Global search	61.1	Not required	3.2
Local search	91.5	Required	0.8
Switched search	91.8	Not required	0.9

Table 3. Experimental results

7.2 Subjective Evaluation by AHP

The final goal of this study is to construct a soccer video production system that meets viewer preferences and contains announcer commentary. Therefore, AHP (Analytic Hierarchy Process) (Saaty, 1997) can be used for the evaluation of the produced video contents because of its ability to represent human subjectivity.

AHP is a multi-criteria decision support method designed to select the best from a number of evaluated alternatives with respect to several criteria. It carries out pair-wise comparison judgments, which are used to decide the overall priorities for ranking the alternatives.

Three items were selected for the evaluation criteria of the video contents by AHP. They are naturalness, video quality and intelligibility. For naturalness, four camera work criteria are selected: zooming, panning, shot size, and shot duration. The video quality is adopted to judge whether or not the produced video content is inferior to TV or HD content. Also, the intelligibility of the game process is also adopted. These criteria are shown in Table 4.

Criteria	Evaluation
1. Zooming	Good <-> Poor
2. Panning	Good <-> Poor
3. Shot size	Good <-> Poor
4. Shot duration	Proper <-> Improper
5. Video quality	Fine <-> Coarse
6. Intelligibility	High <-> Low

Table 4. Evaluation criteria used in AHP

The video content to be evaluated by AHP are HD content, TV content, the produced content by our proposed method and the content produced by our conventional method with only panning from the original HD content (Ariki et al, 2004). All were produced from the same soccer game (38th National High School Soccer Championship).

The HD content was taken for a wide-angle half of the field by HD camera because of the limitation of camera resolution. The TV content was recorded by a video recorder when the

game was broadcast on TV. The reason why the HD content was compared in this chapter was to investigate which was fundamentally more comprehensible, TV content with camera work or wide-angle HD video content.

Figure 11 shows the AHP tree map used in this experiment. The middle row shows the six evaluation criteria. The bottom row shows the four materials to be compared.

The preference weights of AHP for each type of content are shown in Figure 12. The TV content showed high score followed by the contents of the proposed method, HD and the conventional method. The result shows that the presentation of game process is the most important and, therefore, the TV content obtained a high score. It also indicates that the TV content with panning camera work by professional cameramen is important compared with the HD content with no camera work.

The proposed method showed an AHP weight similar to the TV content and improved the intelligibility of the game process compared with the conventional method because of the camera work (panning and zooming based on the situation recognition). The content produced by the proposed method is still inferior to the TV contents since the digital camera work is not so high compared with the professional cameraman and the video quality is also lower than the TV and HD contents.

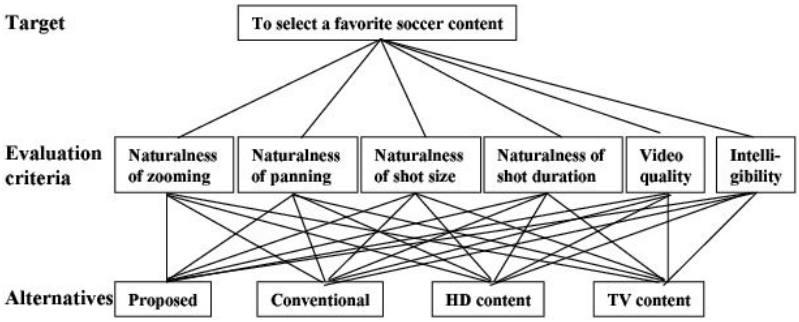


Fig. 11. AHP treemap

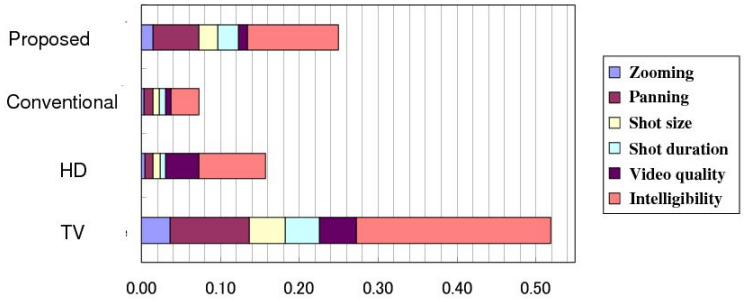


Fig. 12. Evaluation result by AHP

8. Conclusions

In this chapter, we proposed a system to track a small ball accurately on the fixed Hi-vision video by switching between global search and local search. This switching in search strategies is carried out automatically, depending on the situation when the ball is lost. Through the experiment, we showed that the proposed method can achieve continuous and stable ball tracking.

We also proposed a method to produce soccer video content with digital camera work based on situation recognition. AHP evaluation for our method showed a lower preference score than the TV content. However, the basic composition of the AHP scores is almost the same between the content produced by the proposed method and the TV content in terms of evaluation criteria, such as panning and zooming. This indicates that the improvement of our proposed method will lead to a degree of quality close to the technique of the TV cameramen in the future. Multiple cameras will be required to film the various shots that need to be taken from different angles rather than just the three shot types we tested. This problem will be also dealt with in the future.

9. References

Ariki, Y.; Kubota, S. & Kumano, M. (2006). Automatic Production System of Soccer Sports Video by Digital Camera Work Based on Situation Recognition, *Proceedings of Eight IEEE International Symposium on Multimedia*, pp. 851-858, 2006

Yow, D.; Yeo, B-L.; Yeung, M. & Liu, B. (1995). Analysis and Presentation of Soccer Highlights from Digital Video, *Proceedings of 2nd Asian Conference on Computer Vision*, pp. 499-503, 1995

Assfalg, J.; Bertini, M.; Colombo, C.; Bimbo, A. D. & Nunziati, W. (2003). Automatic Interpretation of Soccer Video for Highlights Extraction and Annotation, *Proceedings of ACM symposium on Applied computing*, pp. 769-773, 2003

Ekin, A.; Tekalp, A. M. & Mehrotra, R. (2003). Automatic soccer video analysis and summarization, *IEEE Trans. on Image Processing*, 12-7, pp. 796-807, 2003

Nguyen, N. T.; Thang, T. C.; Bae, T. M. & Ro, Y. M. (2003). Soccer Video Summarization System Based on Hidden Markov Model with Multiple MPEG-7 Descriptors, *Proceedings of the International Conference on Imaging Science, Systems and Technology*, pp. 673-678, 2003

Bebie, T. & Bieri, H. (1998). SoccerMan - Reconstructing Soccer Games from Video Sequences, *Proceedings of 2007 IEEE International Conference on Image Processing*, pp. 898-902, 1998

Kim, H. & Hong, K-S. (2000). Soccer Video Mosaicing Using Self-Calibration and Line Tracking, *Proceedings of International Conference on Pattern Recognition*, pp. 1592-1595, 2000

Utsumi, O.; Miura, K.; Ide, I.; Sakai, S. & Tanaka, H. (2002). An object detection method for describing soccer games from video, *Proceedings of 2002 IEEE International Conference on Multimedia and Expo*, pp. 45-48, 2002

Koyama, T.; Kitahara, I. & Ohta, Y. (2003). Live Mixed-Reality 3D Video in Soccer Stadium, *Proceedings of International Symposium on Mixed and Augmented Reality*, pp. 178-187, 2003

Yokoi, T. & Fujiyoshi, H. (2005). Virtual Camerawork for Generating Lecture Video from High Resolution Images, *Proceedings of 2005 IEEE International Conference on Multimedia and Expo*, pp. 45-48, 2002

Yokoi, T. & Fujiyoshi, H. (2006). Generating a Time Shrunk Lecture Video by Event Detection, *Proceedings of 2006 IEEE International Conference on Multimedia and Expo*, pp. 641-644, 2006

Huang, Q.; Liang, D.; Liu, Y. & Gao, W. (2005). A scheme for ball detection and tracking in broadcast soccer video, *Proceedings of PCM2005*, pp. 864-875, 2005

Tian, Q.; Yu, X.; Xu, C. & Leong, H. W. (2003). A ball tracking framework for broadcast soccer video, *Proceedings of 2003 IEEE International Conference on Multimedia and Expo*, pp. 273-276, 2003

Liu, Q-S.; Tong, X. F. & Lu, H. Q. (2004). An effective and fast soccer ball detection and tracking method, *Proceedings of ICPR*, pp. 795-798, 2004

Isard, M. & Blake, A. (1998). Condensation-conditional density propagation for visual tracking, *International Journal Computer Vision*, 29 (1), pp. 5-28, 1998

Saaty, T. (1997). A scaling method for priorities in hierarchical structures, *Journal of Mathematical Psychology*, 15, pp. 234-281, 1997

Ariki, Y.; Kumano, M. & Tsukada, K. (2004). A method of digital camera work focused on players and a ball, *Proceedings of 2004 Pacific-Rim Conference on Multimedia*, pp. 466-473, 2004

Testing and Evaluation System for Camera Shake and Image Stabilizers (TEVRAIS)

Kazuki Nishi
UEC Tokyo (University of Electro-Communications)
Japan

1. Introduction

Hand movement while taking pictures, that is called camera shake, causes undesirable image blur. A quicker shutter reduces the camera shake blur but degrades the signal-to-noise ratio of image due to insufficient exposure. The camera-shake blur is, therefore, unavoidable especially in darker situation that the shutter speed needs to be set in slower range. Increasing pixel count in recent image sensor is more susceptible to the camera shake because even a slight camera movement causes a large pixel displacement in the imaging area. Additionally the higher pixel density requires a longer exposure because the light-sensing areas in each pixel become smaller, and therefore makes the camera image more likely to be affected by the camera shake.
To prevent or reduce the camera shake, various image stabilizers (see Figure 1) have been developed in many camera manufacturers. As the traditional technology, the electronic image stabilizer (Uomori et al., 1990) has been investigated to suppress the camera shake of

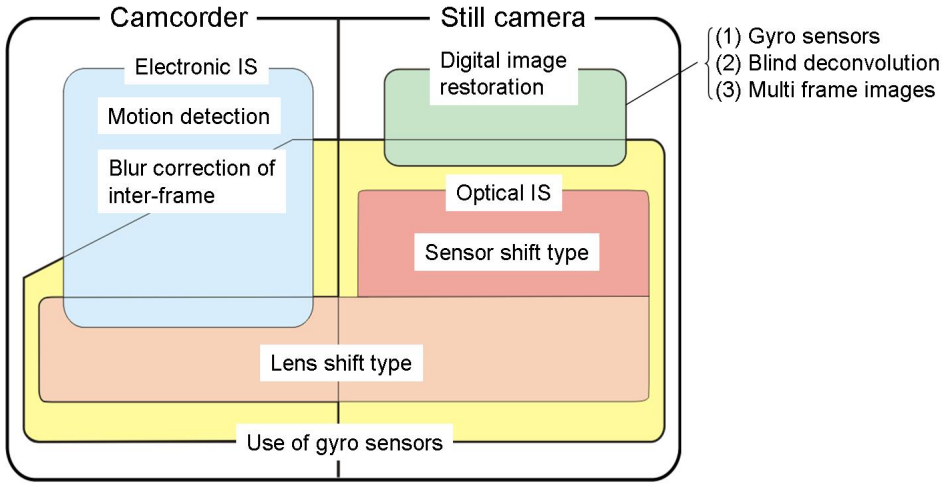


Fig. 1. Classification of image stabilizer technology

camcorders or video cameras. This technique shifts the frame image horizontally and/or vertically frame-by-frame to counteract the motion. The visible frame is limited to a narrower area inside of the maximum possible imaging area to provide a buffer for the motion. The digital image restoration technology has also been studied to recover the image blur due to camera shake of still cameras. This technique is based on the deconvolution filter that is derived from the point spread function (PSF) representing the camera motion. The PSF is estimated from the built-in gyro sensors or the camera image itself without any sensors (Fergus et al., 2006). Especially the latter type is called BLIND deconvolution. Recently, many serial images taken by the continuous shooting mode are used to realize the camera shake correction as well as the super-resolution to break out of the resolution limit (Park et al., 2003).

The most efficient technology that is employed in the most consumer products is the OPTICAL image stabilizer (Sato et al., 1993). The common architecture consists of built-in gyro sensors that identify the hand movement and the movement cancellation mechanism that precisely shifts the optical axis or image sensing device to compensate for camera motion. Since the camera motion is canceled mechanically and directly, the trail of camera shake blur does not remain in the camera image, therefore, the optical image stabilizer has superior performance. It is said that the optical image stabilizer allows us to take handheld shots at three or four-step slower shutter speed than without image stabilizers, but it is not known exactly how effective it works for cancellation of the camera shake. It is important to evaluate the performance qantitavely and objectively, because the performace will be improved if drawbacks of the present image stabilizer can be found through the evalution. This capter focuses on the measurement of camera shakes and the evalution of image stabilizers.

If the camera shake draws just a straight line, it can easily be detected by analyzing the PSF shape (Yitzhaky & Kopeika, 1998) or the spectrum distribution (Choi, et al., 1998) since it takes a simple 1D structure. Recently, the 2D trajectory corresponding to the camera motion in yaw and pitch axes, that can be estimated by shooting a point light source, have been investigated (Xiao et al., 2006). The 3D camera motion can be identified with the optical flow computed from the camera image of a stationary test pattern (Park et al., 2004), however, it is impossible to draw the trajectory in temporal sequence. The most popular method is to use the modulation transfer function (MTF) and examine how much high frequcey components of MTF are retrieved by the image stabilizer (Choi et al., 2008). Any methods has not, however, been successful detecting the 3D trajectory that corresponds actual camera motions and quantifying them.

Our proposed method makes it possible to estimate the exact 3D trajectory of camera shakes and evaluate the quantitative performance of image stabilizers (Nish & Ogino, 2007). The key idea consists of shooting the quickly changing test patterns that are displayed in the custom-made LED panel and calculating displacements between each pattern that is multiply recorded in the camera image while opening the shutter. The concept of the 3D camera shake detection and the performance evaluation results of some optical image stabilizers are presented.

2. How to detect camera shakes and analyze them

2.1 Point light source

To examine camera motions, the simplest method is to shoot a pint light source (Xiao et al., 2006). The trajectory of camera shake appears as a random curve on the camera image (see Figure 2). So we can observe directly the behavior of camera shake and the shrunk trajectory by the optical image stabilizer. However, when camera shakes are small especillay in cases of a short time exposure or after-stabilization, it is difficult to discriminate the feature and extract it numerically.

Our goal is to make it possible to detect detail of camera shakes in any case. For this purpose, not only the point light source but also any static test pattern that is commonly used for camera tests are no longer useful because they overlap densely with slight diplacements due to the camera shake and thus can not be discriminated each other.

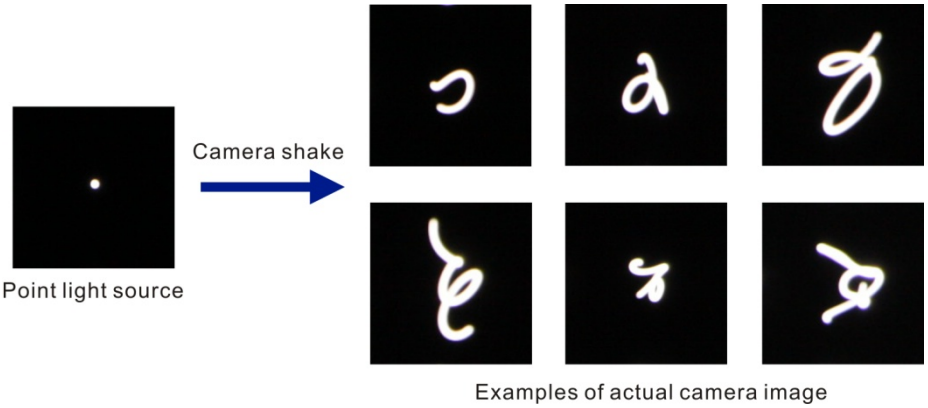


Fig. 2. Camera images of point light source

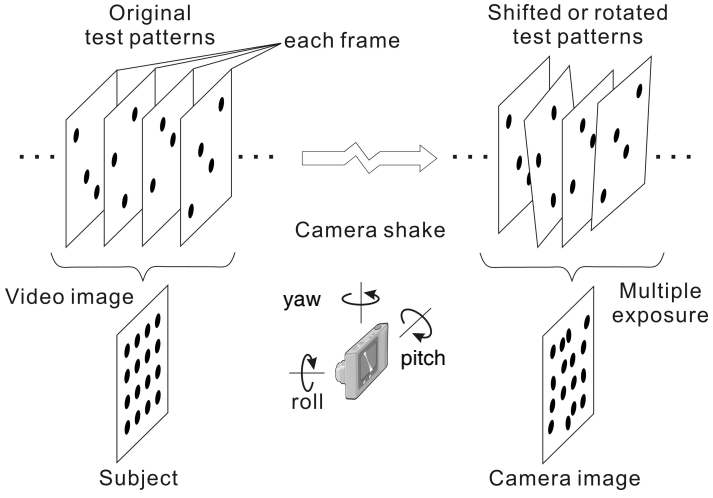


Fig. 3. Shooting of video test pattern with camera shake

2.2 Video test pattern

To realize precise detection of camera shakes, we propose using a VIDEO test pattern instead of static ones as the photographic subject. While the camera shutter is opening, each frame of the video test pattern is multiply recorded on an image with some shifts or rotations due to the camera shake. If those test patterns are chosen to eliminate overlap each other, each pattern can be easily discriminated from the multiply-exposed image. This is the key idea to detect camera shakes. Figure 3 shows the process from the shooting of video test pattern to the multiple exposure of them with some displacements or rotations due to the camera-shake. In this study, dot matrix patterns are used for the video test one.

For generating dot matirix patterns , we have developed a custum-designed LED panel (see Figure 4), that is comopsed of a window frame: 193mm $\times$ 193mm for the reference of pattern detection and 24 $\times$ 24 LED array with 1.5mmϕ lighting holes and 7mm spacing for dot matirx patterns. By displaying dot matrix patterns repeatedly with a horizontal or vertical shift, no-overlapping video test pattern is realized (see Figure 5).

Figure 6 is a camera image of the video test pattern. If camera shakes do not exist, it becomes the distortion-free lattice pattern, or it causes a deformation due to camera shakes.

The video test pattern can also be implemented in a high-response liquid-crystal display where each video frame corresponds to each test pattern, but the frame rate is limited to 60Hz.

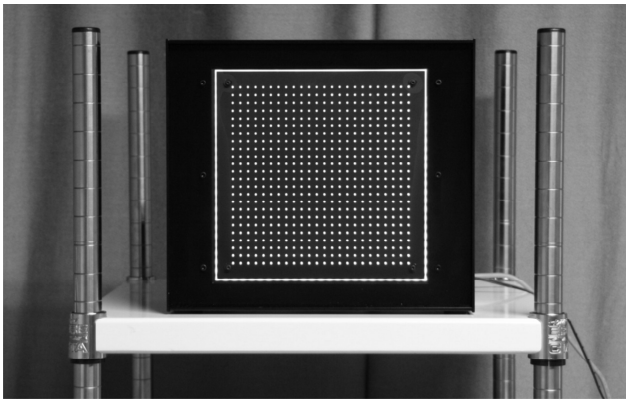


Fig. 4. Custum-made LED display for video test pattern

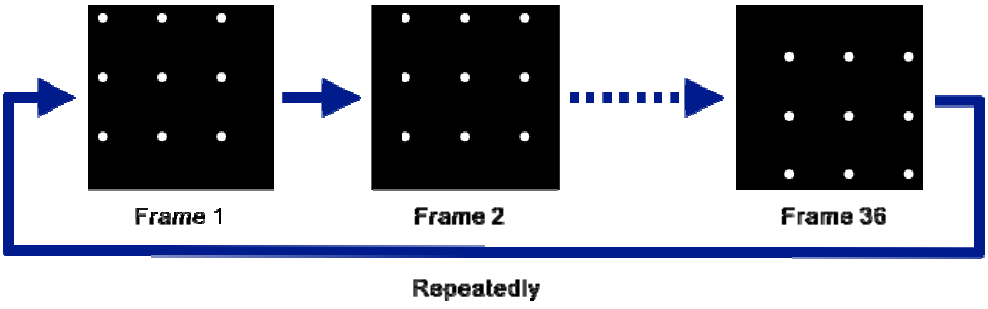


Fig. 5. Formation of video test pattern

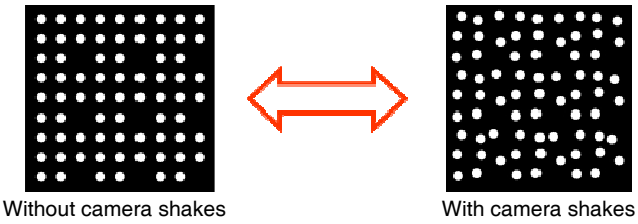


Fig. 6. Camera images of video test pattern

2.3 Computation of 3D trajectory based on pattern matching

The trajectory of camera shake can be obtained by computing displacements between each pattern in the multiply-exposed image. The pattern matching technique is useful for detecting positions of each test pattern from the image and computing displacements or roations between each pattern (see Figure 7). It is realized by using the shape-based pattern matching function of MVTec's machine vision tool HALCON. The shape-based pattern matching is advantageous to detecting in sub-pixel resolution and noise environment. Finally, the 3D trajectory of camera shake can be obtained by rearranging their estimates in time-sequence. Figure 8 shows examples of 2D trajectory (in yaw and pitch axes) computed through the above procedure.

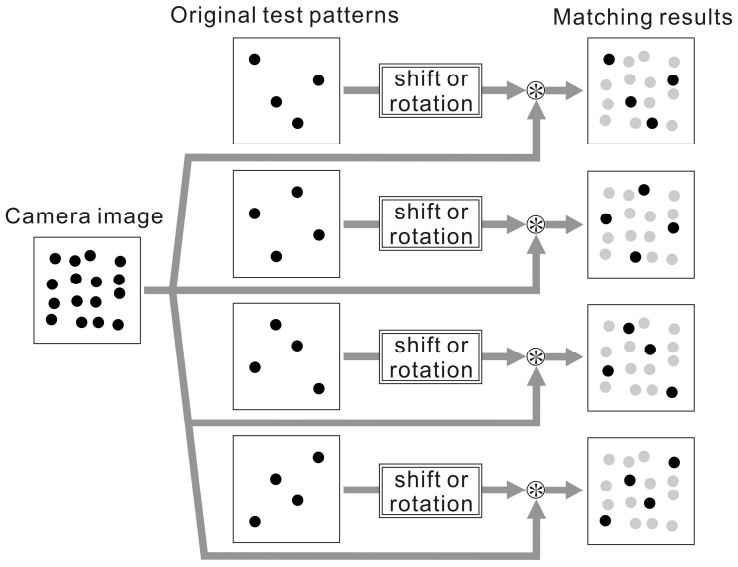


Fig. 7. Detection of displacements or rotations based on pattern matching

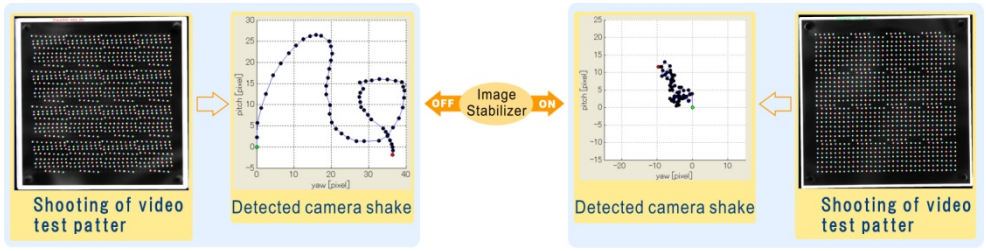


Fig. 8. Examples of detected camera shake trajectory with IS OFF and ON

2.4 Relation between camera motion and image plane

The actual camera shake involves random 3-axis shifts and 3-axis rotations, but the image movement is dominantly affected by rotations in 3-axis because parallel shifts make only a slight movement and are negligible in the camera image. When the camera rotates by $\theta_y, \theta_p, \theta_r$ in yaw, pitch and roll axes, respectively, an arbitrary point (x, y) of the image is moved to the following point (x', y'):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_y \\ d_p \end{bmatrix}, \tag{1}$$

$$\text{where} \quad d_y = L\tan\theta_y \approx L\theta_y \qquad d_p = L\tan\theta_p \approx L\theta_p. \tag{2}$$

d_y, d_p mean image shifts due to yaw and pitch rotations, respectively, where the camera-to-subject distance is L and both of them are assumed to be set to the same scale for simplicity. The above implies that if the displacement or rotation of the target image can be detected from the relation between (x, y) and (x', y') in the camera image, then the camera rotations in 3-axis $(\theta_y, \theta_p, \theta_r)$ can be computed directly (see Figure 9). The left of Figure 10 shows a 3D trajectory of camera shake, that is obtained through the above conversion from the displacements and rotation detected by the pattern matching.

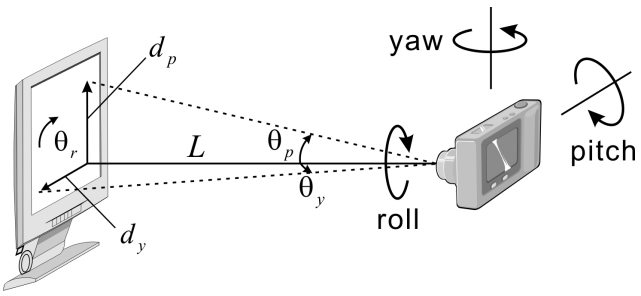


Fig. 9. Relation of camera-to-subject

2.5 Scatter diagram analysis

It is difficult to analyze the general trend of camera shake from just one of detected trajectories because it behaves randomly. The scatter diagram that is computed by accmulation of motion vectors between each frame interval (see the left side of Figure 10) from many images is useful for analyzing the camera shake trend. The quantification can be

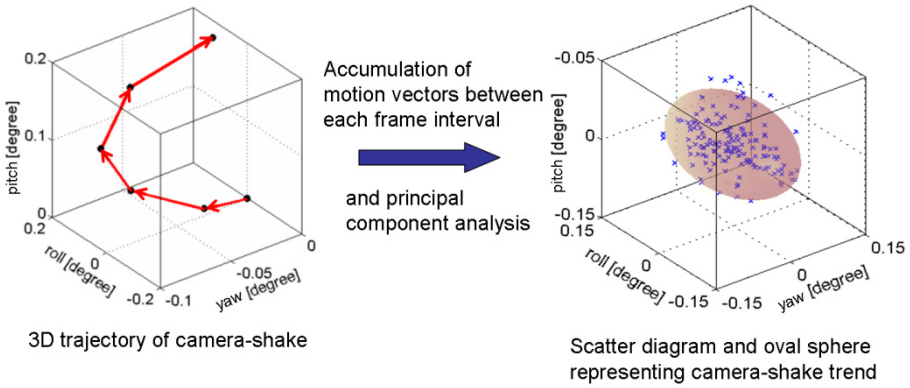


Fig. 10. 3D trajectory of camera shake and scatter diagram

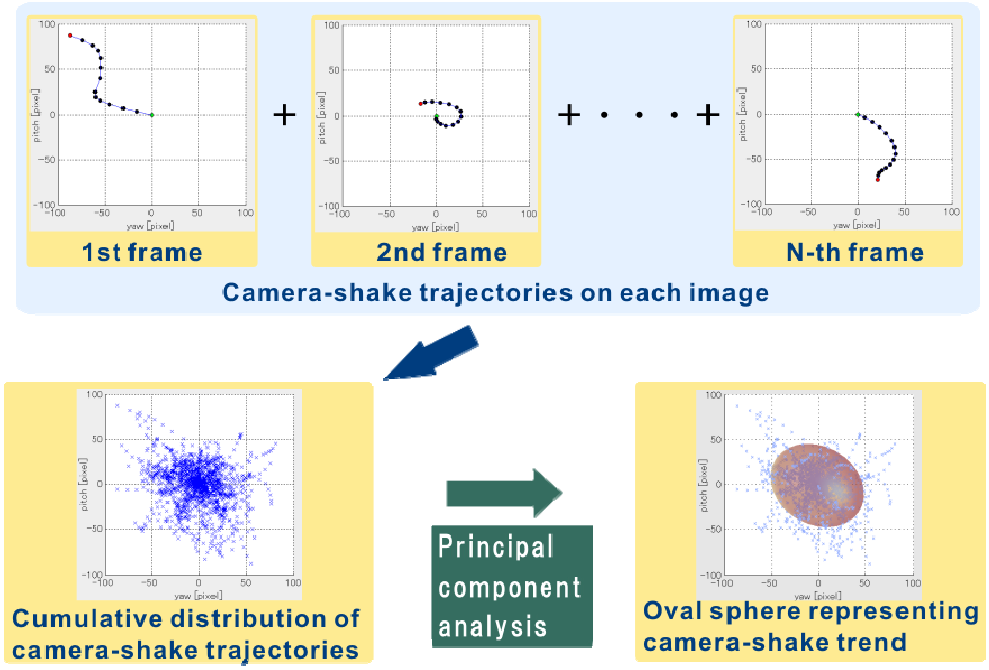


Fig. 11. Cumulative distribution of camera-shake trajectories

given by the principal component analysis, that means the average amplitude and orientation of camera shakes. The right side of Figure 10 shows an example of the scatter diagram obtained from many camera shake images. The oval sphere corresponds to the result of principal component analysis and shows the rough trend of camera shakes. So we can investigate differences of camera shake property in individuals or camera models through this analysis. When taking pictures with the image stabilizer, the distribution shrinks in origin. The degree of shrinkage shows the efficiency of image stabilizer.

Another way for quantification of the camera-shake trend is to compute the cumulative distribution of the detected trajectories, where all the start points of each trajectory are placed in origin, and apply the pricipal component analysis to that distribution (see Figure 11). The longer the shutter speed causes the larger the camera shake, therefore, more spread the distribution.

Figure 12 is an overview of developed GUI for displaying all the above results.

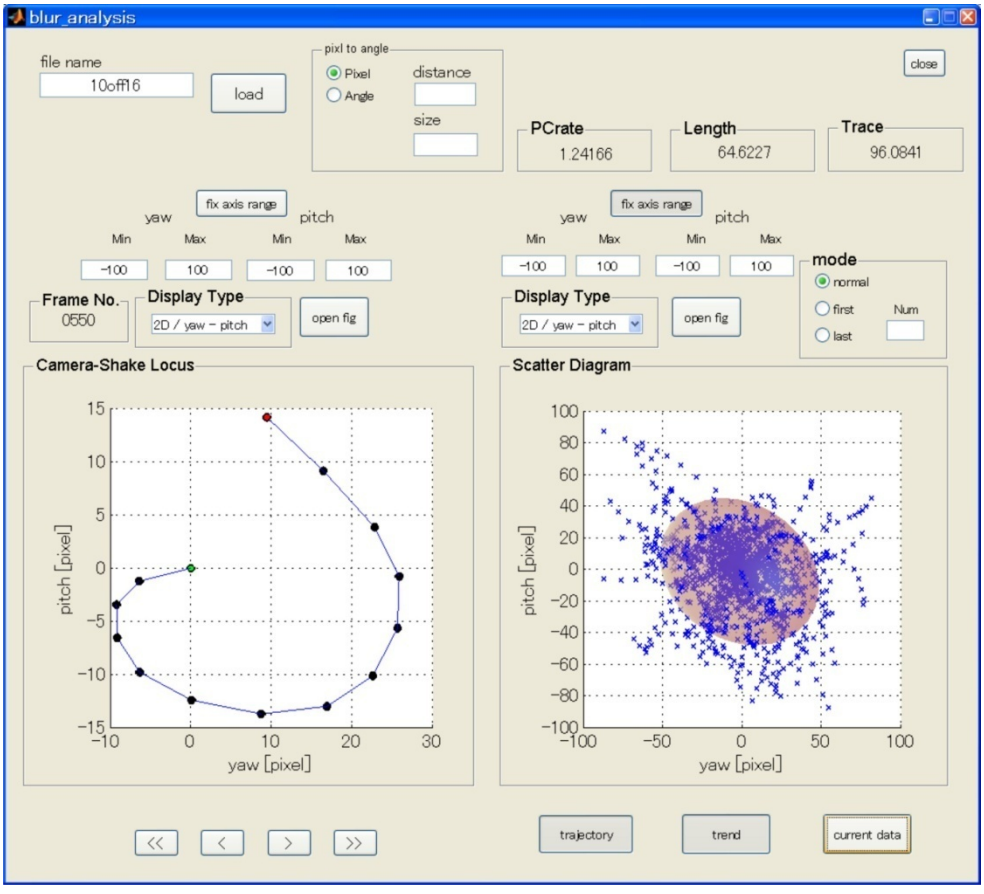


Fig. 12. Visualization and analysis tool of detected camera shakes

3. Experiments

3.1 Camera shake simulator

A camera shake simulator (see Figure 13) is used to examine the accuracy of detected camera shakes and compare the performance of image stabilizers under a definite motion. It consists of two stepping motors for vibrating in two axes chosen from among yaw, pitch and roll directions. The stage configuration for realizing arbitrary motion in yaw and pitch axes is shown in the left side of Figure 14. The case of yaw and roll axes is shown in the right side.

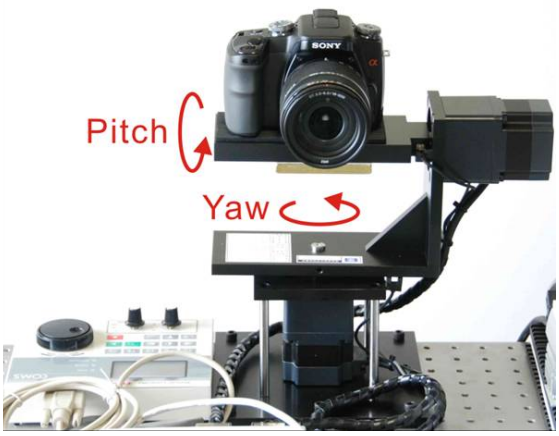


Fig. 13 Camera shake simulator for evaluating the detection accuracy and the effectiveness of image stabilizer

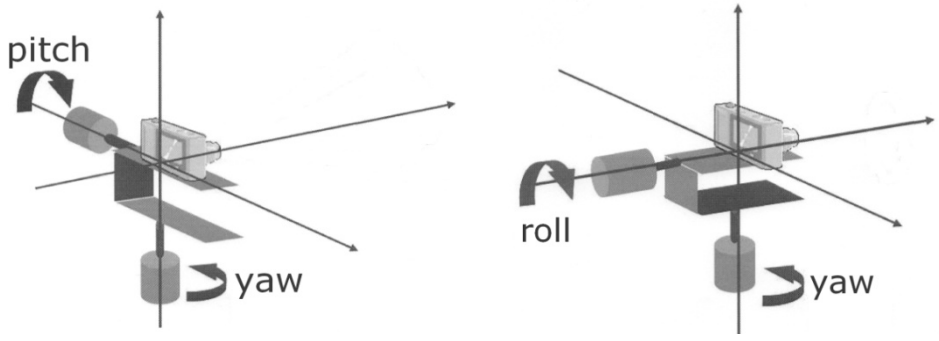


Fig. 14. Two types of vibration mode for realizing artificial camera shakes

3.2 Comparison of detection accuracy

A single-lens reflex camera is used to evaluate the detection accuracy in our measurement system. The left side of Table 1 shows standard deviations of detected vibrations from the camera images in static condition without any motions, where the camera is mounted on a

tripod and no mirror shocks exist. The right side of Table 1 shows measurement results under the motion by the camera-shake simulator that is controlled by an uniform circular motion in pitch and yaw axes.
This experiment is performed under the following conditions. The sizes of test pattern in the camera image are 800×800 pixels and 1200×1200 pixels, respectively. The rotation speed of the camera-shake simulator is constant at 1.68 degree/sec. The camera-to-subject distance is 3090mm.
The use of high resolution image in 1200×1200 pixels succeeds to reducing detection errors by 55%, 64% and 38% in pitch, yaw and roll axes, respectively, compared with the low resolutioni mage in 800×800 pixels. In the case of the circular motion, error reductions by 17%, 20% and 20% in pitch, yaw and roll, respectively, are also observed. These results show that the detection accuracy is improved by use of the high resolution pattern compared with the low resolution one.

	Static			Moving		
	pitch (deg.)	**yaw** (deg.)	**roll** (deg.)	**pitch** (deg.)	**yaw** (deg.)	**roll** (deg.)
800×800 pix.	0.0011	0.0014	0.0214	0.0030	0.0030	0.0270
1200×1200 pix.	0.0005	0.0005	0.0133	0.0024	0.0024	0.0176

Table 1. Comparison of accuracy with standard deviation of detected camera shakes

3.3 Personal differences of camera shake

Here we measured camera shake amplitudes on eight photographers. Each person shot about 140 pieces with the same camera. The size of test pattern in the camera image is about 1200×1200 pixels. The camera-to-subject distance is 3600mm.
Table 2 shows minimum and maximum camera rotations on each person, that is estimated through our measurement system. It shows that average values in pitch and yaw are about 0.03 degree and the average value in roll is 0.1 degree that means greater than pitch and yaw ones.

Person	**Pitch**(degree)		**Yaw**(degree)		**Roll** (degree)	
	min	max	min	max	min	max
A	-0.021	0.023	-0.016	0.025	-0.080	0.065
B	-0.030	0.035	-0.029	0.028	-0.075	0.085
C	-0.036	0.039	-0.033	0.033	-0.140	0.150
D	-0.032	0.032	-0.023	0.022	-0.065	0.065
E	-0.031	0.033	-0.041	0.040	-0.110	0.100
F	-0.021	0.025	-0.020	0.020	-0.105	0.011
G	-0.038	0.040	-0.036	0.040	-0.095	0.100
H	-0.039	0.044	-0.035	0.037	-0.115	0.115
Average	**-0.029**	**0.031**	**-0.031**	**0.034**	**-0.098**	**0.099**

Table 2. Personal differences and average trend of camera shakes

3.4 Performance evaluation of image stabilizers

Next experiment results for comparing the camera-shake reduction performance on four different cameras of single-lens reflex type with image stabilizers are shown. The pattern size is set to be 1200×1200 pixels in the camera image. All cameras on the camera-shake simulator take uniform circular motion at 1.59 Hz in yaw and pitch axes. Tow cameras of them have lens shift-type stabilizers and another two have sensor shift-type ones. The camera-to-subject distance is 3160mm.

Figure 15 shows 2D distributions of motion vectors of the detected camera-shakes in the case that image stabilizers are off and on. It menas that camera A has most effective performance of image stabilization because the distribution is most concentrated in origin.

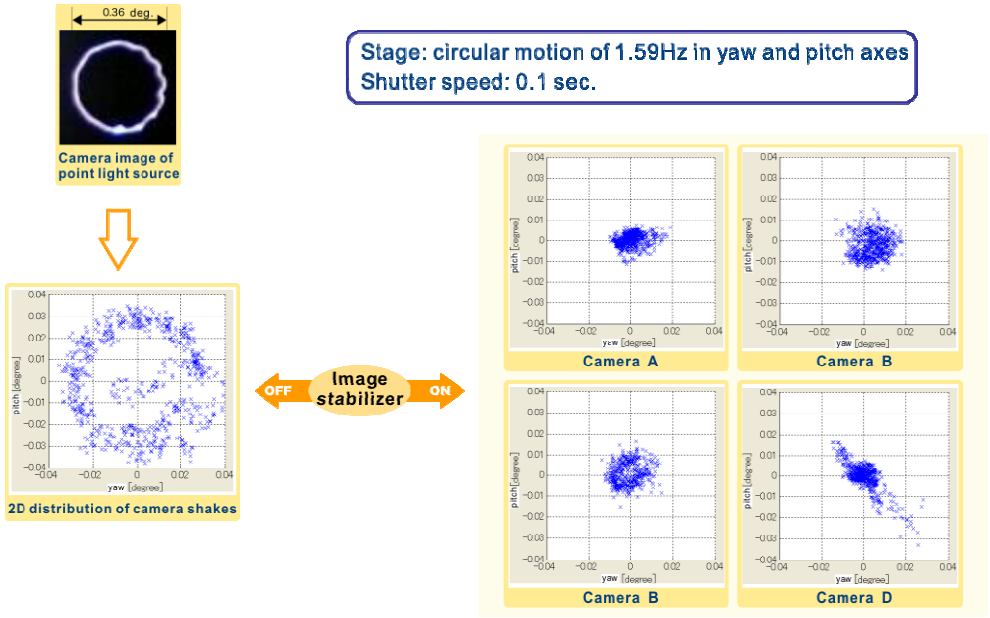


Fig. 15. Performance comparison of image stabilizers

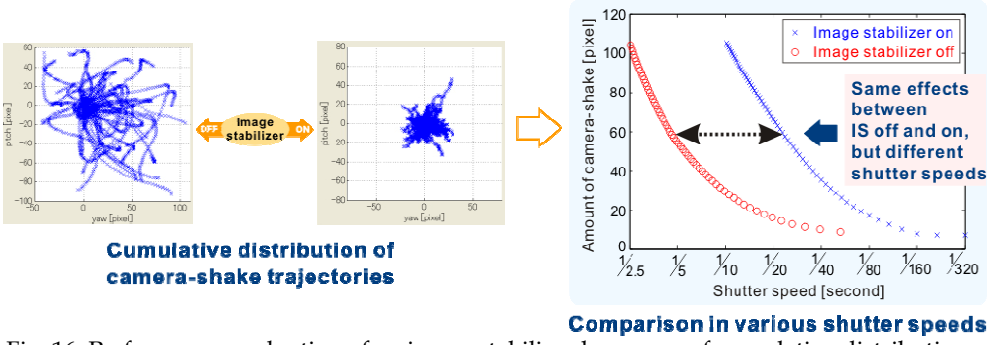


Fig. 16. Performance evaluation of an image stabilizer by means of cumulative distributions

The image stabilization of camera D becomes ineffective in 45-degree direction. Through these analyses, we can examine the tendency of image stabilization and exploit the result for the next development.

An example of evaluation using the cumulative distribution is shown in Figure 16. The distribution is widely spread due to camera-shakes, but the image stabilizer shrinks it in origin. From each distribution, amount of camera-shakes or the remainning one after the image stabilization are defined by the sum of long axis and short one of the oval sphere given by the principal compornent anaysis. The right side graph of Figure 16 is the computational results in various shutter speeds. In this graph, the shift distance between both curves of IS off and on means the total efficiency of image stabilizer.

3.5 Measurement of camera body and tripod vibration

Our system has a sub-pixel resolution in detection of dispacement due to camera motions. So not only camera-shakes but also slight vibrations in camera body and tripod when taking piictures can be detected.

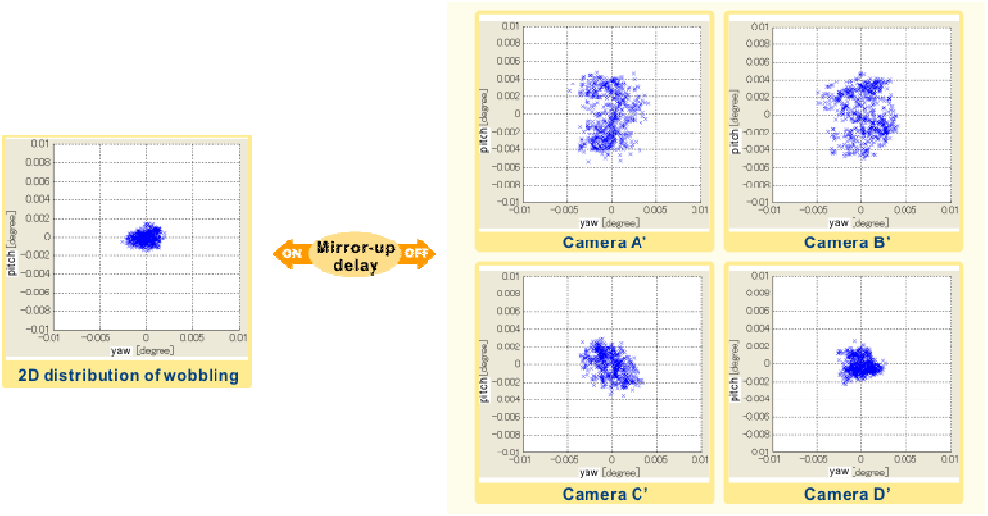


Fig. 17. Comparison of vibration suppression performance

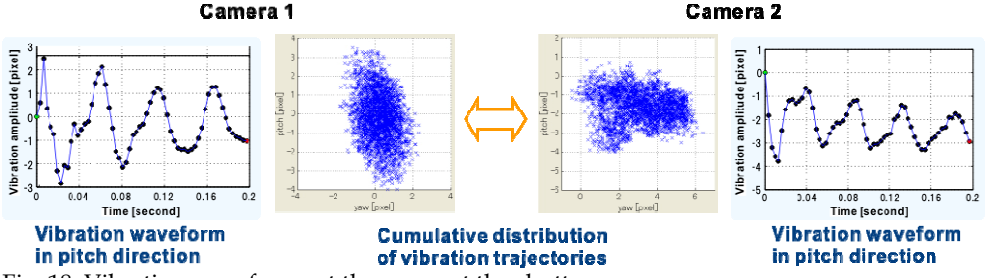


Fig. 18. Vibration waveforms at the moment the shutter opens

Figure 17 shows how much the camera vibration increases without mirror-up delay. Camera D', that the distribution is most concentrated in origin, has the most superior perforrmance for suppression of the camera vibration. The waveform of camera vibration on a triod can be also observed with a sub-pixel resolution (see Figure 18).

3.6 Measurement of camcorder-shakes and efficiency of image stabilizer in camcorder

The video image taken by camcorder is composed of sequential still images. The camera-shake in each still image, that is called *intra*-frame camera-shake, can be detected as with the above method. However the camera-shake arising in the interval between each frame, that is called *inter*-frame camera-shake, needs to be detected by another way. The inter-frame camera-shake corresponds to the image movement from previous frame to next one. Therefore the movement can be easily detected by using the pattern matching because each pattern is recored in different frame. By connecting between the estimated inter and intra-frame camera-shakes, the overall camera-shake can be obtained.
Figure 19 shows examples of detected camcorder-shake when the image stabilizer is off and on, respectively.

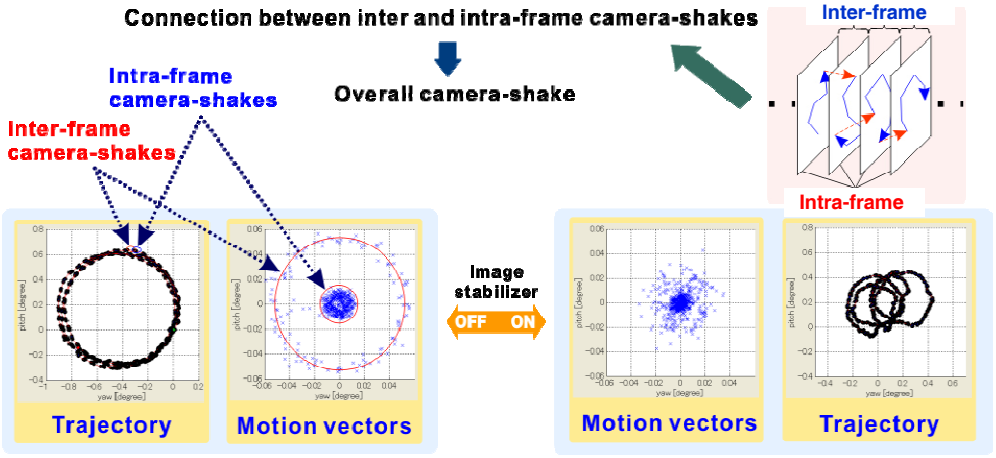


Fig. 19. Detection of the artificial camera-shake under a uniform circular motion

4. Conclusion and future work

A novel method for detecting the trajectory of camera-shake from the camera image itself was presented. The main feature is to use a video test pattern, detect positions of each test pattern multiply-recorded in the camera image by the pattern matching and obtain the camera-shake trajectory by connecting them. The LED display was also developed for realizing the high-response video test pattern. Many experimental resulsts and examples that show usefulness of this system were presented.
Now we are aiming at commercialization of this technology and standardization for measurement of camera-shake and evaluation of image stabilizers.

5. References

Choi, H.; Kim, J.P.; Song, M.G.; Kim, W.C., Park, N.P.; Park, Y.P. & Park, K.S. (2008). Effects of motion of an imaging system and optical image stabilizer on the modulation transfer function. *Optics Express*, Vol. 16, No. 25, pp. 21132-21141.

Choi, J.W.; Kang, M.G. & Park, K.T. (1998). An algorithm to extract camera-shaking degree and noise variance in the peak-trace domain. *IEEE Transactions on Consumer Electronics*, vol. 44, no. 3, pp. 1159-1168.

Fergus, R.; Singh, B.; Hertzmann, A.; Roweis, S.T. & Freemam, W.T. (2006). Removing Camera Shake from a Single Photograph. *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 787-794.

Nishi, K. & Ogino, R. (2007). 3D Camera-Shake Measurement and Analysis. *Proceedings of IEEE International Conference on Multimedia & Expo (ICME)*, pp. 1271-1274.

Park, S.C.; Lee, H.S. & Lee, S.W. (2004). Qualitative estimation of camera motion parameters from the linear composition of optical flow. *Pattern Recognition*, vol. 37, pp. 767-779.

Park, S.C.; Park, M.K. & Kang, M.G. (2003). Super-Resolution Image Reconstruction –A technical Overview-. *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21-36.

Sato, K.; Ishizuka, S.; Nikami, A. & Sato, M. (1993). Control Techniques for Optical Image Stabilizing System. *IEEE Transactions on Consumer Electronics*, vol. 39, no. 3, pp. 461-466.

Uomori, K.; Morimura, A.; Ishii, H.; Sakaguchi, T. & Kitamura, Y. (1990). Automatic image stabilizing system by full-digital signal processing. *IEEE Transactions on Consumer Electronics*, vol. 36, no. 3, pp. 510-519.

Xiao, F.; Silverstein, A. & Farrell, J. (2006). Camera-motion and effective spatial resolution. *Proceedings of International Congress of Imaging Science*, pp. 33-36

Yitzhaky, Y. & Kopeika, N.S. (1997). Identification of Blur Parameters from Motion Blurred Images. *Graphical Models and Image Processing*, vol. 59, no. 5, pp. 310–320.

12

Online surface inspection technology of cold rolled strips

Guifang Wu
Electronic Information Engineering College, Henan University of Science & Technology
China

1. Introduction

With enlargement of cold rolled strips' productive scale and enhancement of productive speed, traditional quality detection by human eyes becomes more and more unsuitable for surface detection of cold rolled strips, and multimedia technique such as machine vision become impendent. Recent years, CCD cameras and image processing techniques make online inspection of cold rolled strips possible (Sugimoto, T., 1998). An online surface inspection with CCD cameras and image processing techniques is a newly developed technique to detect surface defects of steels on production lines, and it can release workers from hard work and improve work efficiency.

Last century, many detection technique raised up, for example, in the early 70's, there are many new techniques in Japan such as Laser Scan Surface Defect Inspection System, and it developed continuously and can detect some of inner defects by the 80's. Till 90's, some of Japanese corporation developed a new system which can be used to detect the surface of electric steel. By the time of the end of 90's, camera techniques were used to detect the surface of metal strips, especially in German.

But because of the reason of computer technique, the techniques still stop in some normal work conditional, and the advantages of camera techniques weren't exerted fully, especially in developing countries and districts as China, India, etc. Consequently this newly developed technique becomes a hot subject, and its application problems are more and more concerned by people.

In this chapter, an automatic surface inspection system (Kim, S. M., 2006) of cold rolled strips is introduced. The system is equipped with 12 CCD area scan cameras which are used to capture surface image of strips simultaneously. With special illumination units, defects can be distinguished from backgrounds of images because of different gray levels. Many multimedia image processing algorithms are developed to decide whether there are defects on the images and to locate the defects in the whole strips. At the same time, pattern recognition algorithms are developed to classify the defects. With statistical data of defects, such as numbers, sizes, areas, classes, and so on, surface qualities of strips are evaluated.

2. General design of online surface inspection system

2.1 Requirement of System

When producing cold rolled strips, because of high speed, the surface of strips cannot be easily inspected by human eyes, if there are some defects on the surface, it cannot be found by inspector, and all the defects will be sometimes, many impurities would be embed in it, therefore, the surface condition would become too bad to be recognized. In order to inspect surface quality of hot rolled strips, the online inspection system must meet such performance indexes as followings:

1. Velocity requirement: It can work at the velocity of 5~15 m/s;
2. Detection size requirement: It can detect defects of strips with size of 0.5mm×0.5 mm;
3. Maximum width requirement: The maximum detection width can meet 2000 mm;
4. Work condition requirement: It can work online under any inner condition;
5. Defect position requirement: It can record exact position of defects;
6. Offline requirement: It can display history process of production, and have many offline function such as offline query, offline analysis and offline statistics;
7. Detection rate requirement: The detection rate of common defects can meet 90%, and about 80% for other defects;
8. Recognition rate requirement: The recognition rate of common defects can meet 85%, and about 80% for other defects;
9. Defect types requirement: It can detect over 10 main defects.

2.2 Chief defect types and their causes

When producing steel plate and strip, there are always such chief defect types as scratch, felt horizontal texture, point felt, feather roll imprint, white spot, roll imprint, Edge folding, rust mark, orange texture, emulsion mark, edge crack, scale, crack, air bubble, and so on. The following will analyze the clauses of these defects detailed.

1. Scratch

Scratch is a straight imprint on the surface of steel plate and strip. It is created by the relative movement between two surfaces of steel plate and strip, or between a hard sharp object and one of surface of steel plate and strip. Commonly, scratch is concave and directional in shape, and it is a bit light in black-white image.

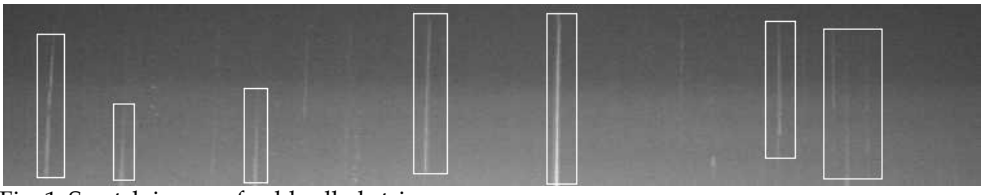


Fig. 1. Scratch image of cold rolled strips

2. Felt horizontal texture

Felt horizontal texture is horizontal wave texture along the edge of steel plate and strip, and it is also called edge pucker or edge folding mark. It is mainly caused by the sudden

deduced thickness of edge before temper rolling and extending not enough at edge when temper rolling, or unfolding not enough in rolling direction .this type defect always appears in cold rolled slight sheet.

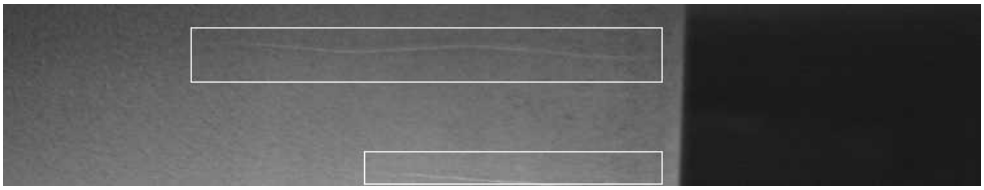


Fig. 2. Felt horizontal texture image of cold rolled strips

3. Point felt

Point felt is a felt phenomenon with a shape of point or block on the surface of steel plate and strip. The main cause is that the liquidity and plasticity of material is changed by bad shape or exceeded curl tension or bruise on surface.

Fig. 3. Point felt image of cold rolled strips

4. Feather roll imprint

Feather roll imprint is the same as a feather in shape. It is caused by linear load odds in the rolling process. At the same time, it is also related with raw material. The feather roll imprint appears generally in cold rolled steel sheet, and it is easy to form crack in bad condition.

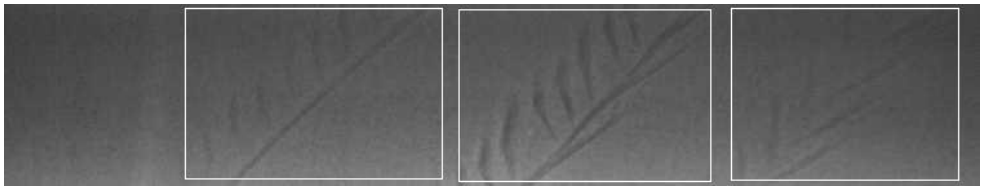


Fig. 4. Feather roll imprint image of cold rolled strips

5. White spot

White spot is a white similar circular spot block on the sur-face of steel plate and strip, its size is variable, but its border is smooth. It is primarily caused by white foam on the surface of plate and strip which has not been disposed immediately, it affect less on surface quality than surface appearance.

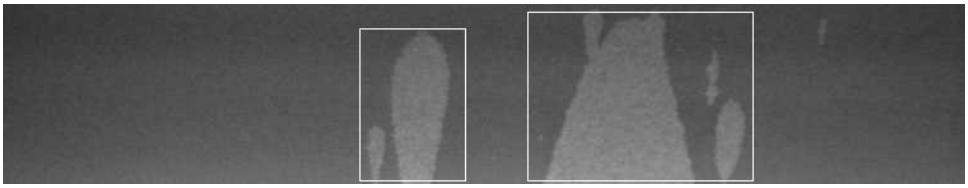


Fig. 5. White spot image of cold rolled strips

6. Roll imprint

Roll imprint is a periodic indentation on the surface of steel plate and strip, and appears more or less in a regular form. It is mainly caused by some hard and sharp foreign bodies or im-purities on work rollers, and every round the roller rolls over, there will leave a same indentation on the surface of steel plate and strip.

Fig. 6. Roll imprint image of cold rolled strips

7. Edge folding

Edge folding is a defect with a folding phenomenon on the edge of steel plate and strip. it is caused by untidy curled edge.

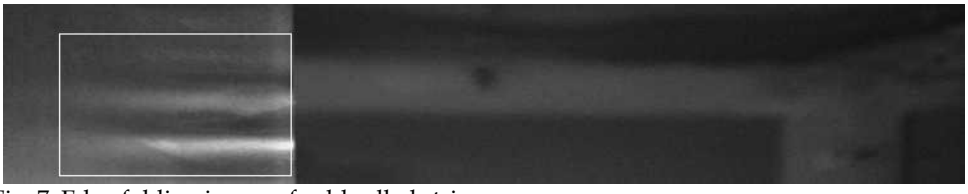


Fig. 7. Edge folding image of cold rolled strips

8. Rust mark

Rust mark is a thin layer of corrosion products on the surface of steel plate and strip. It is caused by high air humidity or small drips on surface under the circumstance of wide temperature fluctuations. If corrosion is very serious, the touch will be obvious, and it will be irregular light spot in image.

Fig. 8. Rust mark image of cold rolled strips

9. Orange texture
Orange texture is a block of irregular rough surface like orange skin and is touchable. It is caused mainly by incompletely removed impurities and greasy dirt on work roller during temper rolling.


Fig. 9. Orange texture image of cold rolled strips

10. Emulsion mark
Emulsion mark is a block of untouchable spot. It is caused by the incompletely washed oiliness of earlier rolling working procedure, it is hard to be removed from surface of steel plate and strip, and it generally presents dark in image.

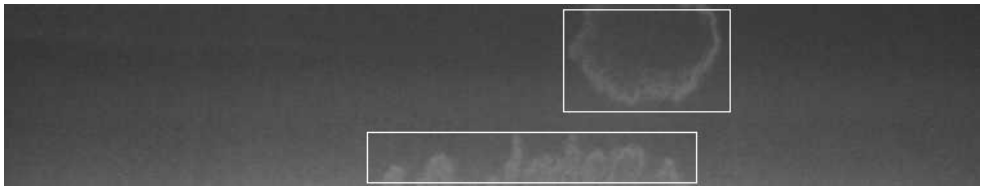


Fig. 10. Emulsion mark image of cold rolled strips

11. Edge crack
Edge crack is a phenomenon of crack or hole at the edge of steel plate and strip. It is caused by linear load odds or strong local stress when rolling or the capacity for deformation of the material is over exceeded.

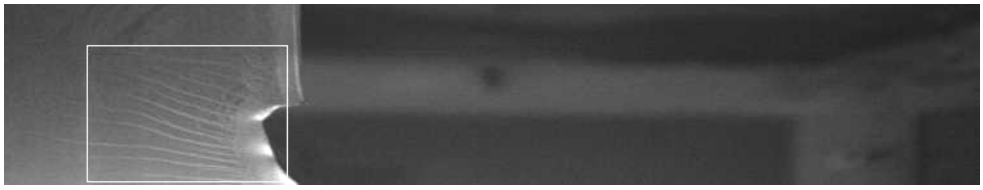


Fig. 11. Edge crack image of cold rolled strips

12. Scale
Scale is a lay of oxide covered on the surface of steel plate and strip. It is caused by high temperature and insufficient descaling when rolling. The scale can be removed by pickling, but it is hard to be completely eradicated.

Fig. 12. Scale image of cold rolled strips

13. Crack
Crack is a phenomenon of local break of steel plate and strip. Because of the surface tension odds caused by impurity inside steel plate and strip or surface stress concentration caused by surface sag of material, crack will appear while rolling. Crack is generally directional, and vertical crack is most frequent, in some special cases, star crack is occasional.

Fig. 13. Crack image of cold rolled strips

14. Air bubble
Air bubble is undischarged air inside steel plate and strip. Some air will discharge when rolling and will induce a pit in the surface of steel plate and strip. It is produced in the steel-making process.

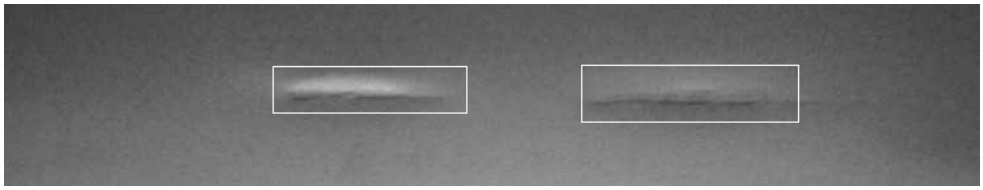


Fig. 14. Air bubble image of cold rolled strips

2.3 General system design

In order to content all the requirements of above, a system as Figure 15 is designed. The figure shows a general structure of an online inspection system. The system mainly includes 6 parts, such as detection devices, illumination, parallel computation system, server system, mass memory and console system.

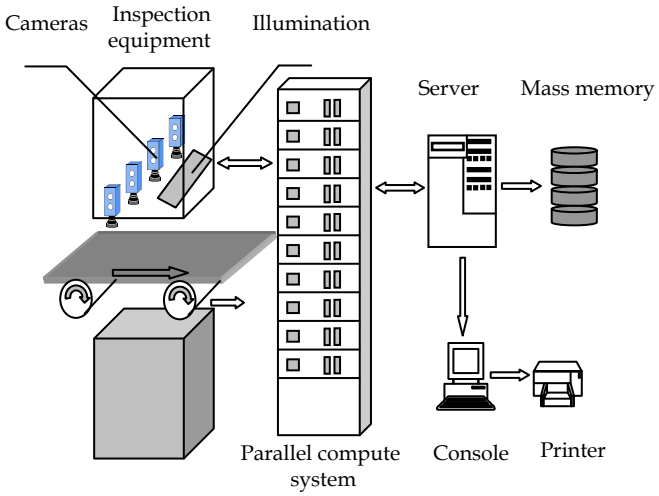
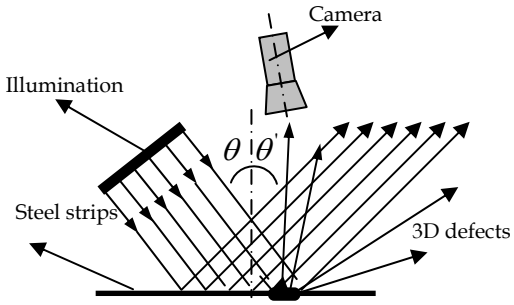
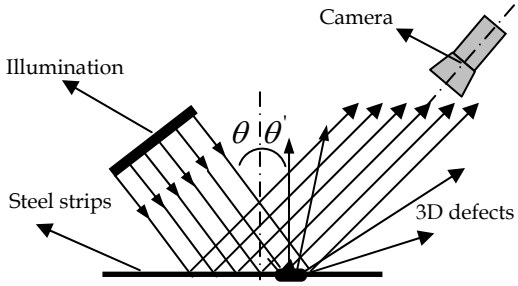


Fig. 15. General structure of an online inspection system

From Figure 15, it can be seen that detection devices are installed in the production line. One set of inspection device consists of several CCD cameras worked together with one illumination unit. Because there are two sides of strips, two sets of inspection devices are designed to meet the requirements, one is for upside inspection, the other is for down side inspection, and both sides of strips must be detected simultaneously.
Illumination part is very important and essential to the system because it determines whether defects are visualized in images captured by the cameras.
There are 2 illumination modes in this system such as dark illumination mode and bright illumination mode. The former is used to detect 3-dimensional defects, and the latter is used to detect 2-dimensional defects. Figure 16 shows these two different illumination modes, Figure 16 (a) is dark illumination mode, and Figure 16 (b) is bright illumination mode.



(a) Dark illumination mode



(b) Bright illumination mode

Fig. 16. Two different illumination modes

In dark illumination mode, the steel strips will reflect the light of illumination devie, and only the reflection light of 3D defects will be acquired by camera, but in bright illumination mode, most of reflection light of 3D defects will not be acquired by camera, the reflecrion light of 2D defects will enter into the camera and be acquired as object images.

All the algorithms of whole system are realized by software method, it brings conviences not only for system upgrade, but also for less outside hardware tache to reduce trouble rate. The software is mainly consist of 2 parts such as client software and server software, Figure 17 and Figure 18 are client software flow chart and server software flow chart respectively.

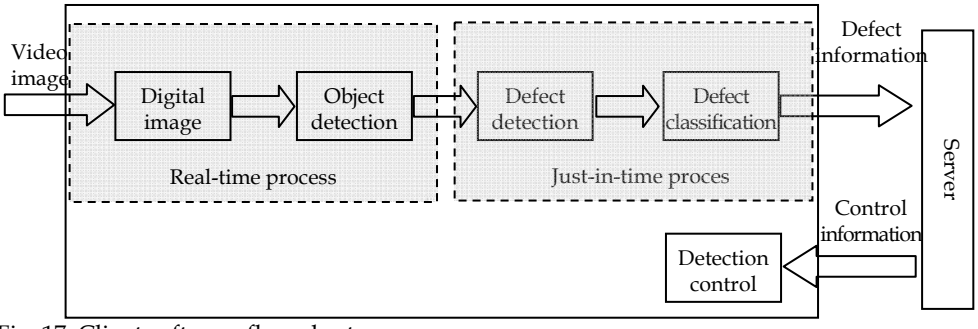


Fig. 17. Client software flow chart

Client software can meet data acquisition function of online inspection and online observation, including image acquisition, image pre-process and analysis, defect inspection, defect feature extraction and defect recognition etc, and it does not require human intervention as long as guaranteeing the veracity, high effect and stability of the client software. Server software mainly realizes offline defect data analysis, statistic report making and all the mutual control of the whole system, etc. At the same time, the server software provides visual man-machine interface, worker of quality inspection department can know about the surface quality of current coil and historic coils directly by the interface and it can make various statistic report according to the requirement of customer.

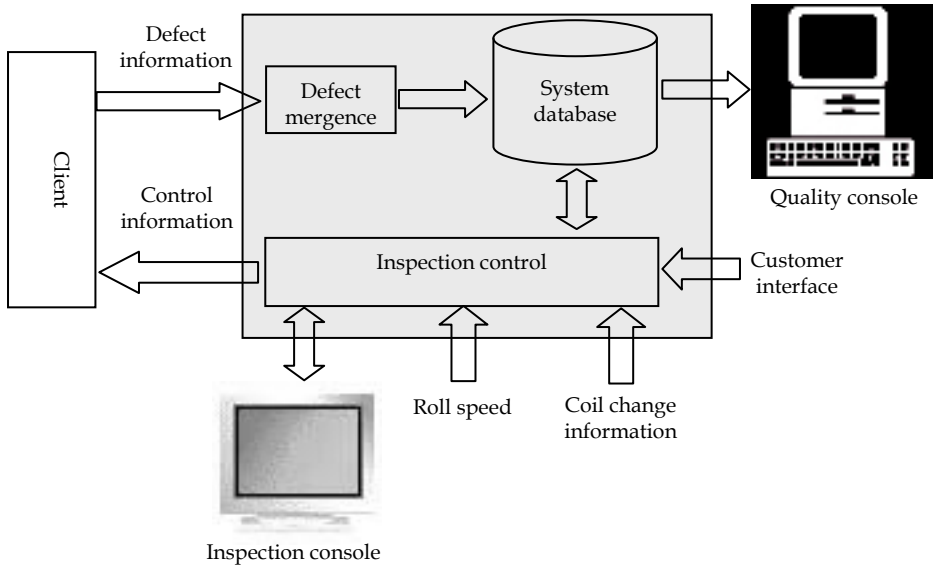


Fig. 18. Server software flow chart

In client software, there are 2 special processes in order realize high speed inspection, they are real-time process and just-in-time process. Real-time process takes charge of some immediate information process, so that the system can meet real-time requirement, and just-in-time process takes charge of other information process when CPU is not in full task, so that the system can settle large amount data problem.

3. Main technologies analysis

3.1 Image frozen techniques

When applies multimedia technologies to detect the defects of cold rolled strips with high speed, how to acquire high precision and clear image is very important, and it is a difficulty of automatic surface inspection system for online application. The image acquisition of moving objects techniques are mainly image frozen technique.

Image frozen techniques are adopted to capture surface images of strips at high speed. Shutters of cameras are set very high as 1/4000 seconds to make images jitter slightly, while the illumination units supply enough luminance intensity and make images captured by cameras clear. All cameras are synchronized to capture images simultaneously. To keep service performance and increase service life, the illumination units are not always in operation. They are synchronized with shutters of cameras to make them only lighten at the exposure of cameras.

3.2 Defects detection techniques of moving strips

As positions and shapes of strips vary continuously during producing, reflection of incident lights is different at different areas of strips. Therefore it leads to different backgrounds in

different images captured by cameras. Moreover, gray levels of background differ everywhere in one image because of uneven illumination and reflection. It is difficult to find out defects from non-uniform and ever-changing backgrounds. However, gray levels of backgrounds are gradually changed, while gray levels between backgrounds and defects are abruptly changed. Effective algorithms of defect detection are developed which consists of 4 steps as followings:

1. Background extracted: In this step, backgrounds are found and separated from original images.
2. Gray level calculation: Differential gray levels of backgrounds and original images are calculated.
3. Thresholds decision: One or more threshold values are decided and applied on differential gray levels to determine whether pixels are in defect areas or backgrounds, and pixels in defect areas are marked as "Suspicious Pixels".
4. ROI (Region of Interest) searching and merging; "Region growing" is used to find out defect areas by merging "Suspicious Pixels", and the defect area are called as "Region of Interest", or ROI.

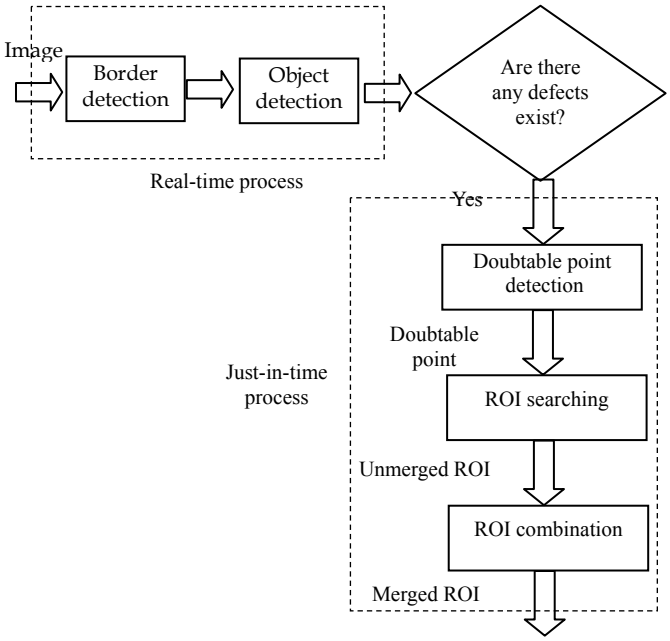


Fig. 19. Defects detection flow chart

3.3 Feature extraction and optimization techniques

3.3.1 Space-domain feature extraction

According to the defect character of cold rolled strips, we can extract 4 types of space domain features.

1. Geometric shape features

Geometric shape features are basic features of object, and they are used to describe geometric shape such as simple descriptor, shape descriptor and fixed quadrature. Simple

descriptor includes area, border perimeter, region center, etc. Suppose x, y are coordinates of horizon and vertical orientation, R is region area, R_x is aggregate of horizon coordinate, R_y is aggregate of vertical coordinate, $f(x,y)$ is image data, R_d is aggregate of defect point in ROI, R_b is aggregate of border point of ROI, some main geometric shape features can be described as followings:

1) Defect area s. It is used to describe the size of defect region as equation (1).

$$S = \sum_{(x,y)\in R_{\mathrm{d}}} 1 \tag{1}$$

2) Defect perimeter P. It is mean defect contour pixel gross as equation (2).

$$P = \sum_{(x,y)\in R_{\mathrm{b}}} 1 \tag{2}$$

3) Region center $\left(x_{\mathrm{c}}, y_{\mathrm{c}}\right)$. It is calculated by barycentric coordinate as equation (3).

$$\begin{cases} x_{\mathrm{c}} = \dfrac{1}{S}\displaystyle\sum_{(x,y)\in R_{\mathrm{d}}} x \\ y_{\mathrm{c}} = \dfrac{1}{S}\displaystyle\sum_{(x,y)\in R_{\mathrm{d}}} y \end{cases} \tag{3}$$

4) Defect compact C. It is calculated by equation (4).

$$C = \frac{\|P\|^2}{4\pi \cdot S} \tag{4}$$

5) Defect quadrature feature. There are total 7 quadrature features, and can be described as equation (5) to equation (11), they are fixed for shift, rotation and scale transform

$$\mathrm{IM}_1 = \eta_{20} + \eta_{02} \tag{5}$$

$$\mathrm{IM}_2 = \left(\eta_{20} - \eta_{02}\right)^2 + 4\eta_{11}^2 \tag{6}$$

$$\mathrm{IM}_3 = \left(\eta_{30} - 3\eta_{12}\right)^2 + \left(\eta_{03} - 3\eta_{21}\right)^2 \tag{7}$$

$$\mathrm{IM}_4 = \left(\eta_{30} + \eta_{12}\right)^2 + \left(\eta_{03} + \eta_{21}\right)^2 \tag{8}$$

$$\begin{aligned} \mathrm{IM}_5 = {} & \left(\eta_{30} - 3\eta_{12}\right)\left(\eta_{30} + \eta_{12}\right)\left[\left(\eta_{30} + \eta_{12}\right)^2 - 3\left(\eta_{21} + \eta_{03}\right)^2\right] \\ & + \left(3\eta_{21} - \eta_{03}\right)\left(\eta_{21} + \eta_{03}\right)\left[3\left(\eta_{30} + \eta_{12}\right)^2 - \left(\eta_{21} + \eta_{03}\right)^2\right] \end{aligned} \tag{9}$$

$$\mathrm{IM}_6 = \left(\eta_{20} - \eta_{02}\right)\left[\left(\eta_{30} + \eta_{12}\right)^2 - \left(\eta_{21} + \eta_{03}\right)^2\right] + 4\eta_{11}\left(\eta_{30} + \eta_{12}\right)\left(\eta_{21} + \eta_{03}\right) \tag{10}$$

$$\begin{aligned} \mathrm{IM}_7 &= \left(3\eta_{21}-\eta_{03}\right)\left(\eta_{30}+\eta_{12}\right)\left[\left(\eta_{30}+\eta_{12}\right)^2-3\left(\eta_{21}+\eta_{03}\right)^2\right] \\ &+\left(3\eta_{12}-\eta_{30}\right)\left(\eta_{21}+\eta_{03}\right)\left[3\left(\eta_{30}+\eta_{12}\right)^2-\left(\eta_{21}+\eta_{03}\right)^2\right] \end{aligned} \tag{11}$$

There are total 19 geometrical feature parameters extracted.

2. Gray level features

Gray level features are very useful in classification and they depict many essential characters of gray image. They are mostly consisted of average, variance, kurtosis, skewness, enrage and entropy of gray level etc., all these characters can be calculated by gray level histogram. There are total 36 gray level character parameters extracted in the software.

1) Gray level average $\bar{b}$

$$\bar{b}=\sum_{b=0}^{L-1} bP(b) \tag{12}$$

2) Gray level variance v

$$v^2=\sum_{b=0}^{L-1}\left(b-\bar{b}\right)^2 P\left(b\right) \tag{13}$$

3) Gray level kurtosis H_s

$$H_{\mathrm{s}}=\frac{1}{v^3}\sum_{b=0}^{L-1}\left(b-\bar{b}\right)^3 P\left(b\right) \tag{14}$$

4) Gray level skewness H_k

$$H_{\mathrm{k}}=\frac{1}{v^4}\sum_{b=0}^{L-1}\left(b-\bar{b}\right)^4 P\left(b\right)-3 \tag{15}$$

5) Gray level enrage H_p

$$H_{\mathrm{p}}=\sum_{b=0}^{L-1} P(b)^2 \tag{16}$$

6) Gray level entropy H_E

$$H_{\mathrm{E}}=-\sum_{b=0}^{L-1} P\left(b\right)\log_2\left[P\left(b\right)\right] \tag{17}$$

At the same time, there are 3 other gray level features such as max gray level, min graylevel and gray level scope, therefore there are total 9 gray level types.

Now, we can extract up 9 gray levels for background images, defect suspicious pixel images, contrast images and defect images respectively, and can get total 36 gray level features.

3. Projection features

It is always very practical to map high-dimensional data into low-dimensional data in engineering, because low-dimensional data are much easier to analyze than high-dimensional data. Projection features are extracted by mapping two-dimensional image signal to one-dimensional digital wave signal, such as wave features, peak value features, impulse features, margin features, skewness features, kurtosis features, and so on. In Figure 20, a defect in image is projected to X direction, and formed a Gray level – X coordinate. There are total 24 projection features in the software.

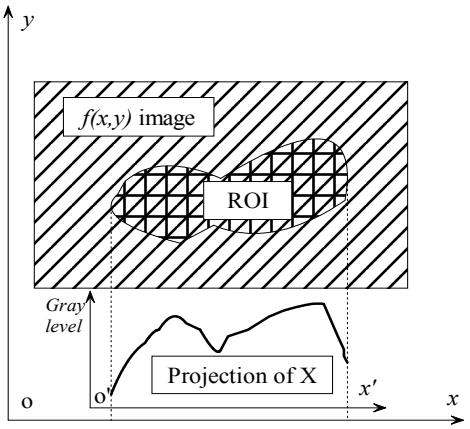


Fig. 20. X direction projection of defect in an image

Defect image can project to X aix, Y aix, 45° diagonal and 135° diagonal, they can be described as equation (18) to equation (21) .

$$p\left(0^{\circ},t\right)=\int_{R_{\mathrm{d}}} f\left(t,y\right)dy \tag{18}$$

$$p\left(90^{\circ},t\right)=\int_{R_{\mathrm{d}}} f\left(x,t\right)dx \tag{19}$$

$$p\left(45^{\circ},t\right)=\int_{R_{\mathrm{d}}} f\left(x,x-\sqrt{2}\,t\right)\mathrm{d}x \tag{20}$$

$$p\left(135^{\circ},t\right)=\int_{R_{\mathrm{d}}} f\left(x,\sqrt{2}\,t-x\right)\mathrm{d}x \tag{21}$$

According to the 4 types projection, we can get

1) Peak value $\hat{X}$:

$$\hat{X}=\max\left(\left|x\left(t\right)\right|\right) \tag{22}$$

2) Average peak value $\bar{X}_{\mathrm{p}}$:

$$\bar{X}_{\mathrm{p}}=\frac{1}{T}\sum_{t}\left|x\left(t\right)\right| \tag{23}$$

3) Mean square root X_{RMS} :

$$X_{\mathrm{RMS}}=\sqrt{\frac{1}{T}\sum_{t}x^{2}\left(t\right)dt} \tag{24}$$

4) Square root scope X_{r} :

$$X_{\mathrm{r}}=\left(\frac{1}{T}\sum_{t}\left|x\left(t\right)\right|^{1/2}dt^{2}\right) \tag{25}$$

Consequently, we define 6 projection features as following:

1) Projection wave features F_{B} :

$$F_{\mathrm{B}} = \frac{X_{\mathrm{RMS}}}{\bar{X}_{\mathrm{p}}} \tag{26}$$

2) Projection impulse features F_{M} :

$$F_{\mathrm{M}} = \frac{\hat{X}}{\bar{X}_{\mathrm{p}}} \tag{27}$$

3) Projection peak value features F_{F} :

$$F_{\mathrm{F}} = \frac{\hat{X}}{X_{\mathrm{RMS}}} \tag{28}$$

4) Projection margin features F_{Y} :

$$F_{\mathrm{Y}} = \frac{\hat{X}}{X_{\mathrm{r}}} \tag{29}$$

5) Projection skewness features F_{S} :

$$F_{\mathrm{S}} = \sum_{x} x^3 p(x)\, dx \tag{30}$$

6) Projection kurtosis features F_{K} :

$$F_{\mathrm{K}} = \sum_{x} x^4 p(x)\, dx \tag{31}$$

According to 4 direction projection, there are total 24 projection features can be extracted, and can be used to recognize the defects.

4. Textural features

For some special defects, they have obvious textural features, Figure 21 shows an image with obvious textural features.

Fig. 21. Typical defect image with obvious textural features

Textural features can be achieved by gray level co-occurrence matrix with statistics, as in equation (32).

$$P(g_1, g_2) = \frac{\#\left\{\left[\left(x_1, y_1\right), \left(x_2, y_2\right)\right] \in S \mid f\left(x_1, y_1\right) = g_1 \,\&\, f\left(x_2, y_2\right) = g_2\right\}}{\#S} \tag{32}$$

According to the gray level co-occurrence matrix, 4 textual features such as second-order moment, entropy, contrast and homogeneity can be achieved.

1) Textural second-order moment W_{M} :

$$W_{\mathrm{M}} = \sum_{g_1} \sum_{g_2} P^2\left(g_1, g_2\right) \tag{33}$$

2) Textural entropy W_{E} :

$$W_{\mathrm{E}} = -\sum_{g_1} \sum_{g_2} P\left(g_1, g_2\right)) \log P\left(g_1, g_2\right) \tag{34}$$

3) Textural contrast W_{C} :

$$W_{\mathrm{C}} = \sum_{g_1} \sum_{g_2} \left|g_1 - g_2\right| P\left(g_1, g_2\right) \tag{35}$$

4) Textural homogeneity W_{H} :

$$W_{\mathrm{H}} = \sum_{g_1} \sum_{g_2} \frac{P\left(g_1, g_2\right)}{k + \left|g_1 - g_2\right|} \tag{36}$$

3.3.2 Frequency-domain feature extraction

In order to extract frequency domain feature, all the defect images are given by the size of 64 $\times$64 shown as Figure 22.

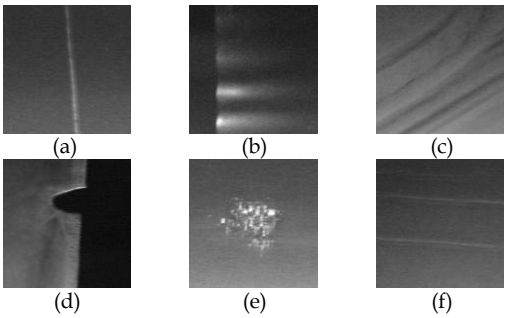


Fig. 22. Some typical defect images with size of 64$\times$64, (a) is scratch, (b) is edge folding, (c) is feather imprint, (d) is edge crack, (e) is point felt and (f) is felt horizontal texture.

Applying Fast Fourier Transform (FFT) (Chu, E., 2000) to the images and the spectrum images can be achieved. According to the qualities of Fast Fourier Transform, Fast Fourier Transform can transform images from space-domain to frequency-domain, and concentrated most of energies on the center, it can not only reflect gray features and geometrical features of defect images, but also make fast convolution and object recognition to become very real. Therefore, regarding gray values of spectrum images as feature information of defect images can recognize defects, especially the pixels near the central area of spectrum images. The typical spectrum images of FFT are shown as Figure 23.

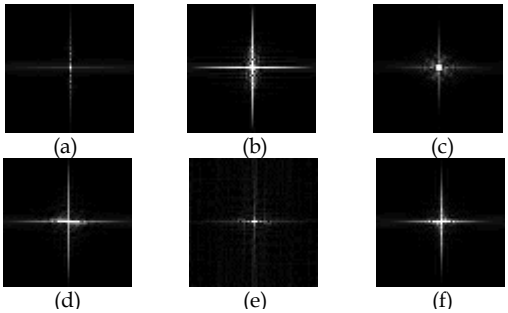


Fig. 23. Typical FFT images of Figure 22 with size of 64×64, (a) is scratch, (b) is edge folding, (c) is feather imprint, (d) is edge crack, (e) is point felt and (f) is felt horizontal texture.

According to the result of FFT, it can be seen that nearly all energies are concentrated on the center area, and at the same time, if make all pixels of FFT image as features, the feature set will be too large and unnecessary, and will affect recognition result to some extent in fact. As a consequence, based on many experiments, the crisscross region shown as Figure 4 is taken as original feature set, and the feature number can be calculated with equation (37).

$$n = 32 \times (a + b) - ab \tag{37}$$

In the paper, parameter a is 4 and parameter b is 4 too. Therefore, the total chosen region includes 240 pixels, and the original feature set includes 240 features. The feature set can reflect local outstanding feature information of frequency-domain.

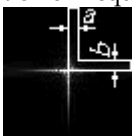


Fig. 24. Optimized crisscross original Feature set region. There are two parameters, parameter a is region width and parameter b is region height.

After given an original feature set, another two extended features are introduced, one is named Sum of Valid Pixels (SVP), and the other is Repletion Ratio of Centre Region (RRCR). The definition of SVP is the count of all the pixels whose values are larger than a given threshold in the full FFT images, therefore by defining SVP, the region outside the crisscross region is taken into consideration, and the feature reflects global statistic feature information of frequency-domain.

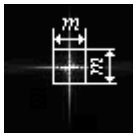


Fig. 25. Definition of Repletion Ratio of Centre Region (RRCR) and parameter m is width of centre square region.

Figure 25 shows the definition of RRCR, and it can be defined as equation (38):

$$\eta_m = N / m^2 \times 100\%, \qquad m < 64 \tag{38}$$

N is amount of features whose value are greater than a given threshold value in a $m \times m$ centre region. Parameter m can be given as a needed value. This feature can reflects local statistic feature information of frequency-domain in low frequency section. In the paper, three RRCR features are extracted, they are η_5 , η_{10} and η_{15}.

3.3.3 Feature optimization

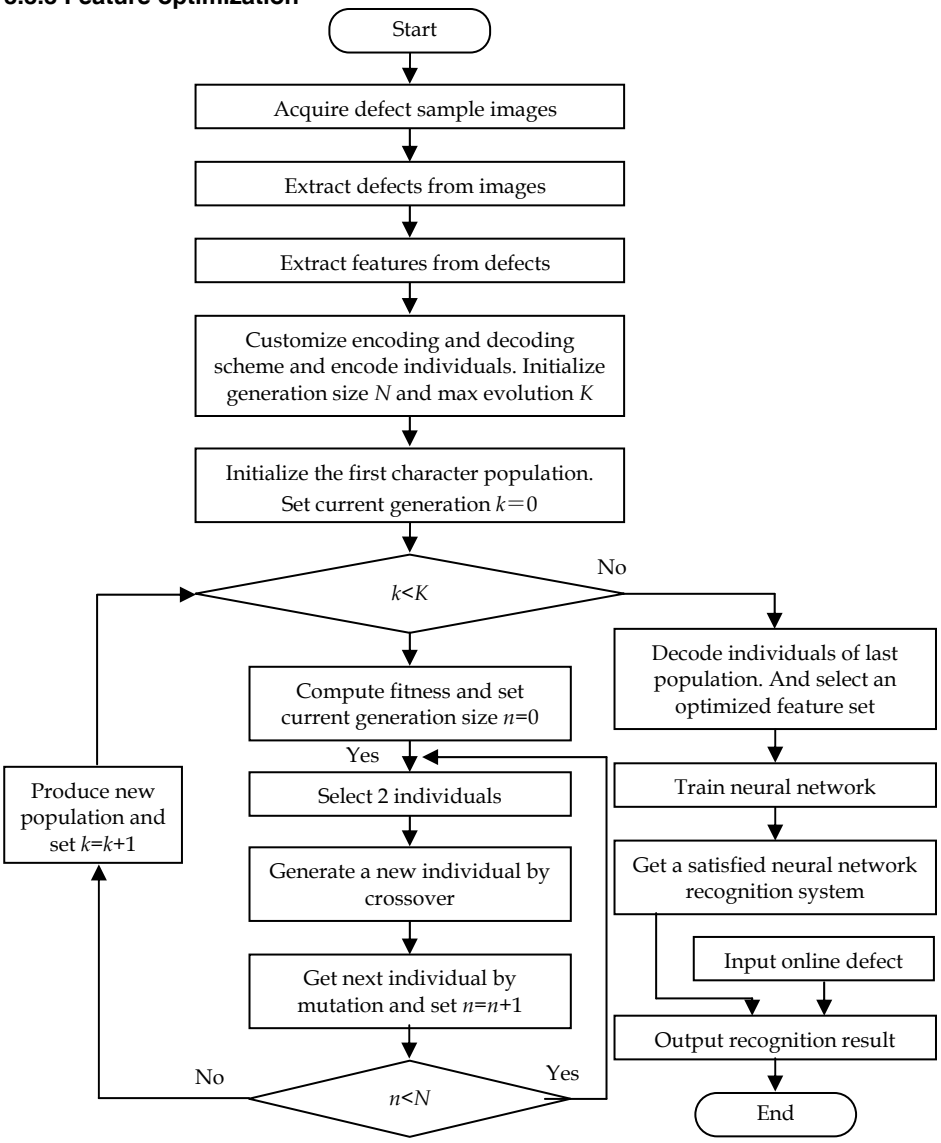


Fig. 26. Feature optimization and defect recognition of hot rolled strips

The process of using genetic algorithm (Hannes, L., 2006) to optimize the feature set is a process of setting the parameters of genetic algorithm which can describe problem best. Genetic algorithm is used to optimize the region of original crisscross frequency-domain feature set which includes 240 features and space-domain feature set which includes 82 features.

In order to find a best fitness function for genetic algorithm, we tried to use mutual information entropy as fitness function. It is known that the object function of mutual information entropy is a problem of the largest values. Based on experiments, and according to analysis of surface defects of steel strips, mutual information entropy is made as fitness function of genetic algorithm, and it is shown as equation (39);

$$\begin{aligned} I(E,F) &= \frac{1}{N}[I^*(E,F) - I(F)] \\ &= \frac{1}{N}[\sum_{i=1}^{n}\sum_{j=1}^{N} I(w_i, x_j) - \frac{1}{N-1}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} I(x_i, x_j)] \end{aligned} \tag{39}$$

where

$$I(w_i, x_j) = \sum_{k=1}^{K} p(w_i, x_{j,k}) \bullet \log \frac{p(w_i, x_{j,k})}{p(w_i) \bullet p(x_{j,k})} \tag{40}$$

$$I(x_i, x_j) = \sum_{h=1}^{K}\sum_{k=1}^{K} p(x_{i,h}, x_{j,k}) \bullet \log \frac{p(x_{i,h}, x_{j,k})}{p(x_{i,h}) \bullet p(x_{j,k})} \tag{41}$$

3.4 Local border search algorithm

According to the statistics of cold rolled strip, there are more than 30% defects in border region, therefore how to extract the border from acquired images fast and accurate is very important for defect detection. Figure 27 shows a local pixel image with a 45° border. From the figure, it can be seen that the next border point will arise in one neighbor pixel, and if the border angle is smaller than 45o, the next border point will arise among on neighbor pixel. Therefore, , the next border point must be found if let the algorithm search a small neighbor region such as 3 pixels, and here bring forward Local Border Search Algorithm (LBSA) as follows. Figure 28 is flow chart of the new algorithm.

After the algorithm finishes, the border can be get, and every horizontal border point coordinates can be expressed with equation (42) and vertical border point coordinate can be expressed with equation (43).

$$\left(x, a[x]\right) \quad x = 0,1,2,\cdots, Width - 1 \tag{42}$$

$$\left(a[y], y\right) \quad y = 0,1,2,\cdots, Height - 1 \tag{43}$$

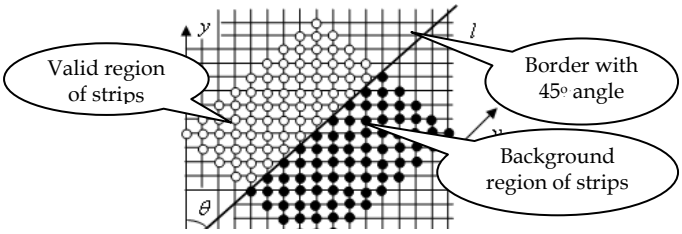


Fig. 27. A local pixel image with a 45 border

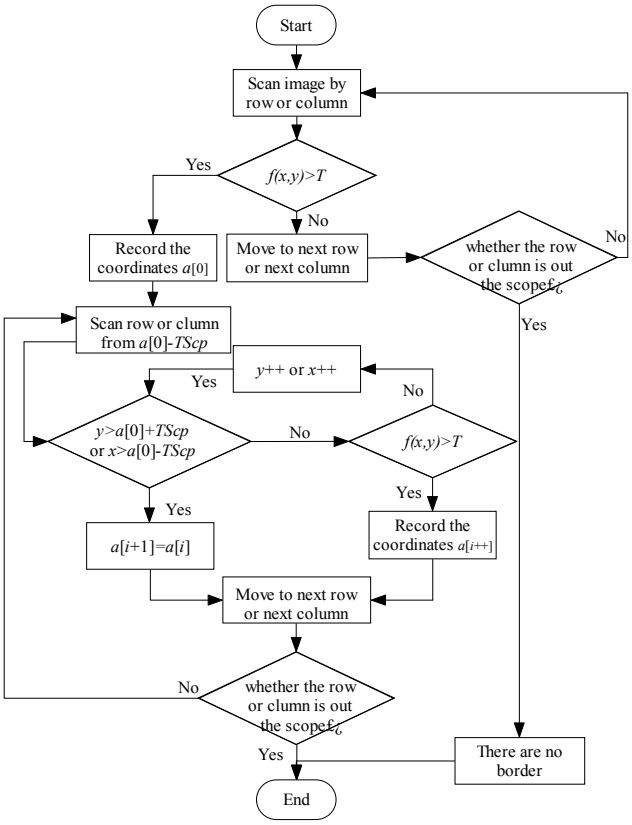


Fig. 28. Flow chart of Local Border Search Algorithm (LBSA)

For some complicated border types shown as Figure 33, it can be get both two horizontal and vertical border lines with equation (44), and then utilize the condition as equation (45) to calculate the point of intersection(X,Y) of the two border line, and the valid region of strips can be achieved.

$$\begin{cases} (x, a[x]) & x = 0,1,2,\cdots, Width-1 \\ (b[y], y) & x = 0,1,2,\cdots, Height-1 \end{cases} \tag{44}$$

$$\begin{cases} x = b[y] \\ y = a[x] \end{cases} \tag{45}$$

Fig. 29. Left border image of cold rolled strips

For example, Figure 29 is a 756×141 cold rolled strip image, and the valid region of steel strip lies in the left part of the image. When input the image into the Local Border Search Algorithm and give its border type, the algorithm can firstly search from right of row 0 with the coordinates of $(755,0)$, till first border point is found, and records the horizontal coordinate $a[0]$, it will meet the condition $\left|f(a[0],0)-f(a[0]+1,0)>T\right|$. After that the algorithm can search the next border point with a given scope of $TScp=3$, and the search scope of row 1 is $[a[0]+3, a[0]-3)$, and the border point must be appeared in the small region, so that a point $(a[1],1)$ can be found and it will meet the condition of $\left|f(a[1],1)-f(a[1]+1,1)>T\right|$. Similarly, the algorithm can run row by row till all border point are found, and get all the coordinates $a[2]$、$a[3]$、$\cdots$、$a[139]$、$a[140]$, and at last, the search result is as Figure 30 With the same method, other border type can be achieved.

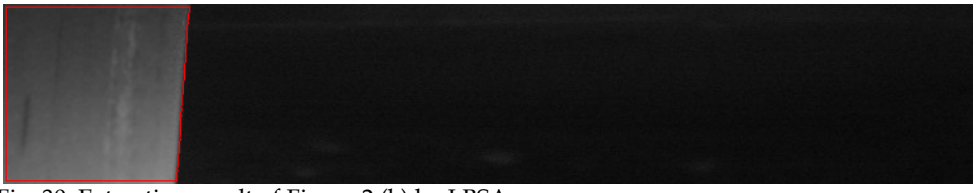
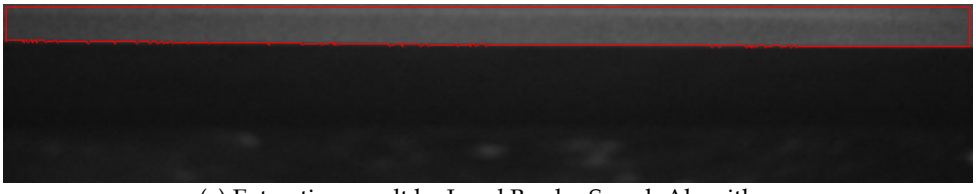
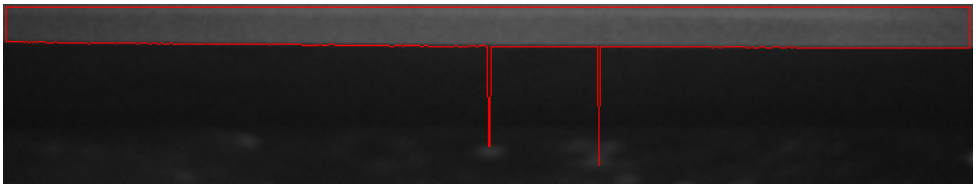


Fig. 30. Extraction result of Figure 2 (b) by LBSA



(a) Extraction result by Local Border Search Algorithm



(b) Extraction result by traditional algorithm

Fig. 31. Contrast effects of extraction results

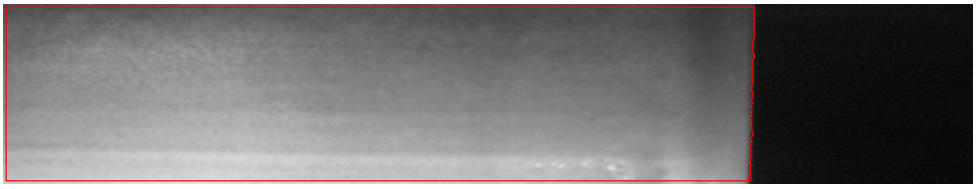


Fig. 32. Extraction result left valid region with right border which is near the border of the image

Fig. 33. Extraction result of complicated border type of top-right border

Figure 31 is the contrast effects of extraction result for the same image with different algorithm, and Figure 31(a) is extracted by Local Border Search Algorithm (LBSA), and Figure 31(b) is extracted by traditional Gray Level Gradient Threshold Algorithm (GLGTA). It can be seen that there are some noise inside Figure 31(b) but the effect is very perfect in Figure 31(a). Figure 33 shows the extraction result of complicated border type of top-right border. Table 1 shows the contrast effects of extraction results from 300 right border images, and it can be seen that the speed of the new algorithm is increased over 22-220 times than the traditional algorithm.

Sample image	Distance between strip border and image border (Pixels)	Time cost of Gray Level Gradient Threshold Algorithm (T1: ms)	Time cost of Local Border Search Algorithm (T2: ms)	T1/T2 (times)
Fig. 30	Greater than 600 pixels	2.0~2.2	0.01~0.03	67~220
Fig. 32	Smaller than 200 pixels	0.65~0.75	0.01~0.03	22~75

Table 1. Contrast table of right border extraction

Although Local Border Search Algorithm is more advantageous in search accuracy and speed of border than traditional algorithms, it is still short in some aspects. The main shortages are: (1) It is strongly dependent on prior knowledge; (2) It is sensitive to initial point.

In order to overcome the weaknesses above, deep research is carried out mainly from following three aspects:
1. Introducing a small neighborhood to the algorithm
Changing comparison between pure pixels to two small neighborhoods, generally, the neighborhood can be selected as a $m \times n\left(m > 0, n > 0\right)$ regular area or crisscross area. Then by calculating average of neighborhood or median etc. replace old gray value with the result. After improving it, the algorithm can remove disturbance of local noise, raise accuracy of border, and make the algorithm owning a higher quality of anti-noise. The neighbors can be chose as Figure 34.

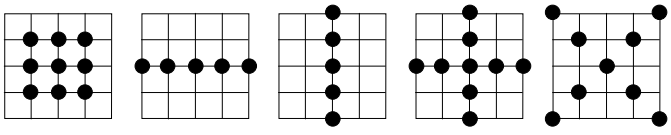


Fig. 34. Neighbor region in common use, (a) Square neighbor; (b) Horizontal neighbor; (c) Vertical neighbor; (d) 3×3 strong connection neighbor; (e) 3×3 weak connection neighbour

2. Modifying search rules of initial point
Whenever the gray value change reaches a given threshold value, recording the point as initial point of border. If there is no such point, enter next loop till such point appears. The advantage of this method is that we can find the border point even with no gray changes.
3. Adding judgment rule of ending point
Defining wrong permitting search times in Local Border Search Algorithm, if no border ending point is found in the scope of wrong permitting search times, the algorithm continues, when there is no border ending point is found beyond the scope of wrong permitting times, the algorithm stops.
After improving Local Border Search Algorithm from the three aspects above, the expanded algorithm can still keep the complexity as $\mathrm{O}(n)$, and meanwhile, it can be used to detect high contrast defects of cold rolled strips fast and accurately, and can be used as fast thinning algorithm of image.

3.5 Defect classification techniques

Artificial neural network is a parallel distributed information processing system. It is a network constituted by many processing units which are connected each other by unidirectional signal path. These processing units are called artificial neural units, and have the abilities of local memory and local information processing. Every processing unit outputs to an expected connection unidirectionally and transmits the same signal which is called output signal of processing unit. The output signal of processing unit can be any demanded mathematic types. Under complete local constraint circumstances, information processing executed in every processing unit can be any definition, namely the information processing depends only on the memorized value of itself and current value of the input stimulated signal of the processing unit. Neural network is an abstraction and simplification of human brain in microstructure and function, and reflects some fundamental characters such as parallel processing, learning, association, pattern classification , memory, and so on.

Obviously, according to the above characters of neural network, applying neural network to surface defect recognition of cold rolled strips is a good choice. In the application, LVQ neural network (Kohonen, T., 1992) is applied to recognize the defects.

1. LVQ neural network

LVQ neural network consists of 3 parts such as input layer, hidden layers and output layer. The hidden layer is also called competitive layer or Kohonen layer, and it is completely linked between input layer and hidden layer but just partly connected between hidden layer and output layer, and every output neuron is connected with different group of hidden neuron. The training process of LVQ neural network is in fact a process of continually competition and weight updating process among all neuron units of hidden layer. Figure 35 is topological structure of LVQ neural network.

Given a sequence of input documents, an initial group of reference vectors w_k is selected. In each iteration, a x_i document is selected and the vectors w are updated, so that it will adapt better to xi. The LVQ algorithm works as follows.

For each class k, a weight vector wk is associated. In each repetition, the algorithm selects a xi input document and compares it with each weight vector w_k, using the Euclidean distance $\|x_i - w_k\|$, so that the winner will be the weight vector w_c nearest to x_i, with c as the index of that weight vector:

$$\|x_i - w_c\| = \min_k \{\|x_i - w_c\|\} \tag{46}$$

The classes compete among themselves in order to find which is most similar to the input vector, so that the winner will be that one with less euclidean distance with respect to the input document. Only the winner class will modify its weights using a reinforced learning algorithm, either positive or negative, depending on whether the classification is correct or not. Thus, if the winner class belongs to the same class as the input vector, it will increase the weights, moving slightly closer to the input vector (prize). On the contrary, if the winner class is different from the input vector class, it will decrease the weights, moving slightly further from the input vector (punishment). Let $x_i(t)$ be an input document at time *t*, and $w_k(t)$ the weight vector for the class *k* at time *t*.

The following equations define the basic learning process for the LVQ algorithm:

$$w_c(t+1) = w_c(t) + s \cdot \alpha(t)[x_i(t) - w_c(t)] \tag{47}$$

where $s = 0$, if $k \neq c$; $s = 1$, if $x_i(t)$ and $w_c(t)$ belong to the same class; and $s = -1$, if they are not the same class. Where $\alpha(t)$ is learning rate, always between 0 and 1, is a monotonically decreasing function of time. It is recommended that $\alpha(t)$ should initially be rather small, especially smaller than 0.1, and $\alpha(t)$ continues decreasing to a given threshold *u*, very close to 0. Usually, $\alpha(0)$ is always initialized in 0.005, decreasing linearly to $u = 0.001$, according to the following equation where *K* is the number of classes：

$$\alpha(t+1) = \alpha(t) - \frac{\alpha(0) - u}{K} \tag{48}$$

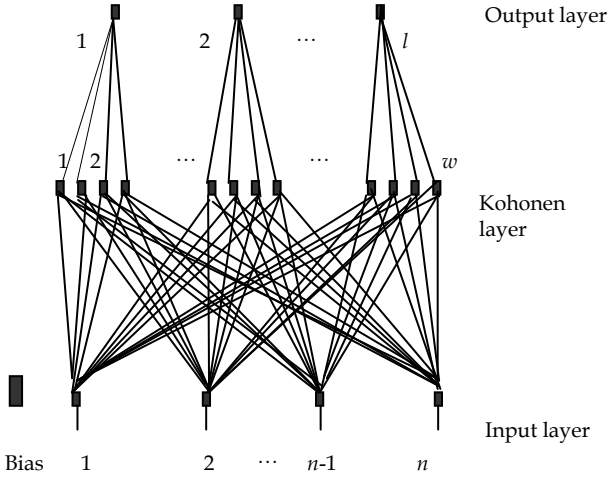


Fig. 35. Topological structure of LVQ neural network

2. Parameter optimization method of neural network

In order to choose best parameters for neural network, traditional methods are all decided by experience, but in fact, these parameters are not optimized parameters, a new parameter optimization method of neural network is put up here, and it can settle the problem perfectly.

The new parameter optimization method of neural network is aiming at finding inner parameters for describing practical problem based on small-samples(Tang, B., 2003), so that the parameters can meet the solution of the problem, and it makes possible for avoiding blindly finding possible parameters of neural network. Figure 36 shows the flowchart of the new parameter optimization method based on small-samples.

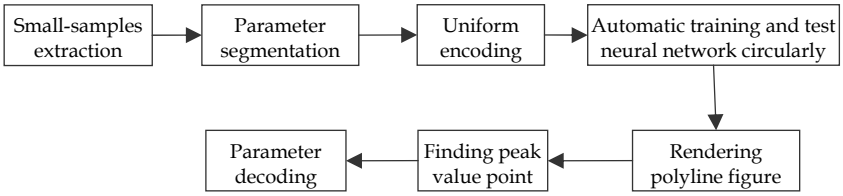


Fig. 36. Flowchart of parameter optimization algorithm

1) Extraction of small-samples

In order to make parameters reflect the essence of practical problem, a certain proportion of samples is extracted from all the samples to form a new sample aggregation, named small-samples, and let the proportion be η. We supposed that the original sample aggregation is P_0 as equation (49).

$$P_0 = \{X_1, X_2, \cdots, X_n\} \tag{49}$$

The amount of every item inside the original sample aggregation is $p_0^1, p_0^2, \cdots, p_0^n$ respectively. At the same time, we supposed that the new sample aggregation is P_s as equation (50)

$$P_s = \left\{ X_1^{'}, X_2^{'}, \cdots, X_n^{'} \right\} \tag{50}$$

Its amount is N, therefore the extraction rule is defined as following:
Extracting every class of samples by its proportion from sample library, and defining the number of every sample as p_s^i, therefore it can describe as equation (51).

$$p_s^i = \frac{p_0^i}{\sum_{j=1}^{n} p_0^j} N \qquad i = 1, 2 \cdots n \tag{51}$$

Consequently the distribution law of extracted new small-samples P_s is the same as original samples P_0, in another word that the extracted small-samples can stand for the original samples to some extent, and the parameters of neural network selected under such small-samples can express the map relationship of the neural network under the whole original samples. Such small-samples selected by the method are so called proportional samples. A group of training small-samples p_{s1} and testing small-samples p_{s2} can be extracted in the practical application.

2) Parameter segmentation and uniform encoding
When applying it to practical engineering project, firstly it is necessary to know all the parameters of neural network, and choose the parameters which are related with the essence of problem among all the parameters. Now defining the number of selected parameters as p, and all parameters are independent and have no relation with samples theoretically, in another word that the selected parameter variables can reflect the essence characters of the practical problem. After determined parameters which need be optimized, it is time to segment all the parameters under a scope of experience values or theoretical values.
Supposed that all these p parameters are divided into such parts respectively as $C_1, C_2, \cdots, C_p$, then combining all the parameters under unrepeated rule, and there are total M combinations, it can be described as equation (52):

$$M = \prod_{i=1}^{P} C_{C_i}^1 \tag{52}$$

What come on next? We must encoding the parameters for every combination, and assign a number to it as 1, 2, ⋯, M. There are mainly 2 steps in this stage as following:
Firstly, determining the order of every parameter and fixing its position from the front to the back. Figure 37 shows the exact order of every parameter in the same group.

P_1	P_2	⋯	P_M

Fig. 37. Exact order of every parameter

Secondly, a uniform serial number is given to every combination. When a specific number is given to a front parameter, the back parameter will traverse all possible values, and the number will increase 1 step by step till all M possibilities is finished, and the encoding finished.

3) Automatic training circularly and plotting polyline figure

After encoding all the parameters, an automatic program must be developed, and every group of parameters is used as parameters of neural network respectively to do training and testing. The procedure can be described as followings:

a. Take out one specific group of parameters as parameters of neural network.

b. Training the neural network with the training small-samples of P_{s1}.

c. When the training finished, saved the trained neural network model for testing.

d. Testing the neural network with testing small-samples p_{s2}.

e. Making a statistic of its recognition rate.

f. Repeating step "a" to "e" again till all *M* group of parameters are traversed for one time.

g. Using the *M* recognition rate values under different *M* group of parameters of neural network to plot a polyline figure on a coordinate system.

h. Finding the peak recognition rate value of the figure, and writing down its related group number of parameters which is selected as the final parameters of neural network under the original samples.

4. Application and experiments

Table 2 is samples distribution table which are all acquired by surface defect online inspection system of cold rolled strips from locale of certain production line in China. From the table, it can be seen that there are total 12 classes of defects of cold rolled strips, and there are total 6360 original training samples and 3180 original testing samples.

According the extraction rule of small-samples of parameter selection method described in preceding part, a training small-samples aggregation P_{s1} and a testing small-samples aggregation P_{s2} will be get from original training samples and testing samples under a given extraction proportion of 8% and 4% respectively.

Item	Scratches	coil Breaks	point sticks	Feathers	white spots	roll Imprints	edge Foldings	rusts	Emulsion marks	Orange peels	edge Cracks	other	sum
Number of training samples	1223	1398	1203	1153	463	295	193	121	89	80	70	71	6360
Number of testing samples	612	699	602	577	232	147	96	61	45	40	35	35	3180

Table 2. Samples distribution table of system

For LVQ3 neural network, the number of neurons in compete layer must be determined firstly, and according the system, it is assigned 106. Secondly, make sure that there are three parameters which are related with the essence of described system, they are initial learn rate $\eta(0)$, relative learn rate ε and window width m. According to the experience of LVQ3 neural network, the experience scope of all the three parameters can be described as equation (53).

$$\begin{cases} 0.01 \le \eta(0) \le 0.1 \\ 0.05 \le \varepsilon \le 0.3 \\ 0.1 \le m \le 0.5 \end{cases} \tag{53}$$

After experiments, polyline will be plotted under coordinates of recognition rate and parameter combination number as shown in Figure 38. It can be seen that the coordinate (48, 96.2%) stands for the peak point in the figure, it means that No. 48 parameter combination can describe the practical problem better than other parameter combination of LVQ3 neural network, and the test result can reach the highest recognition rate of 96.2%. By decoding we can get the parameters of LVQ3 neural network as Table 3.

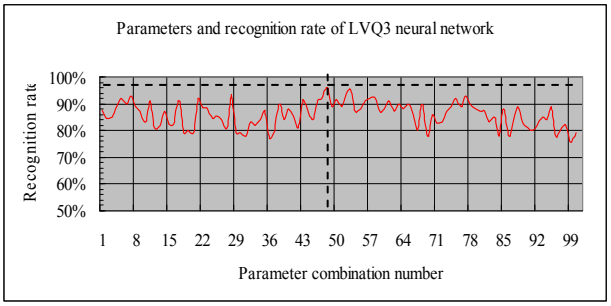


Fig. 38. Curve of optimized parameter selection of LVQ3 neural network

It can be seen from Table 4 that the recognition rate under No.1 group of parameters reaches to 89.69%, and it is the highest in table, but under other 7 group of parameters, the recognition rates are all lower than it. Therefore, the experiments here fully proved that using the optimized parameters of neural network by the new parameter selection method as the parameters of neural network, the classification information can be expressed more accurately, a higher recognition rate can be get when applying to recognize surface defect of cold rolled strips.

Item	Neuron number of input layer	Neuron number of competion layer	Neuron number of output layer	Initial learn rate $\eta(0)$	Relative learn rate ε	Window width m	Iteration times
Value	22	1276	12	0.05	0.1	0.3	51040

Table 3. Parameters of LVQ3 neural network

No.	Training condition			Testing result		
	$\eta(0)$	ε	m	Number of correct recognition	Recognition rate	Difference with No.1
1	0.05	0.1	0.3	2852	89.69%	0.00%
2	0.01	0.05	0.1	2493	78.40%	11.29%
3	0.1	0.3	0.5	2598	81.70%	7.99%
4	0.08	0.2	0.4	2530	79.56%	10.13%
5	0.03	0.3	0.2	2645	83.18%	6.51%
6	0.06	0.08	0.3	2791	87.77%	1.92%
7	0.01	0.3	0.1	2609	82.04%	7.65%
8	0.05	0.1	0.5	2755	86.64%	3.05%

Table 4. Recognition results under different neural network parameters

5. Conclusion

An online surface inspection system of cold rolled strips is researched based on multimedia technology such as image process technology and pattern recognition technology, and it is applied to a production line of a certain iron & steel corporation in China, which includes cameras, illuminations, parallel computer systems and other correlative parts, and can detect and recognize over 10 main defects of cold rolled strips real time. The application effects show that the system can meet most requirements of corporation, and it can detect defects stably for long time. With the improvement of algorithm, some difficulties will be overcome, and the system will become wider use.

6. References

Sugimoto, T.; Kawaguchi, T.(1998). Development of a Surface Defect Inspection System Using Radiant Light from Steel Products in a Hot Rolling Line, *IEEE Transactions on Instrumentation and Measurement*, 47, 409-416

Hannes, L.; Weuster-Botz, Dirk, (2006). Genetic algorithm for multi-objective experimental optimization, *Bioprocess and Biosystems Engineering*, vol.29, No.5-6, 385-390

Kim, S. M.; Lee, Y. C.; Lee, S. C., (2006). Vision Based Automatic Inspection System for Nuts Welded on the Support Hinge, *SICE-ICASE International Joint Conference*, pp. 1508-1512, 2006.10

Tang, B.; Luo, S. W.; Huang, H.(2003). High performance face recognition system by creating virtual sample, *Proceedings of the 2003 International Conference on Neural Networks and Signal Processing*, pp. 972 – 975, 2003. 2

Chu, E.; George, A.(2000). Inside the FFT black box: serial and parallel fast Fourier transform algorithms, In: *CRC Press*, Boca Raton, Fla.

Kohonen, T. (1992) New Developments of Learning Vector Quantization and the Self-Organizing Map, *Symposium on Neural Networks Osaka*, Alliances and Perspectives in Senri

13

Error Resilient Video Coding Techniques Based on the Minimization of End-to-End Distortions

Wen-Nung Lie[1] and Zhi-Wei Gao[2]
[1]*National Chung Cheng University, Chia-Yi, Taiwan*
[2]*TECO Group Research Institute, Taipei, Taiwan*

1. Introduction

Due to successful development of video compression techniques, the huge amount of redundancies contained in raw videos can be removed effectively. This makes it possible to transmit videos over bandwidth-limited channels. The applications can be video conferencing, distance learning, streaming videos on Internet and mobile phones, digital high quality TV, …, etc.

Generally speaking, redundancies in videos can be categorized into different types: spatial, temporal and statistical redundancies. Operations of discrete cosine transform (DCT) and quantization are used to remove the spatial redundancy and motion estimation and compensation (ME/MC) are used to remove the temporal redundancy. Entropy coder is, on the other hand, for removing the last type of redundancy.

However, highly compressed videos are often fragile to noises. Once compressed video bit streams encounter any kinds of errors in transmission, the quality of the reconstructed videos at decoder side degrades seriously. The reason is that error propagation in spatial or temporal direction occurs when the decoder decodes erroneous bit streams to reconstruct videos. In order to enhance the quality of the decoded videos at decoder side, two common techniques are often adopted: one is to generate robust compressed video bit streams at encoder side, known as the "error resilience"; the other is to conceal errors in the reconstructed videos at decoder side, known as the "error concealment". Both techniques are capable of substantially improving the quality of the decoded videos in error-prone transmission environments.

This chapter will focus on the issue of error resilient video coding. The main concept of error resilient coding is to increase the robustness of the compressed videos at the expense of extra bit rates; that is, inserting redundancies important to video error recovery. The coding efficiency (i.e., PSNR/bit-rate) unavoidably lowered down, whereas a lower PSNR degradation at decoder side can be achieved in case of severe channel errors. Researchers often faced a problem of how to schedule the overall bit resources such that error resiliency capability can be maximized.

Among the algorithms developed for video error resiliency, end-to-end distortion, which measures the difference between the raw video data and that finally obtained before display at decoder side (possibly with channel errors and according error concealment), was recently popularly adopted as the criterion for optimization (minimization). Here, in this book chapter, we first propose an algorithm of error-resilient motion estimation and mode decision by considering end-to-end distortions for H.264/AVC standard. Then, this algorithm is extended for the enhancement of H.264-based multi-hypothesis coding (MHC) at a given hypothesis-weighting vector, which was traditionally proposed with its good rate-distortion performance in noise-prone channels. Finally, based on the availability of motion vectors, an adaptive hypothesis-weighting algorithm is proposed to make error resiliency adaptive to video contents, frame by frame.

2. Modelling of end-to-end distortions

End-to-end distortion, the distortion between the raw data and the one reconstructed at decoder side (possibly incurred with channel errors), has been adopted as an optimized criterion in applications such as intra/inter mode decision (Chang et al., 2005; Cote & Kosentini, 1999; Leontaris & Cosman, 2004; Zhan et al., 2000) and motion estimation (ME) (Harmanci & Tekal, 2005; Wiegand et al., 2000; Yang & Rose, 2005). We first model the end-to-end distortions to include (He et al, 2002): source distortion D_s, incurred by n_q, and channel distortion D_c, incurred by n_c. Here, n_q is related to the quantization noise and n_c to the incompleteness of error concealment and motion compensation (or, error propagation). A pictorial illustration of our model is depicted in Fig.1, where the subscript n is the frame index, f_n represents the original video signal, $\hat{f}_n$ represents the encoded video (i.e., decoded video without errors), and $\tilde{f}_n$ represents the video reconstructed at the decoder side in presence of channel noise n_c.

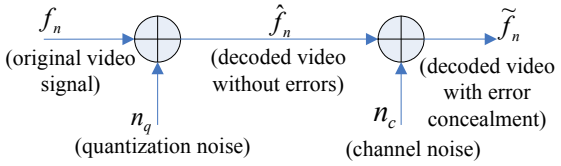


Fig. 1. The modelling of end-to-end distortion.

Via the above model, it is ready to observe that the end-to-end distortion $D_{end\text{-}to\text{-}end}$ can be formulated as

$$\begin{aligned} D_{end-to-end} &= E\left\{(f_n - \tilde{f}_n)^2\right\} \\ &\cong E\left\{(f_n - \hat{f}_n)^2 + (\hat{f}_n - \tilde{f}_n)^2\right\} = D_s + D_c \end{aligned} \tag{1}$$

where $D_s = E\left\{(f_n - \hat{f}_n)^2\right\}$ and $D_c = E\left\{(\hat{f}_n - \tilde{f}_n)^2\right\}$. (2)

Obviously, a cross product term is ignored in deriving Eq.(1). This model was first verified on the H.263 test platform (He et al., 2002) and then proved to have an averaged deviation of only 0.863% on H.264/AVC platform (Lie et al, 2006). Based on this fact, $D_{end\text{-}to\text{-}end}$ can then be reduced by minimizing D_s and D_c separately.

However, reducing D_s may conflict with reducing D_c at a specified constant bit-rate (since a lower D_s causes a higher source bit-rate and hence, a lower channel bit-rate and larger D_c). Therefore, the original problem of minimizing end-to-end distortions becomes a typical problem of multi-objective optimization.

3. Minimizing end-to-end distortions for error resilient motion estimation (ERME)

One criterion to optimize the error resiliency of the transmitted videos in error-prone environments is to consider the end-to-end distortions under a given bit rate budget. The difficulty in estimating end-to-end distotions is mainly on knowing well the effect of error propagation caused by the loss of motion vectors (MVs) and the strategy of error concealment at decoder. Generally, the applicable methods up to now can be roughly categorized into simulation-based and model-based.

For simulation-based methods, a set of error patterns are generated according to channel conditions (e.g., signal-to-noise ratio (SNR), bit error rate (BER), or packet loss rate (PLR)) and coding parameters, such as period of resynchronization, intra-refreshing rate, etc., are tuned accordingly to minimize the end-to-end distortions. The drawback of the simulation-based approaches is the high computing load, which makes them only suitable for off-line video streaming applications (e.g., video transcoding (Reyes et al., 2000; Xia et al., 2004). On the other hand, for model-fitting approaches (He et al., 2002; Stuhlmuller, et al., 2000; Eisenberg et al., 2006), effects of error propagation are described in terms of mathematical models. Model parameters are then determined in accordance with the video contents by collecting a small set of training data. Hence, evaluation of coding parameters becomes a process of model interpolation, which would reduce the computational complexity substantially.

To suppress error propagation without sacrificing much coding efficiency, methods of searching MVs in criteria other than the traditional least SAD (sum of absolute difference) have ever been proposed. For example, Wiegand et al. (2000) adopted the end-to-end distortion as the optimizing criterion in searching MVs. This kind of algorithms could be classified as the Error Resilient Motion Estimation (ERME). However, their method generates/simulates the error patterns and analyzes the possible error propagation by constructing trees whose sizes will grow substantially for a large GOP (group of picture) size. This restricts their practical applications, in views of both memory requirements and computing loads. We need an effective but simple procedure to estimate the end-to-end distortions during the computationally intensive ME process.

An alternative method in estimating end-to-end distortions, well known as the ROPE (Recursive Optimal Per-pixel Estimate) (Zhan et al, 2000), was developed for intra/inter mode decision of each coded frame. They differently modeled error propagation caused by lost MVs by exploiting a statistical approach. Following that, the ROPE algorithm was

applied to motion estimation (Yang & Rose, 2005; Harmanci & Tekal, 2005) for increasing the performance in error resiliency. Essentially, the end-to-end distortions are decomposed into three sources: error propagation, error concealment, and quantization. To estimate the quantization errors, a series of processes like DCT/IDCT, quantization and inverse quantization, should be performed for each candidate under evaluation. This causes no problems in mode decision where only two modes (intra/inter) are considered, but certainly incurs prohibitively high computational loads for motion estimation (Yang & Rose, 2005) (normally over one thousand of MV candidates).

Extending the prior concept and avoiding the high computational complexity in estimating the end-to-end distortions, a new optimization algorithm is proposed for ERME in this chapter. The new optimizing criterion of our proposed algorithm consists of two conflicting objective functions; specifically, one of them relates to the enhancement of error resiliency and the other to the increase of coding efficiency. Hence, finding a MV that minimizes this criterion becomes a problem known as the multi-objective optimization (Ringuest, 1992). In this situation, a solution that minimizes all objective functions simultaneously does not always exist when the objective functions conflict with each other. Instead, a constrained optimization method can be developed to find a solution that compromises among these conflicting objective functions. Applying this concept, MVs thus found is capable of compromising between error resiliency and coding efficiency. Moreover, since the computing procedure of our proposed algorithm does not include the terms relating to the quantization errors, the computational complexity is not as high as that of Yang & Rose (2005).

To derive error resilient MVs, the channel distortion D_c should be modelled first. The search criterion is defined below:

$$(\Delta x^*, \Delta y^*) = \arg \min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{B} E\left\{ \left(\hat{f}_n^i(\Delta x, \Delta y) - \tilde{f}_n^i(\Delta x, \Delta y) \right)^2 \right\} \right\} \tag{3}$$

where the subscript n is the frame index, Δx and Δy are the horizontal and vertical components of the MV, S is the feasible set for motion vectors, B is the number of pixels in a block, both $\hat{f}_n^i(\Delta x, \Delta y)$ and $\tilde{f}_n^i(\Delta x, \Delta y)$ represent the i^{th} pixel of a block motion-compensated by using MV= $(\Delta x, \Delta y)$, and $E\{.\}$ is the expectation operator. Note that $\hat{f}_n^i(\Delta x, \Delta y)$ represents the pixel value correctly decoded, wheras $\tilde{f}_n^i(\Delta x, \Delta y)$ is the pixel value obtained by considering erroneous reconstruction.

Equation (3) is actually not suitable for practical applications since a series of DCT/IDCT and quantization processes is required in computing $\hat{f}_n^i(\Delta x, \Delta y)$ and $\tilde{f}_n^i(\Delta x, \Delta y)$. Approximation of computations is necessary to make it mathematically tractable. Here, Eq.(3) is modified by changing the squared term into an absolute term:

$$(\Delta x^*, \Delta y^*) = \arg \min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{B} E\left\{ \left| \hat{f}_n^i(\Delta x, \Delta y) - \tilde{f}_n^i(\Delta x, \Delta y) \right| \right\} \right\} \tag{4}$$

Based on the inequality $E\{|\hat{x}-\tilde{x}|\} \geq |E\{\hat{x}-\tilde{x}\}|$, Eq.(5) will hold.

$$\begin{aligned} &\min_{(\Delta x,\Delta y)\in S}\left\{\sum_{i=1}^{B} E\left\{\left|\hat{f}_n^i(\Delta x,\Delta y)-\tilde{f}_n^i(\Delta x,\Delta y)\right|\right\}\right\} \\ &\geq \min_{(\Delta x,\Delta y)\in S}\left\{\sum_{i=1}^{B}\left|E\left\{\hat{f}_n^i(\Delta x,\Delta y)-\tilde{f}_n^i(\Delta x,\Delta y)\right\}\right|\right\} \\ &= \min_{(\Delta x,\Delta y)\in S}\left\{\sum_{i=1}^{B}\left|\hat{f}_n^i(\Delta x,\Delta y)-E\left\{\tilde{f}_n^i(\Delta x,\Delta y)\right\}\right|\right\} \end{aligned} \tag{5}$$

Note that the first term in summation is considered to be deterministic with respect to the $E\{.\}$ operator, due to its independence from the channel conditions. The term $E\left\{\tilde{f}_n^i(\Delta x,\Delta y)\right\}$ in Eq.(5) can be easily estimated by (Zhan et al., 2000):

$$E\left\{\tilde{f}_n^i(\Delta x,\Delta y)\right\} = (1-p_e)(E\left\{\tilde{f}_{n-\alpha}^j\right\} + r_n^i(\Delta x,\Delta y)) + p_e \cdot E\left\{\tilde{f}_{n-1}^i\right\} \tag{6}$$

where p_e is the error probability of a considered pixel, $r_n^i(\Delta x,\Delta y)$ is the residual produced after motion prediction (with MV= $(\Delta x,\Delta y)$), $E\left\{\tilde{f}_{n-\alpha}^j\right\}$ is the pixel value compensated from the j^{th} pixel (pointed to by MV= $(\Delta x,\Delta y)$) of the $(n\text{-}\alpha)^{th}$ frame, α is a positive number (accounting for the long-term memory prediction in H.264/AVC), and $E\left\{\tilde{f}_{n-1}^i\right\}$ is the pixel value recovered by adopting zero-motion as the scheme of error concealment.

According to the coding principle, $\hat{f}_n^i(\Delta x,\Delta y)$ can be expressed as:

$$\hat{f}_n^i(\Delta x,\Delta y) = \hat{f}_{n-\alpha}^j + r_n^i(\Delta x,\Delta y), \tag{7}$$

where $\hat{f}_{n-\alpha}^j$ is the pixel value motion-compensated from the j-th pixel of the $(n\text{-}\alpha)^{th}$ frame. Substituting Eqs.(6) and (7) into Eq.(5), we change the optimization problem in Eq. (4) to become:

$$\begin{aligned} &(\Delta x^*,\Delta y^*) \\ &= \arg\min_{(\Delta x,\Delta y)\in S}\left\{\sum_{i=1}^{B}\left|\begin{matrix}\left(\hat{f}_{n-\alpha}^j + r_n^i(\Delta x,\Delta y)\right) - p_e E\left\{\tilde{f}_{n-1}^i\right\} \\ -(1-p_e)\left(E\left\{\tilde{f}_{n-\alpha}^j\right\} + r_n^i(\Delta x,\Delta y)\right)\end{matrix}\right|\right\} \end{aligned} \tag{8}$$

Equation (8) is actually not a good criterion, due to its prohibitively high complexity in computing $r_n^i(\Delta x,\Delta y)$. To yield $r_n^i(\Delta x,\Delta y)$, the processes of DCT, quantization, and IDCT need to be performed for each MV candidate $(\Delta x,\Delta y)$. This is also the reason why algorithms provided in Yang & Rose (2005) and Harmanci & Tekal (2005) are impractical in view of computing complexity. Equation (8) can be rewritten to be

$$
\begin{aligned}
&(\Delta x^*, \Delta y^*) \\
&= \arg \min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{B} \left| \left(\hat{f}_{n-\alpha}^{j} \right) - \left((1-p_e) \cdot E\left\{ \tilde{f}_{n-\alpha}^{j} \right\} + p_e \cdot E\left\{ \tilde{f}_{n-1}^{i} \right\} \right) \right| \right. \\
&\qquad \left. + \left(p_e \cdot r_n^i(\Delta x, \Delta y) \right) \right\}
\end{aligned} \tag{9}
$$

Considering that $r_n^i(\Delta x, \Delta y)$ is the prediction residual (expectedly smaller than the pixel reconstructions $\hat{f}_{n-\alpha}^{j}$, $E\left\{\tilde{f}_{n-\alpha}^{j}\right\}$, and $E\left\{\tilde{f}_{n-1}^{i}\right\}$ and $p_e < 1 - p_e < 1.0$ (for $p_e < 0.5$), the term $p_e \cdot r_n^i(\Delta x, \Delta y)$ can be ignored, with respect to the other three terms, reducing Eq.(9) into Eq.(10):

$$
\begin{aligned}
&(\Delta x^*, \Delta y^*) \\
&= \arg \min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{B} \left| \left(\hat{f}_{n-\alpha}^{j} \right) - \left((1-p_e) \cdot E\left\{ \tilde{f}_{n-\alpha}^{j} \right\} + p_e \cdot E\left\{ \tilde{f}_{n-1}^{i} \right\} \right) \right| \right\}
\end{aligned} \tag{10}
$$

Note that, the ignorance of $p_e \cdot r_n^i(\Delta x, \Delta y)$ eases the computation of channel distortions significantly. In Eq.(10), the first term $\hat{f}_{n-\alpha}^{j}$ is readily available when encoding the n^{th} frame, while the second term (also called the first moment of $\tilde{f}_{n-\alpha}^{j}$) can be derived by using the technique proposed in Zhan et al. (2000). According to Eq.(6), the first moment of $\tilde{f}_n^i$ (i.e., $E\left\{\tilde{f}_n^i\right\}$) can be recursively updated after its motion vector $(\Delta x, \Delta y)^*$ and residual r_n^i are figured out. This guarantees the availability of $E\left\{\tilde{f}_{n-\alpha}^{j}\right\}$ and $E\left\{\tilde{f}_{n-1}^{i}\right\}$ on evaluating Eq.(10) for the n^{th} frame. Clearly, the extra computations required in computing Eq.(10) come from the updating of $E\left\{\tilde{f}_n^i\right\}$.

The quantity to be optimized in Eq.(10) approximates the channel distortion described in Eq.(3). Hence, we can have a constraint on the source distortion D_s when optimizing the channel distortion D_c. That is,

$$
\begin{aligned}
&\min_{(\Delta x, \Delta y) \in S} \left\{ \sum_{i=1}^{B} \left| \left(\hat{f}_{n-\alpha}^{j} \right) - \left((1-p_e) \cdot E\left\{ \tilde{f}_{n-\alpha}^{j} \right\} + p_e \cdot E\left\{ \tilde{f}_{n-1}^{i} \right\} \right) \right| \right\} \\
&\qquad \text{subject to} \quad D_s^{\min} < \sum_{i=1}^{B} \left\{ (f_n^i - \hat{f}_n^i)^2 \right\} \le \sigma_1^2
\end{aligned} \tag{11}
$$

where $D_s^{\min}$ is the achievable lower bound when considering to optimize D_s only and σ_1^2 is a selected threshold. It is known that constraining the source distortion is equivalent to limiting the quantization noise, i.e., keeping the quantization parameter (QP) below a value. Accordingly, the motion-prediction residues should be kept low to control the resulting bit rate to a targeted value (since a small QP will increase the bit rate). Hence, we change the above inequality condition to

$$\gamma^{\min} < \sum_{i=1}^{B} \left(f_n^i - \hat{f}_{n-\alpha}^{j(\boldsymbol{w},\alpha)} \right)^2 \le Thd1 \tag{12}$$

where $(\boldsymbol{w},\alpha)$ stands for a MV candidate $\boldsymbol{w}$ that refers to the $(n\text{-}\alpha)^{\text{th}}$ frame, $j(\boldsymbol{w},\alpha)$ represents the pixel j pointed to by $\boldsymbol{w}$ in frame-$(n\text{-}\alpha)$, $f_n^i - \hat{f}_{n-\alpha}^{j(\boldsymbol{w},\alpha)}$ is the motion-prediction residue for pixel i of the frame n, $\gamma^{\min}$ represents the smallest residual power that can be achievable, and *Thd1* is a selected threshold. This change of constraint obviously eases the evaluation process significantly since the computation of $\hat{f}_{n-\alpha}^{j(\boldsymbol{w},\alpha)}$ is in no need of IDCT for a given $(\boldsymbol{w},\alpha)$, while the computation of $\hat{f}_n^i$ requires r_n^i (Eq.(7)), which needs DCT, quantization, and IDCT for each MV candidate $(\boldsymbol{w},\alpha)$.

Replacing the constraint of Eq.(11) with the inequality in Eq.(12), we have the following constrained optimization problem:

$$\underset{(\boldsymbol{w},\alpha)}{Min}\{EP_{(\boldsymbol{w},\alpha)}\} \qquad \text{subject to} \quad CR_{(\boldsymbol{w},\alpha)} \le Thd1 \tag{13}$$

where

$$EP_{(\boldsymbol{w},\alpha)} = \sum_{i=1}^{B} \left| \hat{f}_{n-\alpha}^{j(\boldsymbol{w},\alpha)} - \left((1-p_e) \cdot E\left\{ \tilde{f}_{n-\alpha}^{j(\boldsymbol{w},\alpha)} \right\} + p_e \cdot E\left\{ \tilde{f}_{n-1}^{i} \right\} \right) \right| \tag{14}$$

$$CR_{(\boldsymbol{w},\alpha)} = \sum_{i=1}^{B} \left(f_n^i - \hat{f}_{n-\alpha}^{j(\boldsymbol{w},\alpha)} \right)^2 \tag{15}$$

In another viewpoint, $EP_{(\boldsymbol{w},\alpha)}$ measures the level of error propagation caused by channel distortion and $CR_{(\boldsymbol{w},\alpha)}$ reflects the power of the motion-prediction residuals relating to source distortion, for a given $(\boldsymbol{w},\alpha)$. By adjusting *Thd1*, different levels of compromise between $EP_{(\boldsymbol{w},\alpha)}$ and $CR_{(\boldsymbol{w},\alpha)}$ can be achieved.

Traditionally, the optimal solution of Eq.(13) can be found via a full or a fast search on all possible $(\boldsymbol{w},\alpha)$'s. Note that our algorithm would not change the nature of full/fast search in the traditional ME process, but to provide a more proper criterion in selecting MVs that are expected to achieve better compromise between error resiliency and coding efficiency.

4. Minimizing end-to-end distortions for error resilient mode decision (ERMD)

In comparison with H.263 and MPEG-1/2/4 standards, mode decision (MD) for both intra and inter-coded frames (here, we only focus on the inter-coding case due to its close relation to the ME topic) is one of the most distinctive and evolving features that make significant progress in coding efficiency. Essentially, a MB is divided into sub-blocks of variable sizes (e.g., 16×16, 16×8, 8×16, 8×8, 4×8, 8×4, and 4×4 pixels), each of which is associated with an estimated MV (via the ERME method previously discussed). A cost is then measured with each combination (or, called a mode) of block partition for a MB. Mode decision is thus to

determine a partition, together with the estimation of associated MVs, that minimizes a selected cost function for each MB. The traditional cost function in H.264/AVC is based on the so-called SATD (Sum of Absolute Transform Difference). It is shown (Stuhlmuller et al., 2000) that for error-free transmission, mode selection in a Lagrangian rate-distortion framework is capable of enhancing the coding efficiency effectively.

Essentially, different goals would require different optimizing criterion or cost functions. It is straightforward to motivate us that both ME and MD based on modified criteria would enhance error resiliency for error-prone transmission.

In H.264/AVC, the cost (i.e., SATD) defined for Lagrangian optimization framework is related to the residuals after motion compensation, or, the difference between the motion-predicted MB and the original data. Instead of that, the end-to-end distortion defined in Eq. (1) is incorporated into the algorithm of Lagrangian optimization. More precisely, after finding MVs for each mode candidate by using the ERME algorithm, the following optimization problem is to be solved:

$$\min_{m\in\boldsymbol{M}} D_s(m) + D_c(m) + \lambda R(m), \tag{16}$$

where $\boldsymbol{M}$ is the set of mode candidates m's allowable in H.264/AVC, D_s and D_c represent the source and channel distortion as defined in Eq.(1), R is the bit rate needed to encode a MB given m, and λ is a Lagrangian multiplier.

D_s can be computed directly from the original data and the reconstructed data at local decoder. To evaluate D_c, cases of inter- and intra-coded MBs are to be separately discussed.

For intra-coded MBs, channel distortion $D_{C,i}^{I}$ totally results from incompleteness of error concealment. Equation (17) below formulates this observation:

$$D_{C,i}^{I,n} = p_e \cdot E\left\{\left(\hat{f}_n^i - \tilde{f}_{n-1}^i\right)^2\right\} \tag{17}$$

where i is the pixel index in an MB, n is the frame index, p_e is the error probability relating to the transmission environment, and $E\{.\}$ is the expectation operator. Equation (17) can be further arranged to yield Eq.(18):

$$\begin{aligned} D_{C,i}^{I,n} &= p_e \cdot E\left\{\left(\hat{f}_n^i - \hat{f}_{n-1}^i + \hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)^2\right\} \\ &\le p_e \cdot E\left\{\left(\left|\hat{f}_n^i - \hat{f}_{n-1}^i\right| + \left|\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right|\right)^2\right\} \\ &\le p_e \cdot E\left\{\left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2\right\} + 2p_e \cdot E\left\{\left|\hat{f}_n^i - \hat{f}_{n-1}^i\right|\left|\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right|\right\} \\ &\quad + p_e \cdot E\left\{\left(\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)^2\right\} \end{aligned} \tag{18}$$

Note that $\left|\hat{f}_n^i - \hat{f}_{n-1}^i\right|$ stands for the frame difference after encoding, which is independent of the channel condition (hence, deterministic with respect to the $E\{.\}$ operator). Hence, Eq. (18) is reduced to

$$\begin{aligned} D_{C,i}^{I,n} &\le p_e \cdot \left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2 + 2p_e \cdot \left|\hat{f}_n^i - \hat{f}_{n-1}^i\right| \cdot \left|\hat{f}_{n-1}^i - E\left\{\tilde{f}_{n-1}^i\right\}\right| + p_e \cdot E\left\{\left(\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)^2\right\} \\ &= p_e \cdot \left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2 + 2p_e \cdot \left|\hat{f}_n^i - \hat{f}_{n-1}^i\right| \cdot \left|\hat{f}_{n-1}^i - E\left\{\tilde{f}_{n-1}^i\right\}\right| + p_e \cdot D_{C,i}^{n-1} \end{aligned}, \tag{19}$$

where $D_{C,i}^{n-1}$ represents channel distortion of the pixel i in frame (n-1).

For inter-coded MBs, distortions resulting from error propagation should be taken into consideration. First, let us consider the case in which MVs and residuals $\hat{e}_n^i$ are received correctly. In this case, the decoded pixel $\tilde{f}_n^i$ will be

$$\tilde{f}_n^i = \hat{e}_n^i + \tilde{f}_{n-\alpha}^k, \tag{20}$$

where $\tilde{f}_{n-\alpha}^k$ stands for the pixel value compensated from the k-th pixel of frame n-a. If errors occur, the error concealment procedure will replace the erroneous MB $\tilde{f}_n^i$ with $\tilde{f}_{n-1}^i$ (zero-motion recovery is assumed). Combining this observation with Eq.(20), channel distortion of a pixel i in an inter-coded MB can be formulated as

$$D_{C,i}^{P,n} = p_e \cdot E\left\{\left(\hat{f}_n^i - \tilde{f}_{n-1}^i\right)^2\right\} + (1-p_e) \cdot E\left\{\left[\hat{f}_n^i - \left(\hat{e}_n^i + \tilde{f}_{n-\alpha}^k\right)\right]^2\right\}. \tag{21}$$

Similar to the treatment of the term in Eq. (17), Eq. (21) can be arranged to yield

$$\begin{aligned} D_{C,i}^{P,n} &\le p_e \cdot E\left\{\left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2\right\} + 2p_e \cdot \left|\hat{f}_n^i - \hat{f}_{n-1}^i\right| \cdot \left|\hat{f}_{n-1}^i - E\left\{\tilde{f}_{n-1}^i\right\}\right| \\ &\quad + p_e \cdot E\left\{\left(\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)^2\right\} + (1-p_e) \cdot E\left\{\left(\hat{f}_{n-\alpha}^k - \tilde{f}_{n-\alpha}^k\right)^2\right\} \\ &= p_e \cdot \left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2 + 2p_e \cdot \left|\hat{f}_n^i - \hat{f}_{n-1}^i\right| \cdot \left|\hat{f}_{n-1}^i - E\left\{\tilde{f}_{n-1}^i\right\}\right| + p_e \cdot D_{C,i}^{n-1} + (1-p_e) \cdot D_{C,k}^{n-\alpha} \end{aligned} \tag{22}$$

where $D_{C,k}^{n-\alpha}$ stands for the channel distortion of the k-th pixel in frame (n-a).

Based on Eqs. (19) and (22), we are able to estimate the channel distortion term in Eq. (16) for each considered mode m. When applying Eq. (19) and (22), it is necessary to estimate $E\left\{\tilde{f}_{n-1}^i\right\}$, $D_{C,i}^{n-1}$, and $D_{C,k}^{n-\alpha}$. The technique in Zhan et al. (2000) can be used to recursively calculate $E\left\{\tilde{f}_{n-\alpha}^i\right\}$, $\alpha = 1,2,3,\cdots$. A general formula to evaluate $E\left\{\tilde{f}_n^i\right\}$, in case of intra-coded MBs, is given in Eq. (23), whereas Eq. (24) is adopted in case of inter-coded MBs.

$$E\left\{\tilde{f}_n^i\right\} = (1-p_e) \cdot \hat{f}_n^i + p_e \cdot E\left\{\tilde{f}_{n-1}^i\right\} \tag{23}$$

$$E\left\{\tilde{f}_n^i\right\}=(1-p_e)\cdot\left(\hat{e}_n^i+E\left\{\tilde{f}_{n-\alpha}^k\right\}\right)+p_e\cdot E\left\{\tilde{f}_{n-1}^i\right\} \tag{24}$$

For n=0, we set $E\left\{\tilde{f}_0^i\right\}=\hat{f}_0^i$, i.e., no channel errors are assumed. On the other hand, the term $D_{C,i}^{n-1}$, and $D_{C,k}^{n-\alpha}$ are available when processing frame n, they are computed by using Eq. (19) and Eq. (22), depending on the encoding type (I or P) of the MB considered.

5. Multi-hypothesis coding based on minimization of end-to-end distortions

The so-called Multi-Hypothesis Coding (MHC) was proposed to find out more than one motion compensated MB, called hypothesis, from different reference frames and combine these hypotheses via weighting coefficients to form a predicted MB. It is verified (Sullivan, 1993; Flierl et al., 1998; Flierl et al., 2000) that by choosing the number of hypotheses and the weighting coefficients, the multi-hypothesis technique improves coding efficiency further when compared with the single hypothesis technique. Theoretical discussions about the rate-distortion performance of the multi-hypothesis technique can be found in literature (Flierl et al., 2002; Girod, 2000).

Error resiliency property of the multi-hypothesis coding technique has also been discussed (Lin & Wang, 2002; Kung et al., 2006), where the effect of temporal error propagation after burst errors is modeled. Specifically, in Kung et al. (2006), the proposed model is applied at encoder to decide determine the hypothesis-weighting to minimize propagation errors. However, their model restricted to a single burst error, which is not feasible in practical situation.

In this chapter, we try to apply the technique of end-to-end distortion optimization for multi-hypothesis video coding to further enhance the robustness of the transmitted video. The application is two folds: 1) finding error resilient MVs for a given set of hypothesis-weighting coefficients, similarly as in Section 3, and 2) adapting hypothesis-weighting coefficients to video contents. Both the above two techniques (ERME for a given hypothesis-weighting vector and adaptive hypothesis-weighting) can be integrated for further enhancing the error resiliency of the transmitted videos.

Rate-distortion theorem tells us that minimization of D_s can be alternatively achieved by minimizing the power of the residual signal. In case of multi-hypothesis coding technique, it is the power of the signal $f-\boldsymbol{h}^T\hat{\boldsymbol{c}}$ that needs to be minimized, where $\hat{\boldsymbol{c}}$ is a column vector composed of N hypotheses, and $\boldsymbol{h}$ is an $N\times 1$ weighting vector. For the channel distortion D_c, it is formulated similarly as in Section 3. Hence, finding MVs that minimize the end-to-end distortions for a given $\boldsymbol{h}$ can be conducted similarly as in Section 3.

In this case, error resilient motion estimation for MHC can be formulated as finding :

$$\begin{aligned}&(\Delta\mathbf{x}^*,\Delta\mathbf{y}^*)=\\&\arg\min_{(\Delta\mathbf{x},\Delta\mathbf{y})\in\mathbf{S}}\left\{\sum_{i=1}^{p}E\left\{\left(\hat{f}_n^i(\Delta\mathbf{x},\Delta\mathbf{y})-\tilde{f}_n^i(\Delta\mathbf{x},\Delta\mathbf{y})\right)^2\right\}\right\},\end{aligned} \tag{25}$$

where **Δx** and **Δy** denote the vectors of x and y components, respectively, of MVs for producing hypotheses (hence, the dimension of **Δx** and **Δy** equals the number of hypotheses, *N*), *i* is the pixel position index, **S** is the feasible region where MVs is evaluated and *p* is the number of pixels in a MB. Notice the similarity between Eq.(3) and Eq.(25). The only difference comes from the fact that $\hat{f}_n$ and $\tilde{f}_n$ are now function of *N* MVs, i.e., $(\mathbf{\Delta x}, \mathbf{\Delta y})_{N\times 2}$, for multi-hypothesis coding.

We will not waste space to derive similar formulas as Eqs.(4)~(6) for ERME of MHC, but go directly to the one similar to Eq.(7). In accordance with the principle of MHC, it implies

$$\hat{f}_n^{\,i}(\mathbf{\Delta x}, \mathbf{\Delta y}) = \sum_{k=1}^{N} h_k \hat{f}_{n-k}^{\,i,j}(\Delta x_k, \Delta y_k) + \hat{r}_n^i(\mathbf{\Delta x}, \mathbf{\Delta y}), \tag{26}$$

where h_k represents the weighting coefficient applicable to the *k*-th hypothesis, $\hat{f}_{n-k}^{\,i,j}(\Delta x_k, \Delta y_k)$ is the hypothesis obtained by using the MV whose *x*- and *y*-components are $(\Delta x_k, \Delta y_k)$ from the prediction frame with time index $n-k$, and $\hat{r}_n^i(\mathbf{\Delta x}, \mathbf{\Delta y})$ is the quantized residual (prediction error). It is assumed that different hypotheses come from different frames. By assuming that each lost or corrupted MB is recovered via zero-motion replacement from the previous frame, $E\{\tilde{f}_n(\mathbf{\Delta x}, \mathbf{\Delta y})\}$ (similarly as in Eq. (4)), with the knowledge of the error probability p_e, can be estimated below (Sun & Reibman, 2001):

$$\begin{aligned} &E\left\{\tilde{f}_n^{\,i}(\mathbf{\Delta x}, \mathbf{\Delta y})\right\} = \\ &\quad (1-p_e)\left\{\left(\sum_{k=1}^{N} h_k E\left\{\tilde{f}_{n-k}^{\,i,j}(\Delta x_k, \Delta y_k)\right\}\right) + \hat{r}_n^i(\mathbf{\Delta x}, \mathbf{\Delta y})\right\} + p_e \cdot E\left\{\tilde{f}_{n-1}^{\,i}\right\} \end{aligned} \tag{27}$$

It should be noted that in Eq. (27), even current residual data is correctively received, the hypothesis prediction source may be erroneous due to error propagation and incompleteness of error concealment. Substituting Eq. (26) and Eq. (27) into Eq (25), we get

$$\begin{aligned} &(\mathbf{\Delta x}^*, \mathbf{\Delta y}^*) \\ &= \arg\min_{(\mathbf{\Delta x}, \mathbf{\Delta y})\in \mathbf{S}} \left\{ \sum_{i=1}^{p} \left| \begin{array}{l} \sum_{k=1}^{N} h_k \hat{f}_{n-k}^{\,i,j}(\Delta x_k, \Delta y_k) + \hat{r}_n^i(\mathbf{\Delta x}, \mathbf{\Delta y}) - p_e E\left\{\tilde{f}_{n-1}^{\,i}\right\} \\ -(1-p_e)\left(\sum_{k=1}^{N} h_k E\left\{\tilde{f}_{n-k}^{\,i,j}(\Delta x_k, \Delta y_k)\right\} + \hat{r}_n^i(\mathbf{\Delta x}, \mathbf{\Delta y})\right) \end{array} \right| \right\} \end{aligned} \tag{28}$$

Again, based on the triangular inequality $|A+B| \le |A| + |B|$ and the fact that all h_k's sum to 1.0, we change Eq. (28) to:

$$\begin{aligned} &(\mathbf{\Delta x}^*, \mathbf{\Delta y}^*) \\ &= \arg\min_{(\mathbf{\Delta x}, \mathbf{\Delta y})\in \mathbf{S}} \left\{ \sum_{i=1}^{p} \left(\sum_{k=1}^{N} h_k \left| \begin{array}{l} \hat{f}_{n-k}^{\,i,j}(\Delta x_k, \Delta y_k) + \hat{r}_n^i(\mathbf{\Delta x}, \mathbf{\Delta y}) \\ -(1-p_e)\left(E\left\{\tilde{f}_{n-k}^{\,i,j}(\Delta x_k, \Delta y_k)\right\} + \hat{r}_n^i(\mathbf{\Delta x}, \mathbf{\Delta y})\right) - p_e E\left\{\tilde{f}_{n-1}^{\,i}\right\} \end{array} \right| \right) \right\} \end{aligned} \tag{29}$$

Note that MV's obtained from Eq. (29) will be sub-optimal with respect to that obtained from Eq. (28) due to a higher end-to-end distortion. However, this arrangement reduces the computing complexity since it divides the original optimization problem into N sub-problems which can be solved individually.

Similarly as in Section 3, considering that the error probability p_e is commonly less than 0.2 and that the magnitude of $r_n^i(\mathbf{\Delta x}, \mathbf{\Delta y})$ is usually smaller than the hypothesis signal $\hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k)$, we ignore the term $p_e \hat{r}_n^i(\mathbf{\Delta x}, \mathbf{\Delta y})$ and rearrange Eq. (29) to be:

$$\begin{aligned}&(\mathbf{\Delta x}^*, \mathbf{\Delta y}^*)\\ &= \arg \min_{(\mathbf{\Delta x},\ \mathbf{\Delta y}) \in \mathbf{S}} \left\{ \sum_{i=1}^{p} \left[\sum_{k=1}^{N} h_k \left| \left(\hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) - (1-p_e) E\left\{ \tilde{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) \right\} - p_e E\left\{ \tilde{f}_{n-1}^{i} \right\} \right) \right| \right] \right\}\end{aligned} \tag{30}$$

Expressing the power (variance) of the prediction residual signals in MHC as:

$$\sigma_n^2(\mathbf{\Delta x}, \mathbf{\Delta y}) = \sum_{i=1}^{p} \left(f_n^i - \sum_{k=1}^{N} h_k \hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) \right)^2, \tag{31}$$

the following constrained optimization problem is formulated to better compromise between coding efficiency and error resiliency:

$$\min_{(\mathbf{\Delta x},\ \mathbf{\Delta y}) \in \mathbf{S}} \left\{ \sum_{i=1}^{p} \left[\sum_{k=1}^{N} h_k \left| \left(\hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) - (1-p_e) E\left\{ \tilde{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) \right\} - p_e E\left\{ \tilde{f}_{n-1}^{i} \right\} \right) \right| \right] \right\} \tag{32}$$

$$\text{Subject to: } \sigma_n^2(\mathbf{\Delta x}, \mathbf{\Delta y}) = \sum_{i=1}^{p} \left(f_n^i - \sum_{k=1}^{N} h_k \hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) \right)^2 < T$$

Obviously, the threshold T determines the tradeoff between coding efficiency and error resiliency. It is not straightforward enough to see from Eq. (32) how the T impacts upon finding error resilient MV for each hypothesis. For more clarity, Eq. (31) is re-written as:

$$\sigma_n^2(\mathbf{\Delta x}, \mathbf{\Delta y}) = \sum_{i=1}^{p} \left[\begin{aligned} &\sum_{k=1}^{N} h_k^2 \left(f_n^i - \hat{f}_{n-k}^{i,j} \right)^2 \\ &+ 2 \sum_{k=1,\ k \neq q,}^{N} \sum_{q=1}^{N} h_k h_q \left(f_n^i - \hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) \right) \left(f_n^i - \hat{f}_{n-q}^{i,j}(\Delta x_q, \Delta y_q) \right) \end{aligned} \right]. \tag{33}$$

Define

$$\delta_k^2(\Delta x_k, \Delta y_k) \equiv \sum \left(f_n^i - \hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) \right)^2 \tag{34}$$

which represents the autocorrelation (or, variance) of the motion residual signal and is always lager than the cross-correlation term (a Gaussian residual signal of zero-mean is

assumed). Therefore, for a given hypothesis number k and for each $q = 1 \sim N$, $q \neq k$, the following inequality will hold,

$$h_k^2 \delta_k^2 \geq \sum_{i=1}^{p} \left[h_k h_q \left(f_n^i - \hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) \right) \left(f_n^i - \hat{f}_{n-q}^{i,j}(\Delta x_q, \Delta y_q) \right) \right] \tag{35}$$

Hence, $\sigma_n^2(\mathbf{\Delta x}, \mathbf{\Delta y})$ in Eq. (33) can be expressed in terms of $\delta_k^2(\Delta x_k, \Delta y_k)'s, k = 1 \sim N$:

$$\sigma_n^2(\mathbf{\Delta x}, \mathbf{\Delta y}) \leq \sum_{k=1}^{N} h_k^2 \delta_k^2 + 2 \sum_{k=1}^{N} h_k^2 (N-k) \delta_k^2 = \sum_{k=1}^{N} h_k^2 \delta_k^2 (1 + 2N - 2k) \tag{36}$$

Hence, a constraint T on $\sigma_n^2(\mathbf{\Delta x}, \mathbf{\Delta y})$ can be decomposed into separate constraints $T_k's$ on $\delta_k^2(\Delta x_k, \Delta y_k)'s, k = 1 \sim N$. In other words, there exists a mapping between (T_1, T_2, …T_N) and T via

$$T = \sum_{k=1}^{N} h_k^2 T_k (1 + 2N - 2k), \tag{37}$$

which is similar to Eq. (36). We can specify the value of the total threshold T indirectly via individual $T_k's$ to tradeoff between coding efficiency and error resiliency separately for each hypothesis. That is, we divide the original constrained optimization problem into N sub-problem, which is much simpler.

The proposed error resilient motion estimation algorithm for a given weighting vector $\boldsymbol{h}$ is now summarized as follows. First, set a constraint $\delta_k^2(\Delta x_k, \Delta y_k) \leq T_k$ for each reference frame f_{n-k} for motion estimation and then choose among the MVs, which satisfy the above constraint, the one that minimizes

$$\sum_{i=1}^{p} \left| \left(\hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) - (1 - p_e) E\left\{ \tilde{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k) \right\} - p_e E\left\{ \tilde{f}_{n-1}^{i} \right\} \right) \right| \tag{38}$$

The above procedure is performed for each hypothesis k, k= 1~N. Finally, generate the motion prediction residual by subtracting the weighted hypothesis $\sum_{k=1}^{N} h_k \hat{f}_{n-k}^{i,j}(\Delta x_k, \Delta y_k)$ from the original video data.

6. Adaptive multi-hypothesis coding

In Section 5, error resilient motion estimation for a given set of hypothesis-weighting coefficients $\boldsymbol{h}$ is discussed. That is, $\boldsymbol{h}$ remains constant along the whole video. Now, in this section, we explore the advantage of varying the weighting coefficients $\boldsymbol{h}$, frame by frame, to further enhance error resiliency of the transmitted videos, according to the channel packet loss rate and video contents.

Before illustrating how to estimate the optimal $\boldsymbol{h}$ for error-prone video transmission, the power of the channel distortion signal is derived similarly as in Section 5:

$$\begin{aligned}\sigma_{\tilde{n}\tilde{n}}^2 &= \sum_{i=1}^{p} E\left\{\left(\hat{f}_n^i - \tilde{f}_n^i\right)^2\right\} \\ &= \sum_{i=1}^{p}\left(p_e \cdot E\left\{\left(\hat{f}_n^i - \tilde{f}_{n-1}^i\right)^2\right\} + (1-p_e)\cdot E\left\{\left[\hat{f}_n^i - \left(\hat{r}_n^i + \sum_{k=1}^{N} h_k \tilde{f}_{n-k}^{i,j}\right)\right]^2\right\}\right)\end{aligned} \tag{39}$$

The i^{th} term in summation can be reformulated as follows.

$$\begin{aligned}&p_e \cdot E\left\{\left(\hat{f}_n^i + \hat{f}_{n-1}^i - \hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)^2\right\} + (1-p_e)\cdot E\left\{\left[\sum_{k=1}^{N} h_k \hat{f}_{n-k}^{i,j} - \sum_{k=1}^{N} h_k \tilde{f}_{n-k}^{i,j}\right]^2\right\} \\ &= p_e \cdot E\left\{\left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2\right\} + p_e \cdot E\left\{\left(\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)^2\right\} + 2p_e \cdot E\left\{\left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)\left(\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)\right\} \\ &\quad + (1-p_e)\cdot E\left\{\left[\sum_{k=1}^{N} h_k \hat{f}_{n-k}^{i,j} - \sum_{k=1}^{N} h_k \tilde{f}_{n-k}^{i,j}\right]^2\right\} \\ &= p_e \cdot \left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2 + p_e \cdot E\left\{\left(\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)^2\right\} + 2p_e \cdot \left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)\left(\hat{f}_{n-1}^i - E\left\{\tilde{f}_{n-1}^i\right\}\right) \\ &\quad + (1-p_e)\cdot E\left\{\left[\sum_{k=1}^{N} h_k \hat{f}_{n-k}^{i,j} - \sum_{k=1}^{N} h_k \tilde{f}_{n-k}^{i,j}\right]^2\right\}\end{aligned} \tag{40}$$

It is further assumed that $\left(\hat{f}_{n-k}^i - \tilde{f}_{n-k}^i\right)$ and $\left(\hat{f}_{n-q}^i - \tilde{f}_{n-q}^i\right)$, $k \neq q$, are independent. Then, the channel distortion power in Eq.(39) can be approximated as:

$$\begin{aligned}\sigma_{\tilde{n}\tilde{n}}^2 \cong \sum_{i=1}^{p}\Bigg(& p_e \cdot \left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2 + p_e \cdot E\left\{\left(\hat{f}_{n-1}^i - \tilde{f}_{n-1}^i\right)^2\right\} + 2p_e \cdot \left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)\left(\hat{f}_{n-1}^i - E\left\{\tilde{f}_{n-1}^i\right\}\right) \\ & + (1-p_e)\cdot \sum_{k=1}^{N} h_k^2 E\left\{\left(\hat{f}_{n-k}^{i,j} - \tilde{f}_{n-k}^{i,j}\right)^2\right\}\Bigg)\end{aligned} \tag{41}$$

Some notes about Eq. (41) are emphasized below. First, for given MVs, the first moment $E\left\{\tilde{f}_{n-1}^i\right\}$ can be estimated by using Eq. (27). Secondly, Eq. (41) is a recursive formula, meaning that the channel distortion power of the previous N frames, i.e., $E\left\{\left(\hat{f}_{n-k}^i - \tilde{f}_{n-k}^i\right)^2\right\}$, $k = 1 \sim N$, are used in estimating the channel distortion power of the current frame. Finally, if a strategy other than zero-motion is adopted for error concealment, the first two terms in Eq. (41) will change accordingly and may result in a lower channel distortion power. However, discussions about finding proper concealment strategies that are able to minimize channel distortion power are beyond the scope of this chapter.

The last term in Eq. (41) reveals that it is possible to find one combination of h_k's such that the channel distortion power is minimized. This problem can be formulated as follows:

$$\begin{aligned} &\text{minimize} \quad \left(1/2\right) \boldsymbol{h}^T \mathbf{D} \boldsymbol{h} \\ &\text{subject to} \quad \mathbf{1}^T \boldsymbol{h} = 1 \quad \text{and} \quad \boldsymbol{h} \succeq 0 \end{aligned} \tag{42}$$

where **D** is an $N \times N$ diagonal matrix whose elements on diagonal, denoted as $\sigma^2_{\tilde{n}_k \tilde{n}_k}$, is

$$\sum_{b=1}^{M} \sum_{i=1}^{p} E\left\{ \left(\hat{f}_{n-k}^{b,i,j} - \tilde{f}_{n-k}^{b,i,j} \right)^2 \right\}, \tag{43}$$

where b is the MB index and M is the total number of MBs in a video frame.

The problem in Eq. (43) is a convex optimization problem and the Karush-Kuhn-Tucker (KKT) conditions can be applied to find a solution of Eq. (42). Here, we consider the case of $N = 3$ for better understanding:

$$\begin{bmatrix} \sigma^2_{\tilde{n}_1 \tilde{n}_1} & 0 & 0 \\ 0 & \sigma^2_{\tilde{n}_2 \tilde{n}_2} & 0 \\ 0 & 0 & \sigma^2_{\tilde{n}_3 \tilde{n}_3} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} + v \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{44}$$

where v is a Lagrange multipliers. Equation (44) is used to estimate the optimal value of h_k, $k = 1 \sim 3$, with a constraint of $h_1 + h_2 + h_3 = 1.0$.

$$h_k = \frac{\sigma^2_{\tilde{n}_i \tilde{n}_i} \sigma^2_{\tilde{n}_j \tilde{n}_j}}{\sigma^2_{\tilde{n}_1 \tilde{n}_1} \sigma^2_{\tilde{n}_2 \tilde{n}_2} + \sigma^2_{\tilde{n}_1 \tilde{n}_1} \sigma^2_{\tilde{n}_3 \tilde{n}_3} + \sigma^2_{\tilde{n}_2 \tilde{n}_2} \sigma^2_{\tilde{n}_3 \tilde{n}_3}}, \quad \text{for } i \neq k \neq l \tag{45}$$

Note that the determination of optimal $\boldsymbol{h}$ is based on the availability of MVs for all the MBs in a frame. That is, multi-hypothesis motion estimation based on the conventional algorithm is performed first, then Eq. (45) (as well as other related formula) is used to figure out the optimal hypothesis-weighting coefficients $\boldsymbol{h}^*$, and finally these N hypotheses are combined to estimate the prediction. By the way, the overhead for transmitting $\boldsymbol{h}$ information is less and can be ignored.

7. Experimental results

7.1 Experiments for ERME and ERMD

The proposed ERME and ERMD algorithms are implemented on H.264/AVC test model JM 9.3 with rate control being enabled to meet channel bandwidth constraint. The number of reference frame and the search range for motion estimation are 3 and ±16 pixels, respectively. There are a total of 100 CIF frames for each test sequence. The performance of the proposed ERME algorithm is verified first by encoding the first frame as I picture and the rest as P pictures without intra-coded MBs therein. To evaluate the proposed ERME algorithm, mode decision for MBs in P pictures is purposely disabled, that is, only the block size of 16x16 pixels is considered. The setting of the threshold *Thd1* in Eq. (13) needs to be explained further. Theoretically, by adjusting *Thd1*, one can explore the trading behaviours

between error resiliency and coding efficiency of MVs. For the purpose of analysis, let $Thd1 = \alpha \cdot \hat{\sigma}^2$, where α is a pre-set constant and $\hat{\sigma}^2$ is derived via dynamic MB analysis. For better understanding, we let $\hat{\sigma}^2$ be equal to $\gamma^{\min}$ (previously stated in Eq. (12)), the smallest residual energy that can be achievable via conventional motion estimation process. Henceforth, α is selected to be larger than 1. It follows that a larger α implies a larger sacrifice on matching optimality (in a metric of square errors), or, the coding efficiency. Note that *Thd1* is dynamically varied on the MB base to account for the variation in video contents.

Figure 2 illustrates the PSNR performance at varying α and transmission bit rates. Curves marked with "alpha_*x*" represent the results obtained by setting α to *x*, e.g., "alpha_10" means $\alpha = 10$. On the other hand, curves annotated with "alpha_*" represent the results obtained by using conventional ME method (i.e., no consideration on error resiliency). It is assumed that each transmitted packet contains a slice which is composed of a row of consecutive MBs. When errors occur during transmission, the method of zero motion is adopted for error concealment at receiver, i.e., the damaged MB is replaced with that at the same location in the previous frame. Different error patterns for each packet loss rate (5% or 15%) are simulated to obtain an ensemble average PSNR of the reconstructed video.

From the figures, it is observed that with an increasing α , the coding efficiency decrease (i.e., less PSNR at a given bit rate) as expected when PSNRs are measured at encoder's local decoder (i.e., error-free scenario, (a)(b)). This is reasonable since the number of MV candidates satisfying the constraint in Eq. (12) is increased and all of them lead to residual energy larger than $\gamma^{\min}$. The sacrificed coding efficiency would lead to a gain on error resiliency, if the PSNRs are measured at receiver's decoder (i.e., error-prone scenario, (c)(d)). Take the curve of $\alpha = 2$ and $\alpha = 6$ for comparison, there exists a crossing point where the average PSNR for $\alpha = 6$ becomes better than that for $\alpha = 2$. However, this is not the similar case (no distinct crossing point exists) between $\alpha = 10$ and $\alpha = 20$.

It is observed from Fig, 3(a) that the anchor's face encounters severe distortion due to the lack of error resiliency for conventional ME algorithms. On the other hand, distortions of the anchor's face in (b) and (c) are much less than that in (a) of Fig. 3, due to the enhancement of error resiliency for increasing α . However, increasing quantization errors in other areas (e.g., words in the background) may lower down the total PSNR performance. This situation becomes clearer when D_s dominates D_c (e.g., $\alpha = 20$ not shown here).

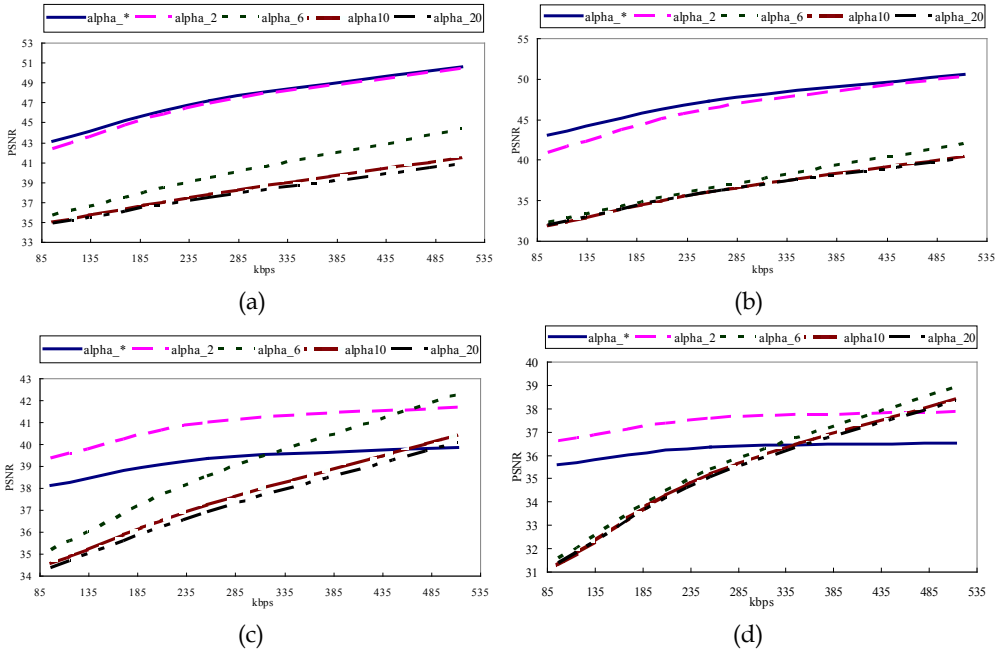
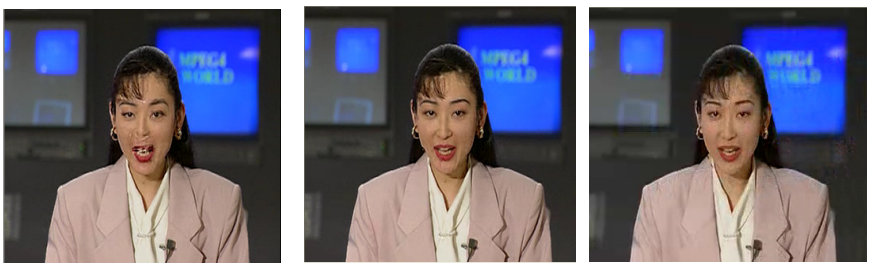


Fig. 2. Illustration of compromising error resiliency with coding efficiency of the proposed ERME algorithm. (a)(b) Reconstructed "Akiyo"at local decoder. (c)(d) Reconstructed "Akiyo"at decoder. The packet loss rate is (a)(c) 5% and (b)(d) 15%, respectively.



(a) Traditional ME (b) $\alpha = 2$ (c) $\alpha = 10$

Fig. 3. Visual quality at varying α, showing different compromises between D_s and D_c. The bit rate is 256 kbps and the packet loss rate is 5%.

Another experiment of compromising error resiliency with coding efficiency is conducted for the video "Foreman" (with high motion), which is also not shown here. In comparison with the experiments for "Akiyo" (Fig. 2), the PSNR crossing point is advanced to a lower bit rate for high motion videos. We can temporarily come to a conclusion that α can be chosen smaller (e.g., 2) for static videos while larger (e.g., ≥ 6) for high motion videos.

Remind that the choice of $\hat{\sigma}^2 = \gamma^{\min}$ is only for comparison purpose. Practically, $\hat{\sigma}^2$ should be derived by a fast and simple analysis for each MB, e.g., by summing the square of the temporal difference with respect to the previous frame. This method is certainly computationally cheaper and makes the threshold *Thd1* MB-adaptive.

Next, rate-distortion performance of our proposed ERME+ERMD algorithm is illustrated. We relax the allowable modes to include 16×16, 8×16, 16×8, 8×8, and intra 16×16. Notice that since we focus on finding MVs which can improve robustness to transmission errors of the compressed videos, the methods for compressing I-pictures here conforms to that recommended in the H.264/AVC standard. The integration of both techniques means that for each mode in $\boldsymbol{M}$, except the intra 16×16, the ERME algorithm is first applied to find the respective MVs, then the ERMD algorithm, with embodiment in term of Eq. (16), is applied to select the optimal mode among $\boldsymbol{M}$. In the following experiments, a value of $\alpha = 2$ is selected for ERME algorithm.

In Fig. 4, rate-distortion performances of several variational methods are compared. Three test sequences are chosen to represent typical videos of high motion (e.g., "Stefan"), moderate motion (e.g., "Foreman"), and static (e.g., "Akiyo") contents. In these figures, curves annotated with "MEO-MDO" represent the proposed "ERME+ERMD" algorithm, those annotated with "MEN-MDO" represent "ME+ERMD" algorithm, and curves annotated with "MEN-MDN" represent the traditional "ME+MD" algorithm adopted in H.264/JVC JM 9.3 model. For packet loss rate of 5% and 15%, the proposed ERME+ERMD improves error resiliency of the compressed video significantly (by 1~7 dB). These experiments also verify that ME+ERMD algorithm is not sufficient to support transmission robustness. The ERME algorithm is promising to provide additional support to complement this deficiency.

In Table 1, statistics of coding modes finally chosen by conventional H.264/AVC and the proposed ERME+ERMD algorithm are listed, which are obtained after gathering the related information for three different video sequences under different bit rates and a given 5% packet loss rate. It is observed that although ERME is able to prevent compressed videos from temporal error propagation, which has been verified in Fig. 2, the most efficient mode to prevent error propagation is "intra". The percentage is higher for higher bit rates. This behavior is reasonable since the additional bit rate will gain more benfits from intra-coding than from finer quantization.

7.2 Experiments for MHC

The multi-hypothesis coding algorithm is also implemented based on the H.264 video coding standard. Also, the first frame is coded as I frame and the rest are coded as P frames without any intra-coded blocks in them. It is assumed that a packet contains of a row of MBs.

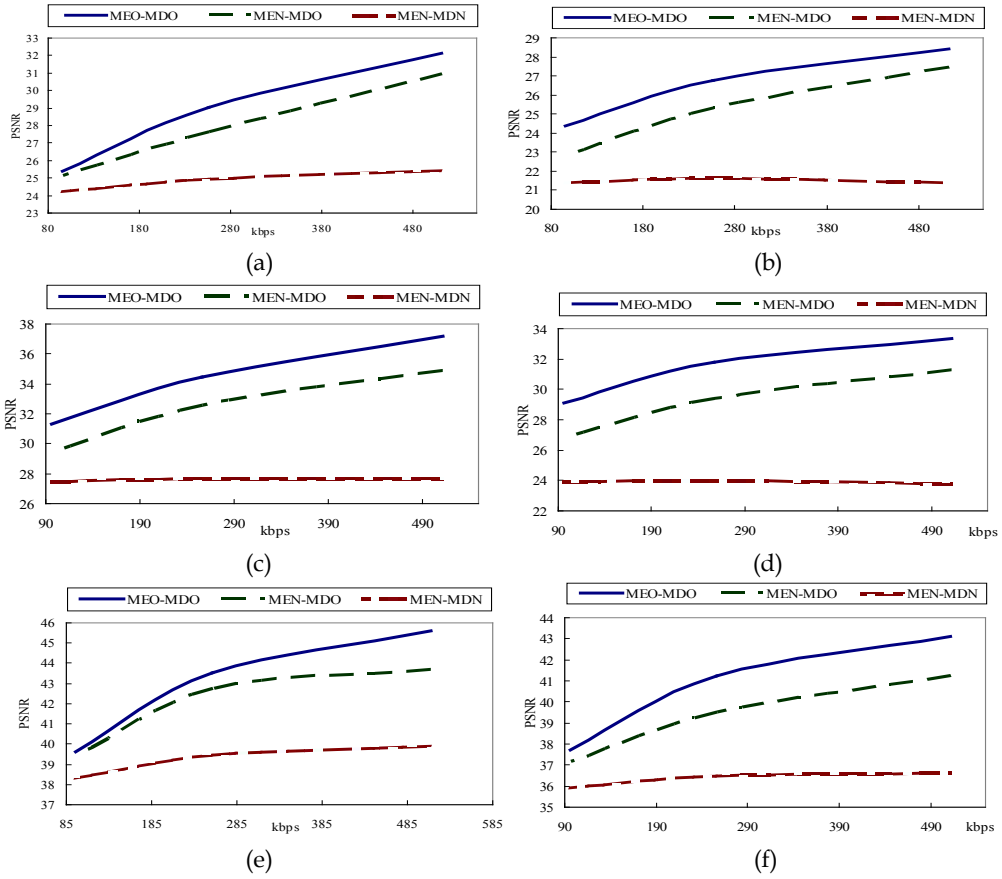


Fig. 4. Rate-distortion performances of reconstructed video at decoder for ERME+ERMD (MEO-MDO), ME+ERMD (MEN-MDO), and ME+MD (MEN-MDN) methods. (a)(b) "Stefan", (c)(d) "Forman", (e)(f) "Akiyo". (a)(c)(e) : PLR=5%, (b)(d)(f): PLR=15%.

Bit rate = 128kbs, Packet loss rate = 5%						
Modes	Foreman`		Stefan		Mobile	
	H.264 (%)	Proposed (%)	H.264 (%)	Proposed (%)	H.264 (%)	Proposed (%)
16x16	47.6	22.3	52.3	22.1	54.1	31.9
16x8	20.5	3.4	19.1	1.8	23.2	6.2
8x16	24.0	3.0	15.9	2.3	22.3	8.6
8x8	0	0	0	0	0	0
Intra	7.9	71.3	12.7	73.8	0.4	53.3

Table 1. Statistics of coding modes chosen by H.264 and the proposed ERME+ERMD.

First, the performance of the proposed error resilient motion estimation algorithm for MHC is evaluated. According to Eq. (38), we have decomposed the multi-hypothesis problem into N single-hypothesis problem. Let the threshold T_k for each hypothesis be in a form of $x \cdot \delta_k$, where δ_k is the energy of the prediction errors based on the MV (i.e., Δx_k and Δy_k) obtained by using the conventional motion estimation algorithm and x is a scale factor. Obviously, for each hypothesis, the threshold T_k is capable of adapting to video contents block-wise rather than frame-wised. On the other hand, adjustment of x makes us able to control the degree of compromise between coding efficiency and error resiliency. It is also easy to observe that the larger scaling factor x is chosen, the larger source distortion D_s is incurred, i.e., the more coding efficiency is sacrificed. Note that the above-mentioned way of getting δ_k is only for comparison purpose. One more practical method is to let δ_k be the energy of the residuals obtained via zero-motion compensation. Note that zero-motion compensation (or direct frame-difference) incurs less computational load and is hence more feasible.

In Figs. 5 and 6, rate-distortion performances in both the error-free and error-prone transmission environments between the proposed error resilient motion estimation and the conventional H.264 motion estimation techniques, are compared. The curves with "alpha_x" represent the results obtained by varying x in $T_k = x \cdot \delta_k$ to account for different levels of tradeoffs between coding efficiency and error resiliency. The curve marked with "alpha_*" represents the conventional ME algorithm.

From the figures, it is observed that in error-free environments, the proposed algorithm always results in a poorer performance than the conventional ME algorithm. This situation is even worse when x is increased (thus T_k is increased). It is observed that with packet losses, our proposed algorithm improves videos quality at higher bit rates. A larger T_k will result in a better error resiliency (up to 1 dB). At low bit rates, since the coding efficiency is quite sacrificed, the gain in error resiliency is not enough to overcome the former loss. It is also observed that similar performances are obtained for different weighting coefficients $\boldsymbol{h}$.

Next, the performance of our adaptive multi-hypothesis coding technique is to be evaluated. The number of hypotheses is 3 and the quantization parameter (QP) remains fixed during encoding a video. The packet loss rate is set to 5% and PSNR is measured between the reconstructed frames at local decoder and the reconstructed frames at receiver's decoder. Since the accuracy of the estimated channel distortion power is essential to the determination of the hypothesis-weighting coefficients to improve error resiliency, the accuracy of Eq. (41) is evaluated first. In experiments not shown here, the average difference between the estimated (via Eq.(41)) and the measured PSNRs for the high-motion video STEFAN is no more than 3.16 %.

Figure 7 relates to the experiment results of adaptive multi-hypothesis coding based on Eqs. (42) and (45). Notice that now MVs for each MB are obtained by using the conventional motion estimation algorithm. The two-state Gilbert channel as described in Zorzi et al. (1997) is used to simulate error conditions of packet error rate = 10% and average packet-error-burst length = 18.

In Fig. 7, the curves marked with "Adaptive" represent our proposed algorithm and curves marked with "Fixed" represent the conventional algorithm (i.e., constant and even weighting coefficients). From the figure, it is observed that our proposed algorithm really enhances the error resiliency of the multi-hypothesis video coding technique by up to 1 dB.

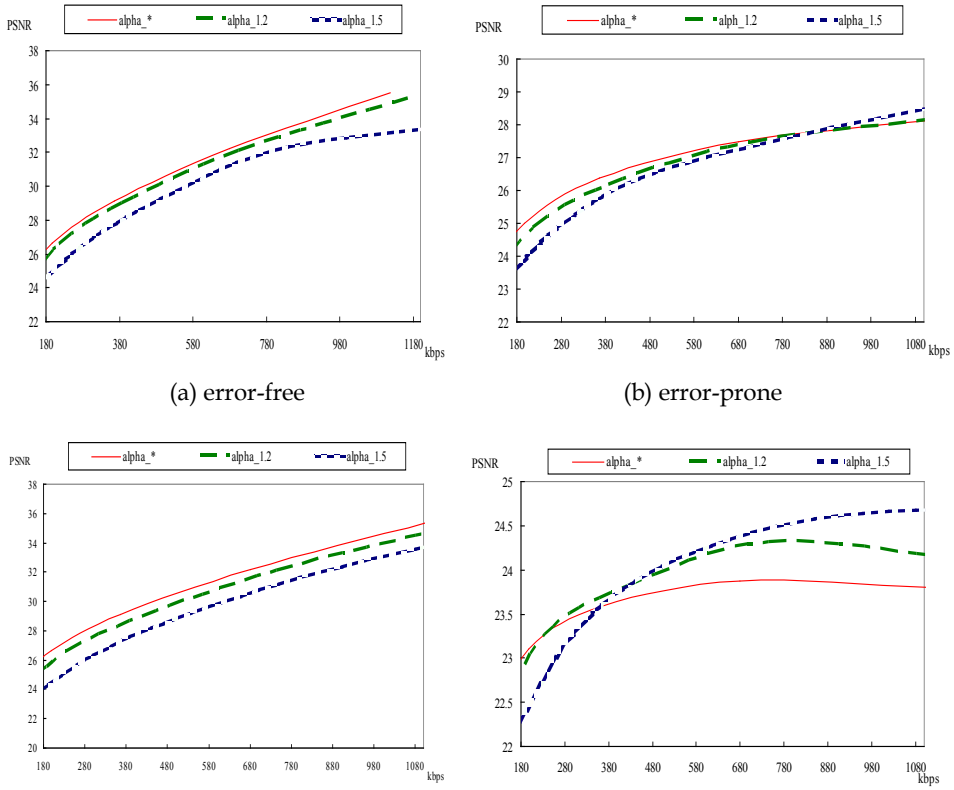


Fig. 5. Rate-distortion performance comparison with hypothesis-weighting coefficients (1/3, 1/3, 1/3) along the whole video "STEFAN". The packet loss rate for (a)(b) and (c)(d) is 5% and 15%, respectively.

8. Conclusion

In this chapter, error resilient coding techniques considering end-to-end distortions for videos transmitted on error-prone channels are discussed. The main concept of the algorithms is to decompose end-to-end distortions into two parts: source distortion relating to coding efficiency and channel distortion relating, on the other hand, to error resiliency. Most often, these two objectives are intrinsically mutual conflicting under a target bit rate. Hence it needs to make a proper compromise between them.

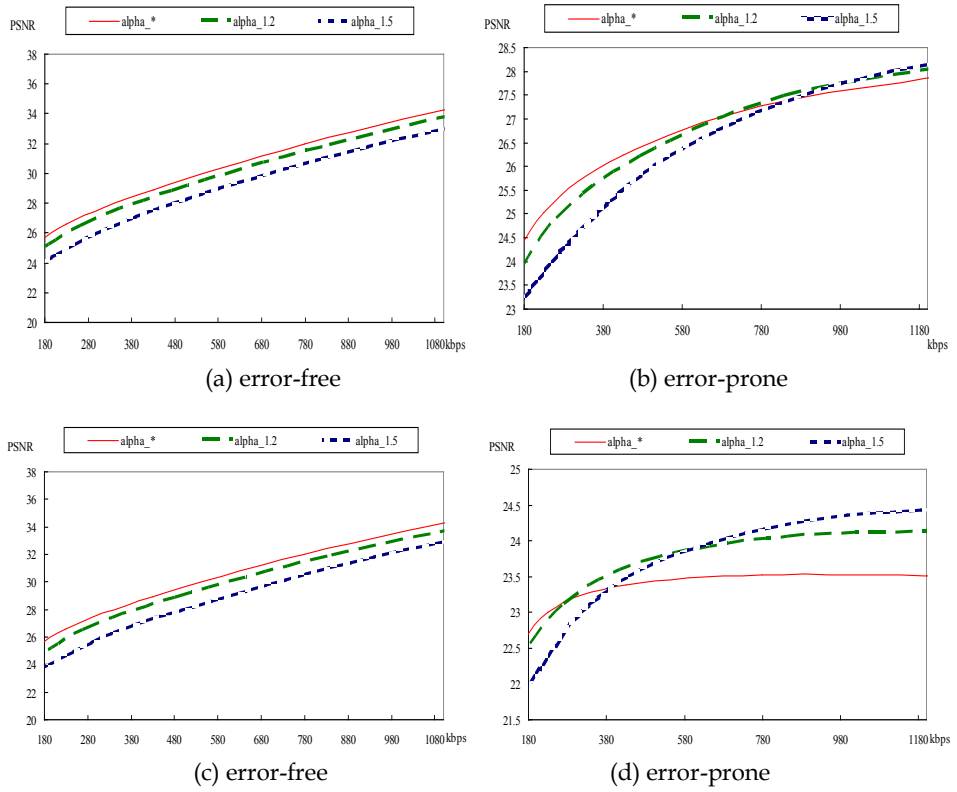


Fig. 6. Rate-distortion performance comparison with hypothesis-weighting coefficients (0.7, 0.2, 0.1) along the whole video "STEFAN". The packet loss rate for (a)(b) and (c)(d) is 5% and 15%, respectively.

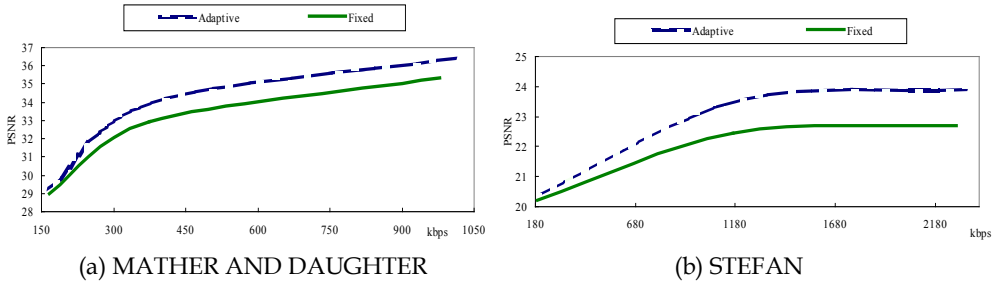


Fig. 7. Performance of the adaptive multi-hypothesis coding technique. The average packet-error-burst length is 18 and the packet-error-rate is 10 %.

In accordanc with the above concept, we propose ERME algorithms for both traditional H.264/AVC and multi-hypothesis coding architectures to suppress the temporal error propagation effectively with relatively low computational overhead. The similar concept is also applied to inter mode selection of H.264/AVC and to adaptive multi-hypothesis coding by finding out the relationship between the end-to-end distortions and the coding parameters, e.g., coding modes or hypothesis weighting coefficients before solving the optimization problem. Experiment results verify the accuracy of the proposed end-to-end distortion model in respective area.

The performance of the ERME algorithm is sunstantially affected by the constraint that controls different degrees of compromise between coding efficiency and error resiliency. How to set the constraint more accurately is a quite important issue that needs to be investigated further. Besides, integrating the ERME and adaptive hypothesis-weighting algorithms for further enhancement of error resiliency is not done yet. Note that the premise of the ERME is a given weighting $\boldsymbol{h}$, while adaptive hypothesis-weighting relies on the availability of MVs, seemingly a chicken-and-egg problem. A possible way is to compute the optimal $\boldsymbol{h}$ for frame t+1 based on the MVs of frame t by assuming that consecutive frames have strong similarity on MVs or channel distortion power, thought it might be violated due to large motion or scene change.

9. References

Boyd, Stephen & Vandenberghe, Lieven (2004). *Convex Optimization*, Cambridge University Press.

Chang, Pao-Chi; Lee, Tien-Hsu; Chen, Jhin-Bin & Tsai, Ming-Kuang (2005). Encoder-originated error resilient schemes for H.264 video coding. *Proceedings of 18th IPPR Conference on Computer Vision, Graphics and Image Processing,* pp. 406–412, Aug. 2005.

Cote, G. & Kossentini, F. (1999). Optimal intra coding of blocks for robust video communication over the Internet. *Signal Processing: Image Commu*., Sep.1999,pp. 25-34.

Eisenberg, Yiftach; Zhai, Fan; Papppas, Thrasyvoulos N.; Berry, Randall & Katsaggelos, Aggelos K. (2006). VAPOR: Variance-aware per-pixel optimal resource allocation. *IEEE Trans. on Image Processing*, Vol. 15, No 2, Feb. 2006, pp. 289-200.

Flierl, M.; Wiegand, T. & Girod, B. (2000). A Video Codec Incorporating Block-Based Multi-Hypothesis Motion-Compensated Prediction. *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, Vol. 4067, pp. 238-249, June 2000.

Flierl, Markus; Wiegand, Thomsa & Girod, Bernd (1998). A locally optimal design algorithm for block-based multi-hypothesis motion-compensated prediction. *Proceedings of IEEE Conf. On Data Compression*, pp. 239-248, March 1998.

Flierl, Markus; Wiegand, Thomas & Girod, Bernd (2002). Rate-constrained multihypotheis prediction for motion-compensation video compression. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 12, No. 11, Nov. 2002, pp. 957-969.

Girod, Bernd (2000). Efficiency analysis of multihypothesis motion-compensated prediction for video coding. *IEEE Trans. on Image Processing*, Vol. 9, No. 2, Feb. 2000, pp.173-183.

Harmanci, Oztan & Tekal, A. Murat (2005). Stochastic frame buffers for rate distortion optimized loss resilient video communications. *Proceedings of IEEE Int'l Conf. Image Processing,* Vol. 1, pp. 789-792, Sep. 2005.

He, Zhihai; Cai, Jianfei & Chen, Chang Wen (2002). Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.12, No.6, Jun. 2002, pp. 511-523.

Kung, Wei-Ying; Kim, Chang-Su & Kuo, C. –C. Jay (2006) Analysis of multi-hypothesis motion compensation prediction for robust video transmission. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 16, No. 1, Jan. 2006, pp. 146-153.

Leontaris, A. & Cosman, P.C. (2004). Video compression for lossy packet networks with mode switching and a dual-frame buffer. *IEEE Trans. on Image Processing*, Vol. 13, July 2004, pp. 885-897.

Lie, Wen-Nung; Gao, Zhi-Wei & Liu, Tung-Lin (2006). Joint source-channel video coding based on the optimization of end-to-end distortion. *Proceedings of Pacific-Rim Symposium on Image and video Technology,* 2006.

Lin, Shunan & Wang, Yao (2002). Error resilience property of multihypothesis motion-compensated prediction. *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, pp.545-548, June 2002.

Reyes, Gustavo de los; Reibman, Amy R.; Chang, Shih Fu & Chuang, Justin C.-I. (2000). Error-resilient transcoding for video over wireless channels. *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 6, June 2000, pp. 1063-1074.

Ringuest, Jeffrey L. (1992). *Multiobjective Optimization: Behavioral and Computational Considerations.* Kluwer Academic Publishers.

Stuhlmuller, Klaus; Farber, Niko; Link, Michael & Girod, Bernd (2000). Analysis of video transmission over lossy channel. *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 6, June 2000, pp. 1012-1032.

Sullivan, G. J. (1993). Multi-hypothesis motion compensation for low bit rate video coding. *Proceedings of IEEE Int'l Conf. Acoustics, Speech, and Signal processing,* Vol. 5, pp. 437-440, Apr. 1993.

Sun, M.-T. & Reibman, A.R. (2001). *Compressed Video over Network*, Marcel Dekker, Inc. New York, Basel.

Wiegand, Thomas; Farber, Niko; Stuhlmuller, Klaus & Girod, Bernd (2000). Error-resilient video transmission using long-term memory motion-compensated prediction. *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 6, June 2000, pp. 1050-1062.

Xia, Minghui; Vetro, Anthony; Sun, Huifan & Liu, Bede (2004). Rate-distortion optimized bit allocation for error resilient video transcoding. *Proceedings of IEEE Int'l Symp. Circuits and Systems,* Vol. 3, pp. 945-948, May 2004.

Yang, Hua & Rose, K. (2005). Rate-Distortion optimized motion estimation for error resilient video coding. *Proceedings of IEEE Int'l Conf. Acoustics, Speech, and Signal processing,* Vol. 2, March 2005, pp. 173-176.

Zhan, R.; Regunathan, S. L. & Rose K. (2000). Video coding with optimal intra/inter mode switching for packet loss resilience. *IEEE Journal of Selected Areas in Commun.*, Vol. 18, No. 6, June 2000, pp. 966-976.

Zorzi, M.; Rao, K.R. & Milstein, L. B. (1997). ARQ error control for fading mobile radio channels. *IEEE Trans. On Vehicle Technology.*, Vol. 46, May 1997, pp. 445-455.

14

Bit Rate Estimation for Cost Function of H.264/AVC

Mohammed Golam Sarwer[1,2], Lai Man Po[1] and Q. M. Jonathan Wu[2]
[1]*City University of Hong Kong, Hong Kong SAR, China*
[2]*University of Windsor, Canada*

1. Introduction

The appearance and development of various new multimedia services have need for higher coding efficiency. The ITU-T/ISO/IEC Joint Video Team established the newest video coding standard known as H.264/AVC (Joint Video Team (JVT), 2002). H.264/AVC offers a significant performance improvement over previous video coding standards such as H.263++ and MPEG-4 part 2 (Erol et al., 1998; Topiwala et al., 2001). New and advanced techniques are introduced in this new standard, such as intra prediction for I-frame encoding, multi-frames inter prediction, small block-size transform coding, context-adaptive arithmetic entropy coding, de-blocking filtering, etc. These advanced techniques make this new standard provides approximately 50% bit rate saving for equivalent perceptual quality relative to the performance of prior standards.

To achieve the best coding efficiency, H.264/AVC employs the rate-distortion (RD) optimization technique to get the best result consider of the visual quality and bit rate. For a macroblock in I-slice, RD optimization exhaustively searches the predefined 13 intra modes (9 modes for 4x4 block and 4 modes for 16x16 block) to produce the best encode mode for this macroblock. However, the latest H.264/AVC standard also defines 9 intra prediction modes for 8x8 block, for simplicity in this chapter only intra 4x4 and 16x16 blocks are considered. When a macroblock is in P-slice, it may be coded in intra mode or inter mode, so RD optimization employs a brute force algorithm to search through all possible inter modes and intra modes to find the best choice. RD optimization ofcourse will get the best encode mode for a macroblock, but it is at the great expanse of higher computational complexity at the encoder. To reduce computational complexity of H.264/AVC, a number of efforts have been made to explore the fast algorithm in motion estimation, intra mode prediction and inter mode prediction for H.264/AVC video coding (Chen et al., 2002; Feng et al., 2005; Han & Lee, 2005; Kim et al., 2006; Lim et al., 2003; Sarwer et al., 2008; Sarwer & Wu, 2009; Wu et al., 2005; Yang et al., 2004). The number of inter modes is reduced by using the amplitude of edge vector (Lim et al., 2003). To reduce the complexity of intra mode decision, H.264/AVC reference software suggested sum of absolute difference (SAD) and sum of absolute transform difference (SATD) based cost functions. These two cost functions reduce computation significantly but performance of RD characteristics is not good enough. In (Chiang & Zhang, 1997) and (Coebera & Lei, 1999) rate models were observed from the

quantizer (Q)-domain. But these are considered only for rate control. To improve the RD performance, an enhanced cost function for intra 4x4 mode decisions was proposed in (Tseng et al., 2006). In this cost function, sum of absolute integer transform difference (SAITD) is used in distortion part and a rate prediction algorithm is used in rate part. The major drawback of this cost is that the bit estimation method cannot give the very good estimation.

In this chapter, we propose a shortcut way to get the number of entropy coded bits as soon as the transform coefficients are quantized. A method for estimation of rate for cost function of intra and inter mode decision is proposed. This method is based on the properties of context-based adaptive variable length coding (CAVLC) and observation of VLC tables. The total number of bits need to encode a quantized residual block is predicted by estimating the rate of each symbols of CAVLC separately.

The remainder of this chapter is organized as follows. Section 2 provides the review of intra and inter mode and rate distortion optimized mode decision technique. In section 3, Context-based adaptive variable length coding (CAVLC) is briefly described. In section 4, we present the proposed fast bit rate estimation method. The performance results of the proposed method are presented in section 5. Finally, section 6 concludes the chapter.

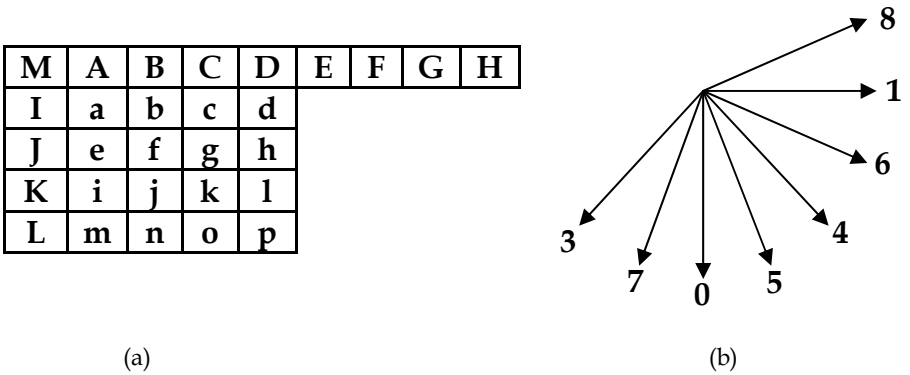


Fig. 1. (a) A 4x4 block with elements (a to p) which are predicted by its neighbouring pixels (b) Eight prediction direction for I4MB prediction

2. Overview of Mode Decision

2.1 Intra mode in H.264/AVC

The H.264/AVC video coding standard supports intra prediction for variable block size. If a macroblock (MB) is encoded in intra mode, a predicted block is formed based on previously encoded and reconstructed upper and left blocks so it exploits the spatial correlation between the adjacent MB. Then the residual data between the current to be coded MB and predicted one is DCT transformed, quantized and entropy coded. For each luma samples, the prediction block may be formed for each 4x4 block (denoted as I4MB) or for an entire MB (denoted as I16MB). When using the I4MB prediction, each 4x4 block of the luma component utilizes one of the nine prediction modes. Besides DC prediction, eight directional modes are specified. When utilizing I16MB, which is well suited for homogeneous area of image, 4 prediction modes are supported. Another 4 prediction modes

are used to predict the U and V 8x8 chroma blocks. Nine prediction modes of 4x4 luma block are shown in Figure 1(a) and Figure (b). In addition, the prediction block is calculated based on the samples labeled A-M.

2.2 Inter mode in H.264/AVC

Inter-prediction is to reduce the temporal correlation with help of motion estimation and compensation. In H.264, the current picture can be partitioned into the MBs or the smaller blocks. A MB of 16x16 luma samples can be partitioned into smaller block sizes up to 4x 4. There are altogether conceptually 7 different block sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4) that are used in a MB that is encoded in inter mode. These block sizes can be classified as 16x16, 16x8, and 8x16 and P8x8. Each 8x8 block of P8x8 MB can be one of the subtypes such as 8x8, 8x4, 4x8 or 4x4. Figure 2 shows the different block sizes in a MB of inter mode. The smaller block size requires larger number of bits to signal the motion vectors and extra data of the type of partition, however the motion compensated residual data can be reduced. Therefore, the choice of partition size depends on the input video characteristics. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas.

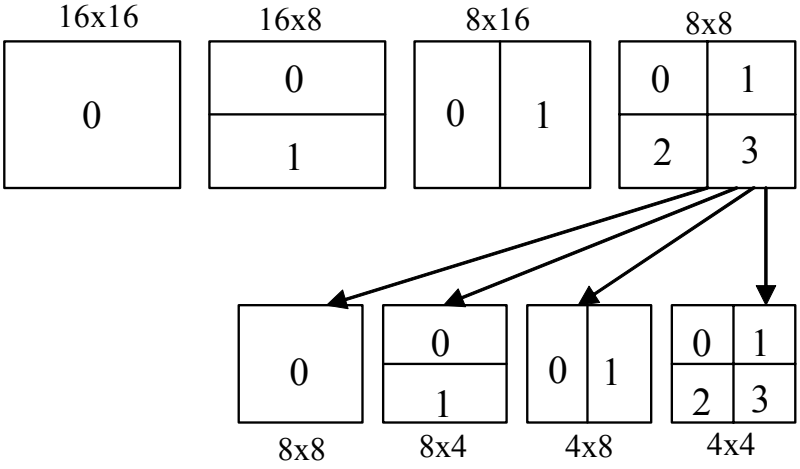


Fig. 2. Variable block sizes of macroblock of INTER mode

2.3 Rate Distortion optimized mode decision

To take the full advantages of all modes, the H.264 encoder can determine the mode that meets the best RD tradeoff using RD optimization mode decision scheme. The optimization approach is based on the assumption that the distortion and rate incurred in coding a MB are independent each other. Let us denote S_t as a block of any rectangular size in a frame at time t; while $C_{t-\tau}$ is a reconstructed block of the same block size as S_t located in the

previously coded frame at time $t-\tau$ ($\tau = 0$ in intra frame coding). Then the best mode for every block that produces the minimum rate-distortion cost is given by

$$J_{RD}(S_t, C_{t-\tau}, m \mid QP, \lambda_m) = SSD(S_t, C_{t-\tau}, m \mid QP) + \lambda_m R(S_t, C_{t-\tau}, m \mid QP) \quad (1)$$

where QP is the MB quantization parameter, λ_m is the Lagrangian multiplier and m is the candidate mode. In (Sullivan & Wiegand, 1998), a strong connection between the local Lagrangian multiplier and the QP was found experimentally as

$$\lambda_m = 0.85 \times 2^{(QP-12)/3} \quad (2)$$

In (1), SSD is the sum of squared difference between original block and reconstructed block, defined separately in terms of intra and inter frame coding

$$SSD_{\text{intra}}(S_t, C_t, m \mid QP) = \sum_i \sum_j \left| S_t(i,j) - C_t(i,j,m \mid QP) \right|^2 \quad (3)$$

$$SSD_{\text{inter}}(S_t, C_{t-\tau}, m \mid QP) = \sum_i \sum_j \left| S_t(i,j) - C_{t-\tau}(i + mv_i, j + mv_j, m \mid QP) \right|^2 \quad (4)$$

where (i, j) represent the ith and jth element and (mv_i, mv_j) represents the motion vector in the inter-frame case.

The R in (1) reflects the number of bits associated with choosing the mode which includes the bit consumption of quantized transform coefficients, motion vector data and the header. Thus R can be written as

$$R = R_{header} + R_{motion} + R_{res} \quad (5)$$

where R_{header} and R_{motion} means the number of bits need for header information and motion vectors respectively. It should be noted that in case of intra frame coding $R_{motion} = 0$. R_{res} indicates the number of bits need to encode a quantized residual block.

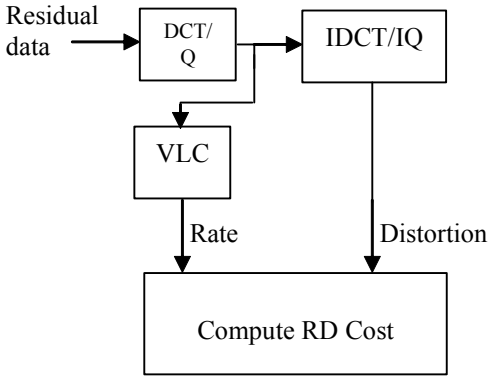


Fig. 3. Computation of RD cost

In intra-frame coding, the final mode decision is selected by the member (either from *I4MB* or *I16MB*) that minimizes the Lagrangian cost in (1). In inter-frame coding, motion estimations with 7 different block-size patterns, as well as the other members in three members (*I4MB*, *I16MB*, and *SKIP*), are calculated. The final decision is determined by the mode that produces the least Lagrangian cost among the available modes. Figure 3 shows the computational process of RD cost of H.264 video coding standard. As indicated in Figure 3, in order to compute RD cost for each mode, same operation of forward and inverse transform/quantization and variable length coding is repetitively performed. All of these processing explains the high complexity of RD cost calculation.
In order to calculate RD cost using (1), 4×4 integer *DCT*+*Q*, CAVLC, and *(DCT*+*Q)-1* in units of 4×4 blocks should be performed. If the amount of RD cost computation in P8×8 mode and Inter16×16 mode and I4MB mode in a MB unit are calculated, it is shown that those computations in Inter16×16 mode are performed 16 times, while those computations in P8×8 mode are performed 64 times and those computations in I4MB mode are performed 144 times. To reduce computation, several fast mode-decision approaches proposed in (Feng et al., 2005; Kim et al., 2006; Yang et al., 2004) focused on how to eliminate unnecessary modes. However, it is noted that the remaining modes should be performed RD optimization. So it still introduces a great deal of computation complexity because of the transform, quantization/inverse quantization and entropy coding to get distortion and bit-rate. A bit rate estimator by modeling the coded bits consumption as a function of the number and levels of the nonzero quantized transform coefficients is introduced in (Yu et al., 2006). In (Tseng et al., 2006), a rate predictor for 4x4 quantized residual blocks is proposed as follows

$$R_{e(res)} = \alpha T_c - \beta T_o + 4P \tag{6}$$

where T_c is the total number of non-zero coefficients, T_o is the number of trailing +/-1 values, α and β are the positive constants and the P equal to 0 for the probable mode and 1 for the other modes. From simulation it is shown that this method cannot achieve very good rate estimation. This is because only two parameters are not enough to estimate the bit rate. In this chapter a rate prediction scheme is introduced based on the properties of CAVLC entropy coding method. Before we propose the new estimation method, review of the CAVLC encoding method is described at following section.

3. Review of Context based adaptive variable length coding (CAVLC)

Context based Adaptive Variable Length Coding (CAVLC) is designed to take advantage of several characteristics of quantized 4x4 blocks. The block is encoded by five syntax elements. These elements are described as follows:

1. *coeff_token:* Both the total number of nonzero coefficients and the number of trailing +/- 1s are coded as combined event. If number of trailing +/- 1s is greater than 3, last 3 trailing +/-1s of zig-zig ordered block are consider as trailing +/-1s and remaining trailing +/-1s are consider as normal coefficient. One out of 4 VLC tables is used (ITU-T Rec., 2002). Since the number of non-zero coefficients in neighboring blocks are usually correlated, the selection of VLC table is depends on the number of non-zero coefficients in neighboring blocks (Richardson, 2003).

2. *Sign of trailing +/- 1:* One bit is used to signal sign information of trailing +/- 1s. 0 is used for positive and 1 is used for negative.
3. *Level:* The magnitude (level) of non-zero coefficients gets larger near the DC coefficient and gets smaller around the high frequency coefficients. CAVLC takes advantage of this by making the choice of the VLC look-up table for the level adaptive in a way where the choice depends on the recently coded levels (Richardson, 2003). One out of 7 VLC tables is used to encode the level information (ITU-T Rec., 2002).
4. *Total_zeros:* The codeword Total_zeros is the number of zeros between the last non-zero coefficient of the zig-zag scan and its start. One out of 15 VLC tables is chosen based on the number of non-zero coefficients (ITU-T Rec., 2002).
5. *Run_before:* Run_before means the number of preceding zeros before each non-zero coefficient and Zeros_left called the number of zeros left before each non-zero coefficient. Based on Run_before and Zeros_left, one out of 7 VLC tables is used to encode this element (ITU-T Rec., 2002).

Element	Value	Code
coeff_token	TotalCoeffs=6, T1s=3 (use VLC0)	00000100
T1 sign (5)	+	0
T1 sign (4)	-	1
T1 sign (3)	+	0
Level (2)	+1 (use Level_VLC0)	1
Level (1)	-2 (use Level_VLC1)	011
Level (0)	+4 (use Level_VLC1)	00010
TotalZeros	2	111
run_before(5)	ZerosLeft=2; run_before=0	1
run_before(4)	ZerosLeft=2; run_before=0	1
run_before(3)	ZerosLeft=2; run_before=1	01
run_before(2)	ZerosLeft=1; run_before=1	0
run_before(1 & 0)	ZerosLeft=0; run_before=0	No code required

Table 1. Example of CAVLC

For example, the coefficients in the zig-zag order of a 4x4 block are [4,-2,0,1,0,1,-1,1,0,0…]. Here total number of non-zero coefficients (T_c) is 6, total number of zero before the last nonzero coefficients (T_z) is 2, number of trailing one's (T_o) is 3. Based on the VLC tables (ITU-T Rec., 2002) as shown in Table 1, the transmitted bit stream for this block is 000001000101011000101111101 0.

4. Proposed Fast Bit Rate estimation Method

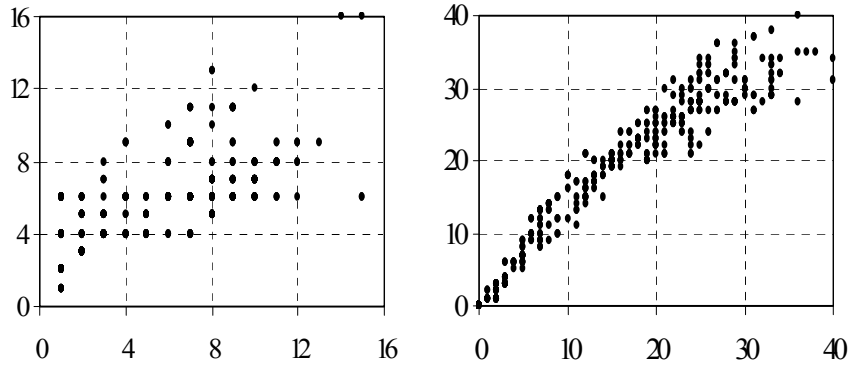
To estimate the bits for quantized transform coefficients, we estimate the number of bits for each of five different types of symbols of CAVLC separately.

1. *The coefficient token (the number of coefficients, the number of trailing ones):* Four VLC tables are used for encoding coefficient token. Selection of VLC table is context adaptive.

Figure 4(a) shows the plot of actual bit rate to encode the coefficient token versus the number of coefficients of foreman video sequence at QP=28. Similar results were found for other video sequences. It is clearly shown that bit consumption to encode the coefficient token is increased with number of coefficients. Based on VLC tables (ITU-T Rec., 2002), it is also shown that bit rate for coefficient token is decreased with number of trailing ones. Based on this criteria, we propose the number of bits require to encode the coefficient token is

$$R_{coff} = w_1 T_c - w_2 T_o + w_3 \tag{7}$$

where T_c and T_o are same as equation (6). These w_1, w_2 and w_3 are weighting constants. In order to set the weighting factors, we have done several experiments for different video sequences (akiyo, foreman, stefan, mobile, table tennis, paris) with QCIF format at different QP values. We have observed rate-distortion performance of these video sequences at different combinations of weighting factor. Better rate-distortion performance was found at $w_1 = w_2 = 1$, $w_3 = 0$.



(a) (b)

Fig. 4. Plot of (a) no. of non-zero coefficients vs true value of coefficient token (X-axis: T_c, Y-axis: True rate of Coeff_token) (b) SAT_l vs actual rate of level (X-axis: SAT_l, Y-axis: True rate of level)

2. *The sign of trailing ones:* For each T_o, a single bit encodes the sign (0=+, 1=-). So bit consumption to encode the trailing ones as follows:

$$R_{trail1} = T_o \tag{8}$$

3. *The level of nonzero coefficients:* From the observation of level-VLC tables (ITU-T Rec., 2002), it is shown that bit requirement is increased with magnitude of non-zero coefficients. Number of bits to encode the level information is proposed as follows:

$$R_{level} = w_4 SAT_l \tag{9}$$

with the SAT_l given by

$$SAT_l = \sum_{k=1}^{Tc} |L_k| \tag{10}$$

where $|L_k|$ is the absolute value of k_{th} non-zero coefficient, SAT_l is the sum of absolute values of all levels of quantized transform residual block. w_4 is a positive constant. Suppose two coefficients are encoded using same level_VLC table. If the magnitude of first coefficient $(|L_1|)$ is larger than that of second coefficient $(|L_2|)$, from level_VLC table it is shown that rate for first coefficient $R(L_1)$ also greater than that of second coefficient $R(L_2)$. So equation (9) is valid. Let us consider two coefficients are encoded using multiple level_VLC tables. If the earlier coefficient is not larger than the other coefficient, by observing the all level_VLC tables (ITU-T Rec., 2002) there is $R(L_1) > R(L_2)$ if $|L_1| > |L_2|$. In some times, $R(L_1) > R(L_2)$ if $|L_1| < |L_2|$ but fortunately this event does not occur high frequently and is slightly influence the estimation result. Figure 4(b) shows the plot of actual bit rate to encode the level information with SAT_l of foreman video sequence at QP=28. Similar results were found for other video sequences. By changing the value of w_4, we have observed RD performance of different sequences. Better results were found with w_4=1.

4. *Encode the total number of zeros before the last coefficient:* From the observation of total zero VLC tables, it is shown than bit consumption to encode the total zero is increased with number of total zero. So we can propose the estimated bits for total zero is as follows

$$R_{zero} = w_5 T_z \tag{11}$$

where w_5 is a positive constant and T_z is the total number of zeros before the last non zero coefficients. Here w_5=1, which is found in similar way of w_4.

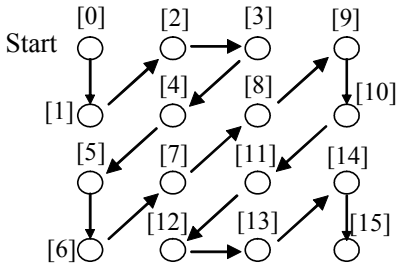


Fig. 5. Zig-zag scan and corresponding frequency of 4x4 luma block

5. *Encode each run of zeros:* After the DCT transform, the high frequency coefficient usually has small energy. By quantization, more zeros are found at the high frequency position of quantized transform block. So the value of *Run* for the high frequency non-zero coefficients is larger. From the observation of run VLC tables, it is shown that more bits

are required for large value of run. So bit consumption is higher to encode the run of high frequency non-zero coefficients. Based on this idea, we propose the rate for run of each non-zero coefficients as follow:

$$R_{run_{(k)}} = w_6 f_k \ , \ 0 \geq f_k \leq 15 \tag{12}$$

where, f_k is the frequency of k_{th} non-zero coefficient of recorded block and w_6 is the positive constant. For example, given a string of coefficients [0, 3, 0, 1,-1,-1, 0, 1, 0, 0…], frequency of first non-zero coefficient (3) is 1 and frequency of last non-zero coefficient (1) is 7. The weighting factor w_6=0.3 is found in similar way of w_4. Figure 5 shows the zig-zag scan of a residual block with corresponding value of frequency.

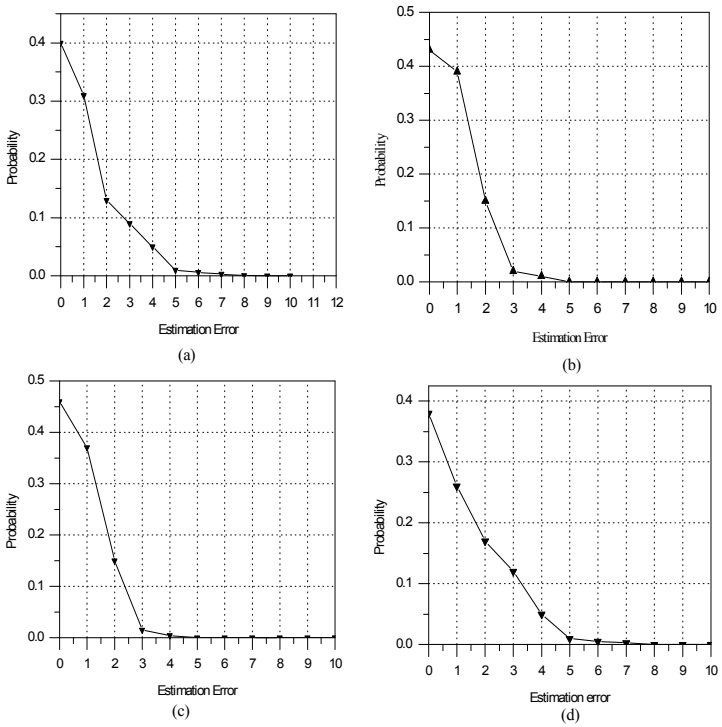


Fig. 6. Probability of estimation error of (a) Coeff_token (b) level (c) total zero (d) run

From above analysis, we have estimated the bits needed to encode a 4x4 residual block ($R_{est_{(res)}}$) is

$$\begin{aligned} R_{est_{(res)}} &= R_{coeff} + R_{trail1} + R_{level} + R_{zero} + \sum_{k=1}^{T_c} R_{run_{(k)}} \\ &= w_1 T_c - w_2 T_o + w_3 + T_o + w_4 SAT_l + w_5 T_z + \sum_{k=1}^{T_c} (w_6 f_k) \end{aligned} \tag{13}$$

By putting the values of different constants, the proposed rate estimator becomes

$$R_{est_{(res)}} = T_c + T_z + SAT_l + 0.3\sum_{k=1}^{T_c} f_k \tag{14}$$

Figure 6 shows the probability of estimation error of four different types of symbols. Estimation error of the symbol is the absolute difference between actual bit rate and estimated bit rate of that symbol. Y-coordinate of Figure 6 is the probability of corresponding estimation error. It is shown that most of the estimation errors of each symbol are between 0 to 4 and the probability of each symbol that the estimated rate perfectly match with CAVLC is about 40%.

5. Simulation Results

To verify the proposed technique, JM 8.3 reference software is used in simulation. Six well known video sequences are used as test materials. The test conditions are as follows:

a) Hadamard transform is used
b) RD optimization is enabled
c) CAVLC is enabled.
d) Frame rate is 30
e) MV search range is ±32 pels for QCIF and CIF
f) Fast motion estimation algorithm is used (Chen et al., 2002).

A group of experiments were carried out on the test sequences with different quantization parameters. Comparison results were produced based on the percentage of difference of coding time (Δ T %), the PSNR difference (ΔP_{snr}) and percentage of the bit rate difference (Δ Bit %). In order to evaluate complexity reduction, Δ T (%) is defined as follows

$$\Delta T = \frac{T_{original} - T_{proposed}}{T_{original}} \times 100\% \tag{15}$$

where, $T_{original}$ denotes the total encoding time of the JM 8.3 encoder with rate distortion optimization and $T_{proposed}$ is the total encoding time with proposed fast rate estimation technique.

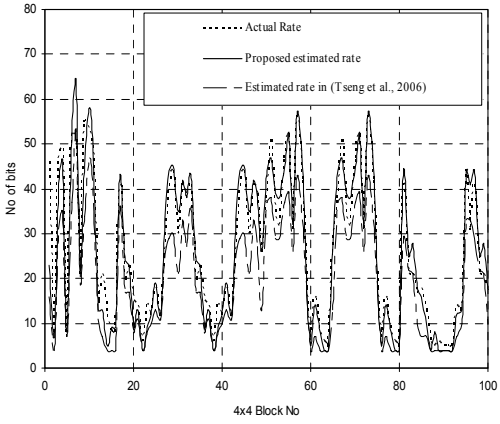


Fig. 7. Comparison of our proposed method with rate estimation method described in (Tseng et al., 2006)

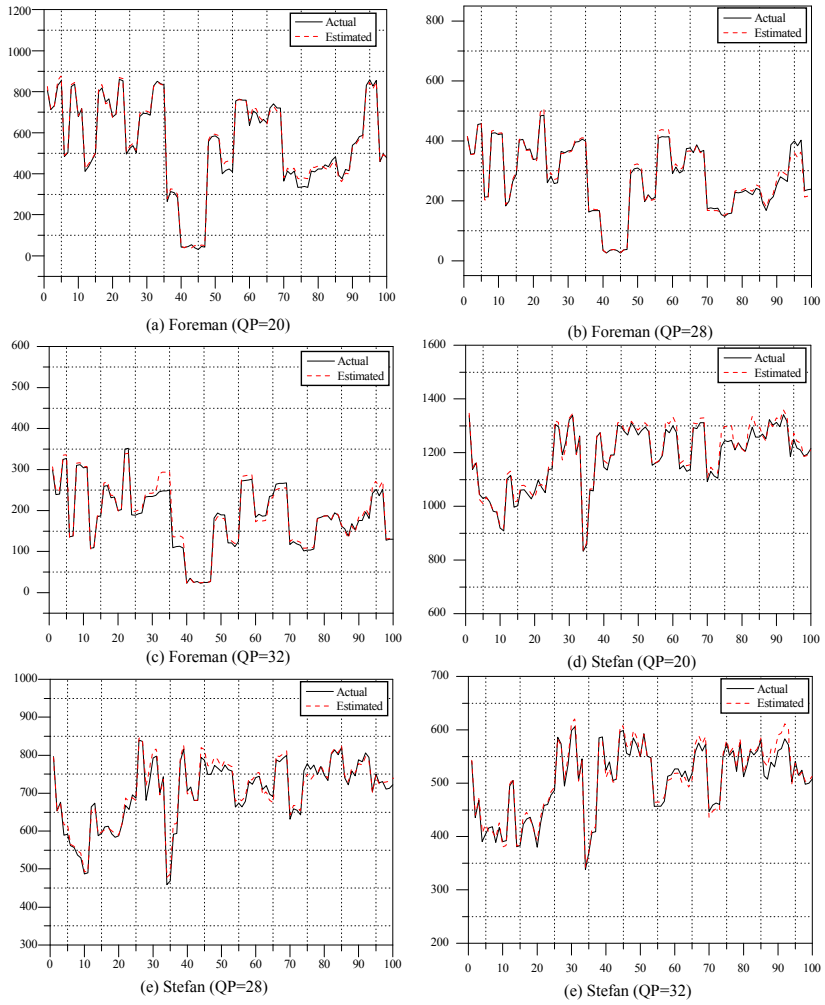


Fig. 8. Curves of the estimated and the actual rates of first 100 macroblocks of I frame of intra coding of foreman and Stefan sequences (X-axis: Macroblock number, Y-axis: number of bits)

5.1 Experiments with all intra frame sequences

In this experiment, a total number of 100 frames are used for each sequence, and period of I-frame is set to 1, i.e., all the frames in the sequence are intracoded. In (Tseng et al., 2006), a rate predictor for 4x4 intra mode decisions is introduced based on the number of non-zero coefficients and number of trailing ones. Figure 7 shows the comparison of our proposed method with actual rate and the rate predictor described in (Tseng et al., 2006) for the Foreman sequence. Data is collected from the first 100 4x4 block of first frame of IIIIII sequence. QP factor is set as 28. It is shown that the proposed method is very closely

matched with actual rate as compared to rate predictor in (Tseng et al., 2006). By using the proposed estimation method, Figure 8 shows the predicted rates $R_{est}(=R_{header}+R_{motion}+R_{est(res)})$ and actual rates $R(=R_{header}+R_{motion}+R_{res})$ are very closely identical, which are obtained from first 100 MBs of IIIIII sequence of Foreman-QCIF and Stefan_CIF at three different (20, 28, 32) QP values. The proposed estimation achieves a precise prediction in intra frame coding.

Sequence	QP	ΔP_{snr}	Δ Bit%
Akiyo (QCIF)	20	-0.06	+0.64
	24	-0.05	+0.50
	28	+0.02	+0.53
	32	+0.17	+0.70
	36	+0.12	+1.49
	40	+0.16	+2.59
Foreman (QCIF)	20	-0.06	+0.49
	24	-0.05	+0.89
	28	-0.02	+1.16
	32	+0.10	+1.75
	36	+0.11	+2.35
	40	+0.15	+3.86
Mobile (QCIF)	20	-0.16	+0.85
	24	-0.09	+0.78
	28	-0.10	+0.54
	32	-0.08	+0.27
	36	-0.06	+0.70
	40	+0.03	+1.83
Paris (CIF)	20	-0.16	+0.46
	24	-0.11	+0.25
	28	-0.10	+0.45
	32	-0.06	+0.88
	36	-0.02	+0.64
	40	+0.04	+1.61
Table Tennis (CIF)	20	-0.14	+0.31
	24	-0.12	+0.36
	28	-0.04	+1.01
	32	+0.01	+0.88
	36	+0.02	+1.48
	40	+0.06	+2.23
Stefan (CIF)	20	-0.19	+0.62
	24	-0.12	+0.44
	28	-0.11	+0.42
	32	-0.02	+0.68
	36	-0.02	+1.69
	40	+0.13	+3.15

Table 2. Performance of PSNR and bit rate of proposed plgorithm while all frames are intra coded

Sequence	Quantization Parameter, QP					
	20	24	28	32	36	40
Akiyo_QCIF	51.21 %	47.29 %	44.11 %	39.34 %	35.35 %	30.42 %
Foreman_QCIF	55.20 %	51.16 %	46.66 %	42.42 %	37.28 %	33.33 %
Mobile_QCIF	62.87 %	61.78 %	59.45 %	57.14 %	51.21 %	47.14 %
Paris_CIF	58.11 %	54.96 %	52.81 %	47.94 %	43.30 %	41.08 %
Table_tennis_CIF	57.04 %	53.56 %	49.54 %	43.41 %	39.11 %	35.21 %
Stefan_CIF	57.41 %	54.12 %	50.55 %	44.96 %	39.20 %	34.52 %
Average = 47.36 %						

Table 3. Computational complexity reduction of proposed algorithm while all frames are intra coded

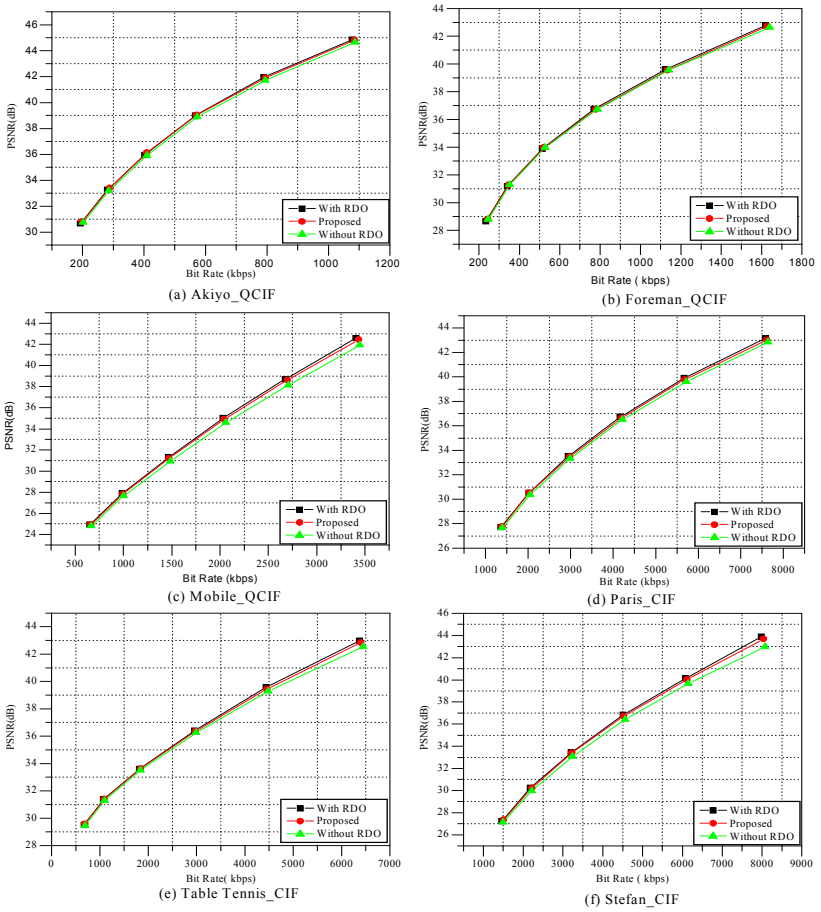


Fig. 9. Rate-distortion performance of proposed rate estimation method of different video sequences while all frames are intra coded.

To evaluate the rate distortion performance six different sequences (Akiyo, Foreman, Mobile, Paris, Table tennis and Stefan) are used in simulation. As shown in Table 2, it is clear that PSNR loss and bit rate increment is negligible. Figure 9 shows the rate-distortion curves of different sequences. The proposed method is very close with RD optimized curve. Comparing with the original H.264/AVC encoder with RD optimization, the proposed algorithm achieves about 47% time reduction of total encoding time on average. As shown in Figure 10, for the sequence "Mobile" and "Paris", the coding speed is high because both the sequences contains high detail such as different books in bookshelf of "Paris". On the other hand sequence "Akiyo" show strong spatial homogeneity and low detail. Therefore, the time saving of this sequence is not much as compared to other sequences. Figure 10 also indicates that percentage of complexity reduction is decreased with increasing the QP values. This is because there is large number of non-zero coefficients at small QP values. If true encoding process is used large entropy coding time is spent at small QP values whereas in our proposed method no entropy coding is required during mode-decision process.

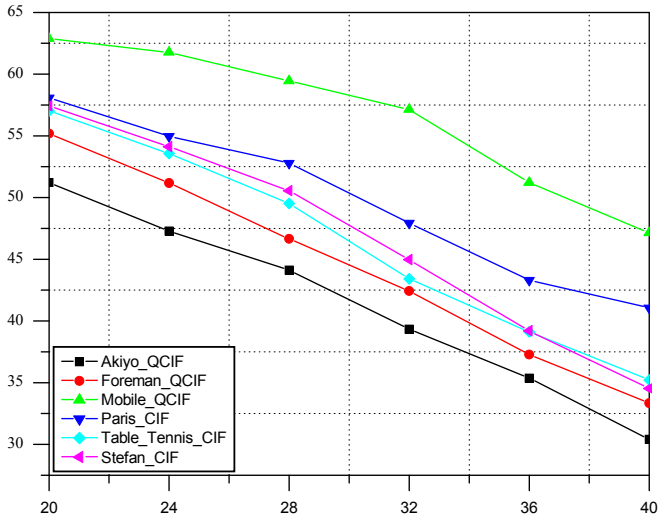


Fig. 10. Complexity reduction of proposed algorithm during intra frame coding (X-axis: QP; Y-axis: % of complexity reduction)

5.2 Experiments with IPP sequences

To evaluate the performance of proposed method during inter frame coding, several test video sequences are used. In this experiment total number of frames is 100 for each sequence, and the period of I-frame is 3. Figure 11 shows the actual and predicted rates of foreman and Stefan at different QP values. Data is generated as similar ways of intra frame coding.

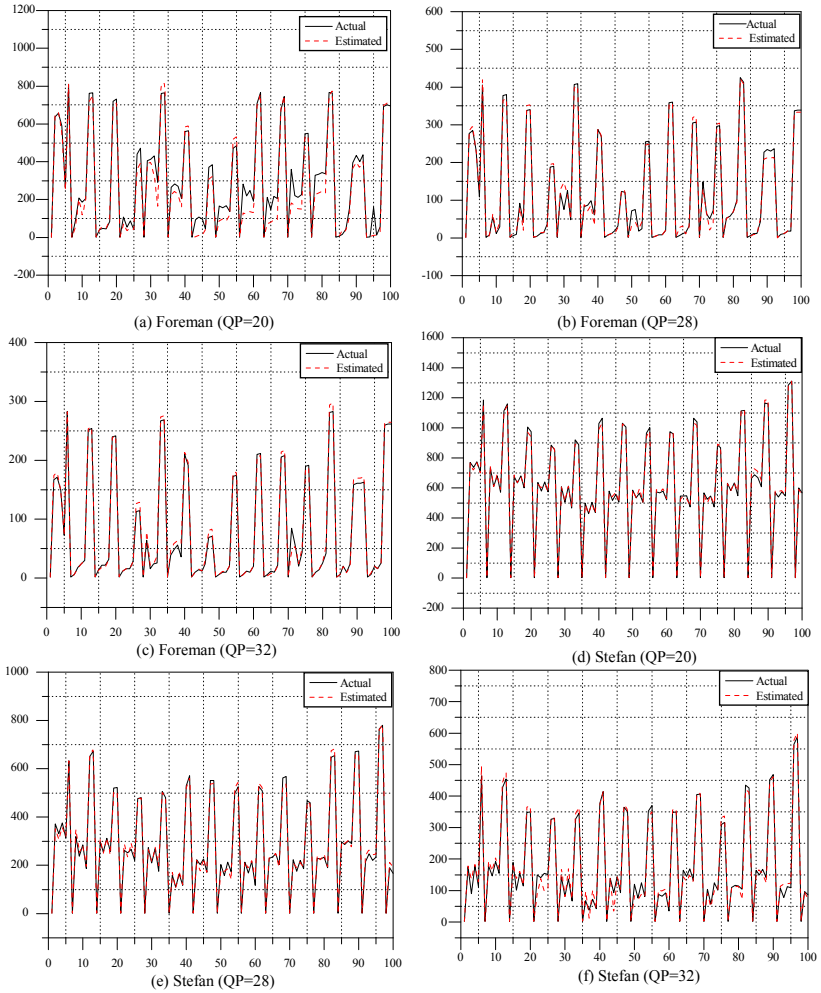


Fig. 11. Curves of the estimated and the actual rates of first 100 macroblocks of P frame of inter coding of foreman and Stefan sequences (X-axis: Macroblock number, Y-axis: number of bits)

Experimental results are tabulated as Table 4 which means that PSNR reduction and bit rate increments are negligible. The positive values mean increments whereas negative values mean decrements. From the experimental results in Table 5, it is observed that the proposed approach has reduced the encoding time by 34% on average. RD performances are given in Figure 12 which shows that the proposed method is closely matched with actual RD curves. Figure 13 presents the plot of computational complexity with quantization parameter of different sequences. Figure 13 show the general tendency that time saving increases as QP decreases. This is understandable that at small QP, coding quality is high and many detailed

are retained. From this figure it is also shown that computation saving is large for those sequences which have high motion and large detail.

Sequence	QP	ΔP_{snr}	Δ Bit%
Akiyo (QCIF)	20	+0.01	+1.47
	24	+0.01	+1.38
	28	+0.09	+1.97
	32	+0.14	+2.79
	36	+0.11	+3.00
	40	+0.16	+2.89
Foreman (QCIF)	20	+0.02	+2.12
	24	+0.03	+2.23
	28	+0.05	+2.57
	32	+0.07	+3.01
	36	+0.13	+3.05
	40	+0.15	+4.44
Mobile (QCIF)	20	-0.05	+0.95
	24	-0.04	+0.99
	28	-0.03	+1.17
	32	-0.01	+2.39
	36	+0.02	+2.89
	40	+0.07	+2.94
Paris (CIF)	20	-0.05	+1.17
	24	-0.05	+1.37
	28	-0.02	+1.43
	32	+0.00	+2.38
	36	+0.02	+2.33
	40	+0.08	+3.21
Table Tennis (CIF)	20	-0.03	+2.20
	24	+0.01	+2.33
	28	+0.04	+2.52
	32	+0.06	+2.88
	36	+0.06	+2.95
	40	+0.06	+3.11
Stefan (CIF)	20	-0.10	+1.17
	24	-0.09	+0.08
	28	-0.03	+1.97
	32	-0.02	+3.01
	36	+0.07	+3.65
	40	+0.14	+4.74

Table 4. Performance of PSNR and Bit Rate of Proposed Algorithm of Inter frame (IPP sequences) coding

Sequence	Quantization Parameter, QP					
	20	24	28	32	36	40
Akiyo_QCIF	36.63 %	31.91 %	28.73 %	23.45 %	22.07 %	21.33 %
Foreman_QCIF	41.66 %	37.85 %	33.33 %	29.47 %	27.58 %	22.22 %
Mobile_QCIF	53.08 %	50.99 %	46.71 %	40.49 %	34.57 %	29.89 %
Paris_CIF	45.15 %	41.82 %	38.39 %	33.98 %	28.64 %	24.29 %
Table_tennis_CIF	41.61 %	36.59 %	32.45 %	28.16 %	25.94 %	23.87 %
Stefan_CIF	46.19 %	42.65 %	38.61 %	35.13 %	30.56 %	25.14 %
Average = 34.20 %						

Table 5. Computational complexity reduction of proposed algorithm during Inter frame (IPP sequences) coding

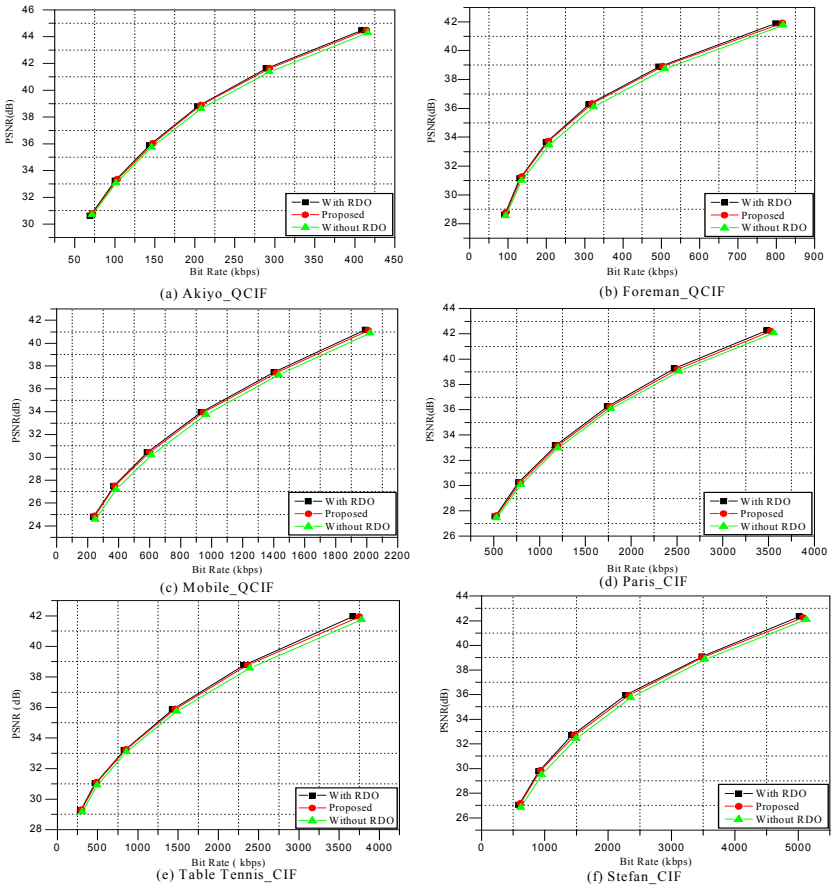


Fig. 12. Rate-distortion performance of proposed rate estimation method of different video sequences during Inter frame (IPP sequences) coding

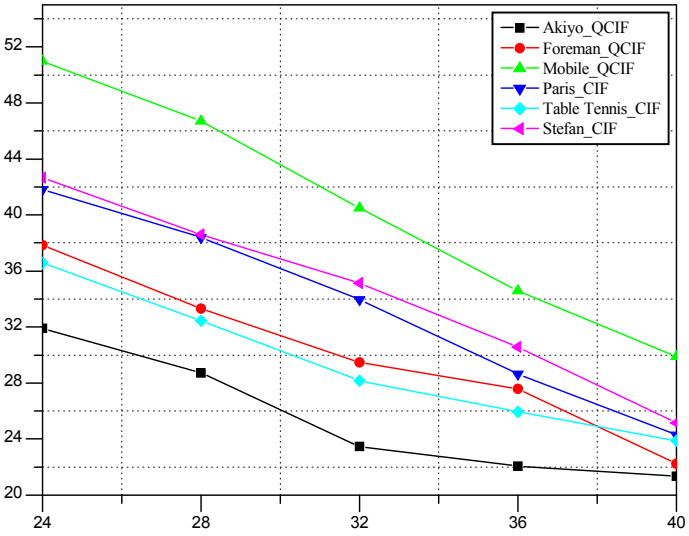


Fig. 13. Complexity reduction of proposed algorithm during IPP sequences (X-axis: QP; Y-axis: % of complexity reduction)

Sequence	Frame Skip	QP	Δ P_{snr}	Δ Bit%	Δ T%
Akiyo (QCIF)	1	20	+0.09	+3.84	30.41
		28	+0.14	+3.00	20.84
		32	+0.06	+3.02	17.76
	2	20	+0.11	+2.85	33.20
		28	+0.14	+3.55	25.99
		32	+0.10	+3.67	20.46
Foreman (QCIF)	1	20	+0.09	+3.24	36.86
		28	+0.11	+3.96	29.60
		32	+0.10	+2.98	23.88
	2	20	+0.06	+2.03	34.51
		28	+0.10	+4.13	26.25
		32	+0.13	+4.41	20.27
Mobile (QCIF)	1	20	+0.01	+3.04	46.26
		28	+0.11	+2.82	39.67
		32	+0.08	+3.33	35.35
	2	20	+0.04	+1.60	47.46
		28	+0.11	+3.10	40.06
		32	+0.08	+2.87	36.51

Table 6. Experimental results of IBPBP sequences

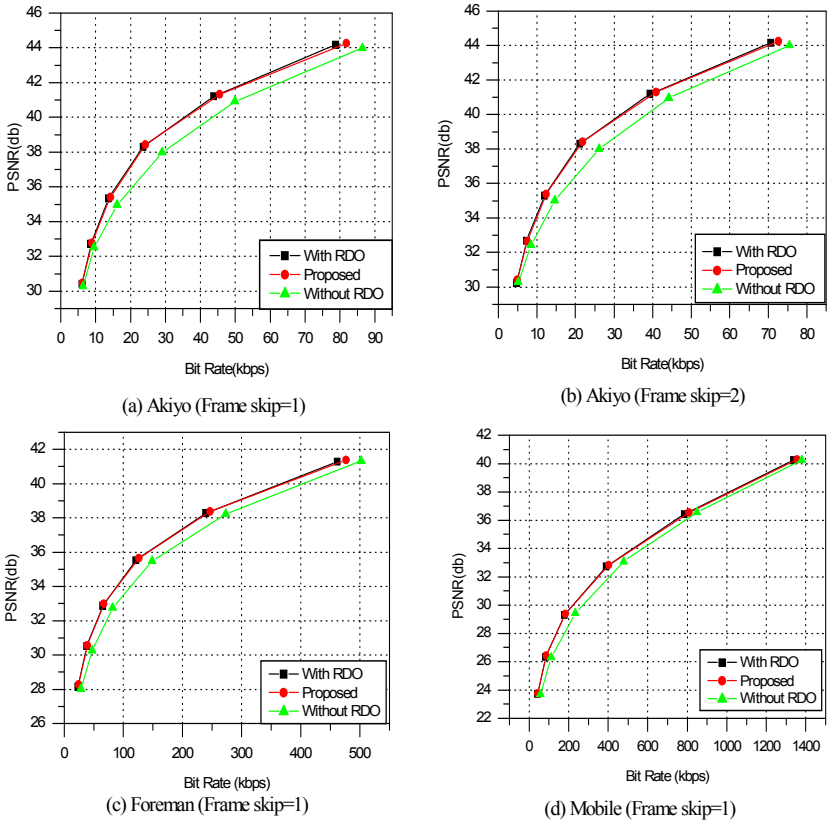


Fig. 14. Rate distortion performance of proposed method with IBPBP sequences

5.3 Experiments with IBPBP sequences

In this experiment, there is one B-frame between any two I- or P-frames with different frame skip values. Period of I-frames is set to 100. Number of frames is 100. Experimental results were tabulated at table 6. As shown in table 6, the proposed algorithm achieves time saving of about 32% (on average) with slight increment in bit rate. Rate-distortion performances of different sequences are shown in Figure 14 with different frame skip value.

5.4 Experiments with Full Search Motion Estimation

It is well known that motion estimation requires major portion of the processing power. In all of the previous experiments, fast motion estimation technique described in (Chen et al., 2002) is utilized. In order to show the complexity of proposed method with full search motion estimation technique, several video sequences are encoded. In this experiment, IPP sequence is used and number of frames is set to 50. Experimental results are tabulated at table 7. It is shown that bit rate increment is up to about 4%. Since the PSNR also increases, the resulting RD performance is very much closed with original one. The proposed

algorithm reduces about 17% (on average) of computation time when full search motion estimation method is used.

Sequence	QP	ΔP_{snr}	Δ Bit%	Δ T%
Akiyo (QCIF)	20	+0.09	+3.96	12.53
	28	+0.12	+2.94	6.66
	32	+0.13	+3.65	5.72
Foreman (QCIF)	20	+0.07	+2.86	20.66
	28	+0.04	+3.59	9.09
	32	+0.08	+3.50	6.62
Mobile (QCIF)	20	-0.01	+0.64	32.19
	28	+0.11	+2.52	24.42
	32	+0.10	+4.21	21.95
Stefan (QCIF)	20	+0.02	+1.48	24.80
	28	+0.07	+2.46	20.11
	32	+0.08	+2.57	15.78

Table 7. Experimental results with full search motion estimation

Sequence	QP	ΔP_{snr}	Δ Bit%	Δ T%
Akiyo (QCIF)	20	+0.02	-0.91	-4.11
	28	+0.01	-1.38	-4.54
	32	+0.05	-1.47	-4.02
Foreman (QCIF)	20	+0.05	-1.14	-4.47
	28	+0.07	-0.77	-4.61
	32	+0.11	-0.67	-4.68
Mobile (QCIF)	20	+0.22	-0.29	-5.86
	28	+0.13	-0.78	-5.17
	32	+0.06	-1.02	-5.02
Stefan (CIF)	20	+0.33	-0.52	-5.59
	28	+0.10	-0.88	-5.12
	32	+0.11	-1.23	-4.92

Table 8. Comparison of proposed method with rate estimation method stated in (Tseng et al., 2006)

5.5 Comparison with other methods

In this experiment, the proposed method is compared with two different methods in terms of RD performance and complexity. Fast motion estimation technique described in (Chen et al., 2002) is used. Table 8 shows the comparison of our proposed method with rate estimation method defined in (Tseng et al., 2006). In this simulation, only intra 4x4 modes are used. RD performance of proposed method is better from method (Tseng et al., 2006). Positive value of Δ T% means complexity reduction and negative value means increment of complexity. It is shown that as compared with (Tseng et al., 2006), this algorithm reduces about 1% of bit rate and increases about 0.10 db PSNR with slight increment (about 4%) of computational time. Table 9 shows the RD performance as well as complexity reduction of

this algorithm as compared with fast inter mode decision algorithm described in (Lim et al., 2003). IPP sequences are used in this comparison. Complexity reduction is about 14% (on average) for medium and high motion i.e. foreman, mobile and stefan sequences. It is also shown that the proposed method increases about 7% (on average) of computation of low motion sequence such as akiyo as compared with mode decision method stated in (Lim et al., 2003). RD performance is better for all types of sequences.

Sequence	QP	ΔP_{snr}	Δ Bit%	Δ T%
Akiyo (QCIF)	20	+0.05	-0.24	-10.71
	28	+0.21	-0.11	-6.56
	32	+0.15	-0.67	-4.16
Foreman (QCIF)	20	+0.10	-0.49	9.09
	28	+0.11	-0.95	7.93
	32	+0.12	-1.48	6.12
Mobile (QCIF)	20	+0.03	-0.39	31.53
	28	+0.05	-0.42	21.50
	32	+0.04	-0.84	14.28
Stefan (CIF)	20	+0.08	-0.35	19.60
	28	+0.04	-0.60	10.94
	32	+0.09	-1.03	8.34

Table 9. Comparison of proposed method with fast inter mode decision method stated in (Lim et al., 2003)

QP	Akiyo (QCIF)		Foreman (QCIF)		Stefan(QCIF)	
	$\Delta Rate$ %	$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$
20	1.00	-0.06	1.30	0	0.14	-0.14
24	2.16	-0.02	1.68	+0.04	0.61	-0.15
28	2.57	+0.04	2.68	+0.08	1.34	-0.08
32	2.23	-0.01	3.68	+0.09	1.85	-0.06
36	3.39	0	3.88	+0.12	2.92	-0.02
40	3.22	-0.03	4.41	+0.06	4.09	+0.01

Table 10. Experimental results with CABAC entropy coding method

QP	Akiyo (QCIF)	Foreman (QCIF)	Stefan(QCIF)
20	32.20	40.58	46.25
24	27.78	39.13	43.42
28	23.53	25.45	37.68
32	22.45	25.00	35.48
36	15.22	20.41	29.82
40	11.36	17.02	24.53

Table 11. Percentage of complexity reduction with CABAC

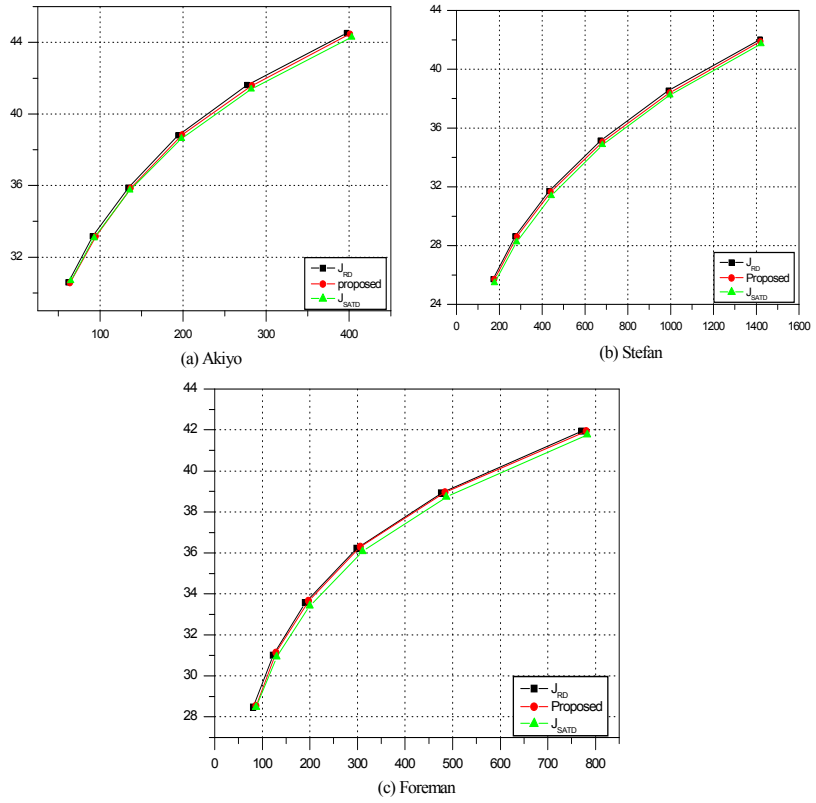


Fig. 15. RD performance with CABAC entropy coding method

5.6 Experiments with CABAC entropy coding method

In all of the above experiments CAVLC entropy coding method was used. Although the proposed algorithm is developed based on CAVLC, it is also suitable during mode decision of CABAC. This is because the proposed algorithm is used only for RD optimization. After selection of best mode, true entropy coding is used. To justify the performance of proposed rate estimation with CABAC entropy coding method, some video sequences are tested. IPP sequence is used in simulations and number of frames was set to 50. Table 10 shows the experimental results in terms of PSNR and bit rate. Table 11 shows the complexity reduction is also significant. The rate-distortion performance is given in Figure 15, which shows that the proposed algorithm is also suitable while CABAC entropy coding method is used. The proposed method is very close to RD optimized curves.

6. Conclusion

In this chapter, simple and fast bit rate estimation method for mode decision of H.264/AVC is proposed. This method is based on the VLC tables used in CAVLC entropy coding

method. The experimental results verified that the proposed technique is suitable both for inter and intra mode decision of H.264/AVC. With the proposed scheme, entropy coding can be skipped during the mode decision process. The proposed technique reduces encoding time by 47 %, 34% and 32 % on average during intra frame, IPP sequences and IBPBPBP sequences respectively. The RD performance of this algorithm is better than both methods stated in reference (Tseng et al., 2006) and (Lim et al., 2003). The proposed method is also suitable for mode decision while CABAC entropy coding method is utilized.

7. Acknowledgement

The work described in this chapter was substantially supported by a GRF grant from Hong Kong SAR Government with project number of 9041251 (CityU 119207).

8. References

Chiang T. & Zhang Y. Q. (1997) A new rate control scheme using quadratic rate distortion model, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, No. 1, February 1997, pp. 246-250, ISSN 1051-8215

Coebera J. R. & Lei S. (1999). Rate control of DCT video coding for low-delay communication, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, No. 1, February 1999, pp. 172-185, ISSN 1051-8215

Chen Z.; Zhou P. & He Y. (2002) Fast Integer Pel and Fractional Pel Motion Estimation for JVT, *Joint Video Team (JVT) Docs*, JVT-F017, Dec. 2002.

Erol B.; Gallant M.; Cote G. & Kossentini F. (1998). The H.263+ video coding standard: complexity and performance, *Proceedings of IEEE Data Compression Conference*, pp. 259-268, Snowbird, Utah, March 1998.

Feng Pan F.; Lin X.; Rahardja S.; Lim K. P.; Li Z.G.; Wu D. & Wu S. (2005). Fast Mode Decision Algorithm for Intra-prediction in H.264/AVC Video Coding, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 7, July 2005, pp. 813-822, ISSN 1051-8215

Han K. H. & Lee Y. L. (2005). Fast macroblock mode determination to reduce H.264 complexity, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, No. 3, March 2005, pp. 800-804, ISSN 1745-1337

ITU-T Rec. (2002). Advanced Video Coding, ITU-T Rec. H.264/ISO/IEC 11496-10 Final Committee Draft, Documents JVT-E022, September 2002.

Iain E. G. Richardson (2003) *H.264 and MPEG-4 Video Compression – video coding for next generation multimedia*, John Wily & Sons Ltd, ISBN: 0-470-84837-5, West Sussex, England

Joint Video Team (JVT) (2002). Joint Video Specification (ITU-T Rec. H.264/ISO/ISE 14496-10)- Joint Committee Draft, Doc. JVT-G050r1.doc, Dec. 2002.

Kim C.; Shih H.H. & Kuo C. C (2006). Fast H.264 Intra-prediction mode selection using joint spatial and transform domain features, *Journal of Visual Communication and Image Representation*, vol. 17, No. 2, April 2006, pp. 291-310, ISSN 1047-3203.

Lim K. P; Wu S.; Wu D. J; Rahardja S.; Lin X.; Pan F. & Li Z. G. (2003). Fast inter mode selection, *Joint Video Team (JVT) Docs*, JVT-I020, Sep. 2003.

Sullivan G. J. & Wiegand T. (1998). Rate-distortion optimization for video compression, *IEEE Signal Processing Magazine*, vol. 15, No. 6, November 1998, pp. 74-90, ISSN: 1053-5888

Sarwer M. G.; Po L. M. & Wu J. (2008) Fast Sum of Absolute Transformed Difference based 4x4 Intra Mode Decision of H.264/AVC Video Coding Standard, *Elsevier Journal of Signal Processing: Image Communications*, Vol. 23, Issue 8, September 2008, pp. 571-580, ISSN: 0923-5965.

Sarwer M. G. & Wu J. Q. M (2009) Region Based Searching for Early Terminated Motion Estimation Algorithm of H.264/AVC Video Coding Standard, *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 468-471. St John's, NL, May 2009

Topiwala. P.; Sullivan G.; Joch A. & Kossentini F. (2001). Performance evaluation of H.26L TML 8 vs. H.263++ and MPEG4, *Document VCEG-042, 15th Meeting, ITU-T Q.6/SG16*, Pattaya, Thailand, December 2001.

Tseng C. H.; Wang H.M. & Yang J. F. (2006), Enhanced Intra 4x4 Mode Decision for H.264/AVC Coders, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 8, August 2006, pp. 1027-1032, ISSN 1051-8215

Weigand T.; Schwarz H.; Joch A.; Kossentini F. & Sullivan G. (2003). Rate-constrained coder control and comparison of video coding standards, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, No. 7, July 2003, pp. 688-703, ISSN 1051-8215

Wu D.; Pan F.; Lim K. P.; Wu S.; Li Z. G.;Lin X.; Rahardja S. & Ko C. C. (2005). Fast inter mode Decision in H.264/AVC Video Coding, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, No. 6, July 2005, pp. 953-958, ISSN 1051-8215

Yang C. L.; Po L. M. & Lam W. H. (2004). A fast H.264 Intra Prediction algorithm using Macroblock Properties, *Proceedings of IEEE International Conference on Image Processing*, pp. 461-464, Singapore, October 2004.

Yu K.; Yang J. F. &Sun M. T. (2006) Efficient rate-distortion estimation for H.264/AVC coders, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 5, May 2006, pp. 600-611, ISSN 1051-8215

15

Secure Multimedia Streaming over Multipath Wireless Ad hoc Network: Design and Implementation

Binod Vaidya[1], Joel J. P. C. Rodrigues[1,2] and Hyuk Lim[3]
[1]*Instituto de Telecomunicações, Covilhã, Portugal*
[2]*University of Beira Interior, Covilhã, Portugal*
[3]*Gwangju Institute of Science and Technology, Gwangju, Korea*

1. Introduction

Wireless ad hoc networks are becoming increasingly popular as they provide users access to information at anytime and from anywhere. A wireless ad hoc network is a system of wireless mobile nodes that dynamically self-organize in arbitrary and temporary network topologies allowing people and devices to internetwork without any pre-existing communication infrastructure.

Furthermore, with an increase in the bandwidth of wireless channels and computational power of mobile devices, multimedia applications are expected to become more prevalent in wireless ad hoc networks in the near future. Examples of multimedia transmission over wireless ad hoc networks include multimedia streaming, transmitting audio and\or video in the battlefield as well as search and rescue operations after a disaster.

In mobile ad hoc network (MANET), several routing protocols such as Ad hoc On-demand Distance Vector (AODV) (Perkin *et al*., 2003) and Dynamic Source Routing (DSR) (Johnson *et al*., 2001) are widely used. However, the unreliability of the wireless medium and the dynamic topology due to nodes mobility or failure result to frequent communication failures, and high delays for path re-establishments, so single path routing may not be suitable for many applications, such as multimedia streaming and voice over Internet Protocol (VoIP).

In wireless ad hoc network, multipath routing is used to establish multiple paths between each source–destination pair. In fact, a multipath routing is a very promising alternative to single path routing as the former provides higher resilience to path breaks, alleviates network congestion through load balancing and reduces end-to-end delay (Mueller *et al*., 2004). Thus the multipath routing is highly appealing for multimedia streaming over wireless ad hoc networks.

Nonetheless, as security remains important factor that hinders the rapid deployment of multimedia applications over wireless ad hoc networks, security issue must be addressed in multipath multimedia streaming over wireless ad hoc networks.

In this chapter, we provide a generalized framework for secure and reliable multimedia streaming over multipath wireless ad hoc network. The aim of this chapter is to design and

implement a secure multipath routing scheme that can be efficiently utilized for multimedia streaming over wireless ad hoc network. This framework provides security not only for ad hoc routing but also for real-time data transfer. For securing multipath ad hoc routing and real-time data transfer, we have considered not only self-certified public keying technique and self-certificate but also digital signature and encryption technique. Moreover, we have considered Information Dispersal algorithm (IDA) in order to transmit real-time data through multiple paths.

The remainder of this chapter is organized as follows. Section 2 covers theoretical background mainly focused on issues in designing wireless ad hoc network and the existing works related to the proposed framework while Section 3 describes a framework for secure multipath multimedia streaming over wireless ad hoc network. Section 4 describes key distribution mechanism and Section 5 presents secure multiple route discovery scheme as well as real-time data forwarding scheme. Section 6 gives security analysis, whereas Section 7 exemplifies a performance evaluation of the proposed framework. Finally, Section 8 concludes the chapter.

2. Theoretical Background Theory

This section covers not only issues and challenges in designing wireless ad hoc network, but also some of the existing approaches for multipath routing in wireless ad hoc network, security framework for multipath wireless ad hoc network and multimedia transmission over wireless ad hoc network.

2.1 Issues in designing Wireless ad hoc network

While designing a wireless ad hoc network, we need to consider the following issues (Mohapatra & Krishnamurthy, 2004).

- Dynamic topology: The topology in a wireless ad hoc network may change randomly due to nodes' mobility. As nodes move in and out of the range of each other, some links break and new links are created.
- Multi-hop paths: Nodes inside an ad hoc network are often not within direct communication range. Thus the support of multi-hop paths is essential to the design of an ad hoc network. Also because of multi-hop paths, the end-to-end packet drop due to channel error is increased and the end-to-end throughput is greatly decreased, compared to the single-hop infrastructure based wireless networks.
- Unreliable wireless medium: The wireless communication medium has variable and unpredictable characteristics. Due to varying environmental conditions, such as different levels of electro-magnetic interference (EMI), the signal strength and propagation delay fluctuate with respect to time and environment.
- Self-organizing: The ad hoc network must autonomously determine its own configuration parameters including: addressing, routing, clustering, identification, power control, and etc.
- Energy conservation: Most ad hoc nodes, e.g. laptops, PDAs (Personal Digital Assistants) and sensors, have limited power supply, and cannot generate power themselves. Thus energy efficient protocols are critical for the longevity of the operation of the network.

- Scalability: Because of the extensive mobility and the lack of fixed infrastructure, pure ad hoc networks do not tolerate mobile IP or a fixed hierarchy structure. Thus, mobility, jointly with large scale is one of the most critical challenges in the design of ad hoc networks.
- Security: Because of the ability of the intruders to eavesdrop and jam/spoof the channel, the security problem of ad hoc networks is severe.

As a result of the above issues, a wireless ad hoc network is prone to numerous faults including:

- Transmission errors: Packets can be corrupted and dropped due to the unreliability of the wireless medium.
- Node failures: Nodes may fail at any time in the network. Nodes can also drop out of the network either voluntarily or when their energy is depleted.
- Link failures: Node failures and varying environmental conditions, e.g. increasing level of EMI, can cause links between two nodes broken. Both node and link failures can break a route, causing packet drop.
- Congestion: Depending on the topology of the network and the traffic flows, certain areas of the network can be congested, causing longer delays or packet loss.

Many different routing protocols have been proposed to solve the multi-hop routing problem in wireless ad hoc networks based on different assumptions and intuitions. Wireless ad hoc network routing must be simple and robust, and minimizes control message exchanges. On-demand routing protocols adapt well with dynamic topologies of the wireless ad hoc networks, due to their lower control overhead and quick response to route break. AODV (Perkin *et al*., 2003) and DSR (Johnson *et al*., 2001) are the most popular on-demand routing protocols in wireless ad hoc network. However, in a single-path routing, previously created multi-hop route could frequently break because of node mobility and interference. New route discovery would initiate for each failure, in turn, inducing routing overheads and latency. The topology of wireless ad hoc network is inherently volatile and routing algorithms must be robust against frequent topology changes caused by host movements.

A multipath routing protocol is a promising technique to overcome problems of frequent topological changes and link instability as the use of multiple paths could diminish effect of possible node and link failures.

2.2 Multipath routing in wireless ad hoc network

We briefly present the related works to multipath routing. DSR based multipath protocols (Leung *et al*., 2001; Lee & Gerla, 2001; Wang *et al*., 2001) as well as AODV based multipath protocols (Marina & Das, 2006; Ye et al., 2004; Lee & Gerla, 2000) have been proposed for wireless ad hoc networks.

Some of well-known multipath source routing protocols are Split Multipath Routing (SMR) (Lee & Gerla, 2001), and Multipath source routing (MSR) (Wang *et al*., 2001).

Split Multipath Routing (SMR) (Lee & Gerla, 2001) is one of the multipath extensions to DSR protocol. SMR is similar to DSR, and is used to construct maximally disjoint paths. It uses a modified route request (RREQ) packets flooding scheme in the process of route query. Duplicate RREQs are not necessarily discarded. Instead, intermediate nodes forward RREQs that are received through a different incoming link, and whose hop counts are not larger

than the previously received RREQs. By doing this, SMR increases the probability of two disjoint paths to the destination. Unlike DSR, intermediate nodes do not keep a route cache, and do not reply to RREQs. This is to allow the destination to receive all the routes so that it can select the maximally disjoint paths. Maximally disjoint paths have as few links or nodes in common as possible. The destination node returns the shortest path and another path that is maximally disjoint with the shortest path to the source node.
Multipath source routing (MSR) (Wang *et al*., 2001; Wang *et al*., 2002), is an extension of the on-demand DSR protocol. It consists of a scheme to distribute traffic among multiple routes in a network. MSR uses the same route discovery process as DSR with the exception that multiple paths can be returned, instead of only one. As in DSR, a source node will initiate a route discovery by flooding a RREQ packet throughout the network. Once the RREQ reaches the destination, a RREP will reverse the route in the route record of the RREQ and traverse back through this route. Each route is given a unique index and stored in the cache, so it is easy to pick multiple paths from there. Independence between paths is very important in multipath routing, therefore disjoint paths are preferred in MSR. Since source routing is used in MSR, intermediate nodes do nothing but forward the packet according to the route in the packet-header. The routes are all calculated at the source. A multiple-path table is used for the information of each different route to a destination. The traffic to a destination is distributed among multiple routes; the weight of a route simply represents the number of packets sent consecutively on that path.
However both these schemes do not include any security mechanisms.

2.3 Security mechanism in multipath wireless ad hoc network

Most of the routing protocols designed for wireless ad hoc networks generally assume all nodes in the network are cooperative and well behaving. But this assumption does not hold in many scenarios, in which routing information is vulnerable and misbehaving nodes could easily change the routing information to disrupt the network. Therefore, a number of secure routing protocols (Hu *et al.,* 2005; Sanzgiri *et al.,* 2005; Papadimitratos & Haas, 2002) have been proposed to prevent a set of attacks that attempt to compromise the route discovery. These protocols could be used to guarantee the acquisition of correct network topological information.
For securing multipath routing in the wireless ad hoc network, several protocols such as Secure Routing Protocol (SRP) (Papadimitratos & Haas, 2002), Secure Multipath Routing Protocol (SecMR) (Mavropodi *et al*., 2007) and Secure, Disjoint, Multipath Source Routing Protocol (SDMSR) (Berton *et al*., 2006) can be used.
Secure Routing Protocol (SRP) (Papadimitratos & Haas, 2002) assumes the shared symmetric key between the source and destination and data protection using Message Authentication Codes (MACs) in order to validate the authenticity and integrity. SRP is quite simple and lightweight solution, however, intermediary nodes are not authenticated, which leaves room for a lot of potential attacks. In fact one of the main security issues in SRP is that it has no defence against the invisible node attack (INA) (Marshall *et al*., 2003) that simply puts itself (and possibly a large number of other invisible nodes) somewhere along the message path without adding itself to the path, thereby causing potentially big problems as far as routing goes.
Mavropodi et al proposed an on-demand multipath routing protocol called Secure Multipath Routing (SecMR) (Mavropodi *et al*., 2007) protocol that can find multiple node

disjoint routes with protection against denial-of-service (DoS) attacks from a bounded number of collaborating insider attackers. However, the authors have mentioned that SecMR protocol does not fully protect from Man-In-Middle (MIM) and invisible node attacks.
Furthermore, these schemes do not consider secure real-time data transmission.
Earlier, Zhou and Haas (Zhou & Haas, 1999) proposed an approach using multiple routes between nodes to defend routing against denial-of-service (DoS) attacks. Then several protocols (Papadimitratos & Haas, 2006; Lou *et al*., 2009) using multiple paths between source and destination to provide secure data transmission in wireless ad hoc networks have been studied.
Secure Message Transmission Protocol (SMT) (Papadimitratos & Haas, 2006) proposed by Papadimitratos and Haas requires a security association between the source and the destination. SMT can operate with any underlying secure routing protocol. It uses Active Path Set (APS), a set of diverse, node disjoint paths to transfer dispersed pieces of each outgoing message using Information Dispersal Algorithm (IDA). The message and redundancy data are divided into a number of pieces so that if M out of N transmitted pieces are received successfully; the original message can be correctly reconstructed. The sender updates the rating of each path in its APS based on the feedback provided by the destination. The destination validates the incoming pieces and acknowledges the successfully received ones through a feedback across multiple routes back to the source. It can be seen that SMT provides limited protection against the use of compromised topological information, although its main focus is to safeguard the data forwarding operation. The use of multiple routes compensates for the use of partially incorrect routing information, rendering a compromised route equivalent to a route failure. Nevertheless, the disruption of the route discovery can still be the most effective way for adversaries to consistently compromise the communication of one or more pairs of nodes. Furthermore, SMT has not accounted re-sequencing mechanism.
Lou *et al*. presented a scheme called Security protocol for reliable data delivery (SPREAD) (Lou *et al*., 2009), which provides further protection to the existed data confidentiality service in an ad hoc network using multipath routing. It aims to protect secret message from being compromised. A secret message is transformed into multiple shares using the threshold secret sharing algorithm, are delivered via multiple node-disjoint paths to the destination. As the shares are delivered through multiple node-disjoint paths, the secret message as a whole is not compromised even if a small number of shares are compromised. However, as it mandates all the paths to deliver at least one share, the natural parallel redundancy of the multiple paths is reduced to serial redundancy and therefore, a malicious node dropping all packets or a broken link may disrupt the protocol. Moreover, SPREAD is not suitable for multimedia streaming, as it is not meant for real-time data transfer.

2.4 Multimedia transmission over multipath wireless ad hoc network

With an increase in the bandwidth of wireless channels and computational power of mobile devices, multimedia transmission over wireless ad hoc network is getting appealing. Nonetheless, the performance of real-time multimedia communication suffers from the quality variations of the wireless links in wireless ad hoc network. Thus the quality of real-time multimedia streaming is degenerated. The noisy communication channel can cause bit errors in the data transmission. This requires either additional redundancy or

retransmissions and therefore reduces the bandwidth. Moreover, fluctuations in the received signal strength due to interference or other changes in the environment can cause link failures. High delays or packet loss rates for the transmission are the consequence. The end user then gets bad quality in his received transmission, or even interruptions. This makes the deployment of real-time applications a challenging task. To overcome these challenges in multimedia transmission over wireless ad hoc network, several solutions (Mao *et al*., 2003; Wei & Zakhor, 2004; Hsieh *et al.,* 2007) have been proposed.

The authors in (Mao *et al*., 2003) introduced multi-stream coding with MultiPath Transport (MPT) for video traffic over ad hoc network. In their approach, a video bit stream is divided into several sub-streams by the video encoder and then packets from different substreams are sent over different paths.

The general architecture for multipath transport of video streams (Mao *et al*., 2003; Mao *et al*., 2005) is depicted in Fig. 1. At the sender, the raw video is compressed by a multi-stream encoder into *M* streams. Then the streams are partitioned and assigned to *K* paths by a traffic allocator. These paths are maintained by a multipath routing protocol. When the flows arrive at the receiver, they are first put into a re-sequencing buffer to restore the original order. Finally, the video data is extracted from the re-sequencing buffer to be decoded and displayed.

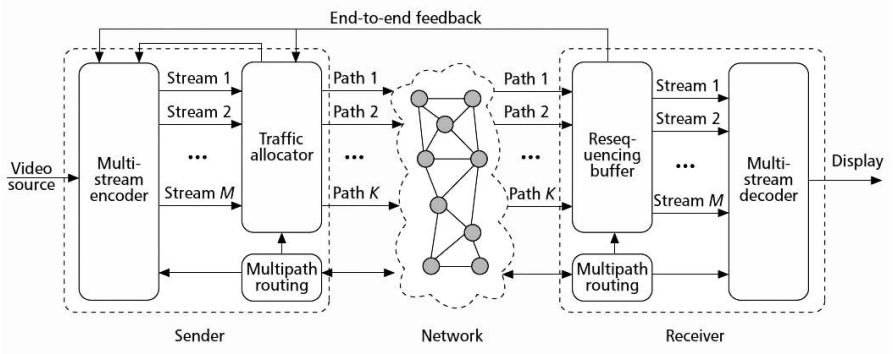


Fig. 1. General Architecture of multipath transport of video streams

Wei and Zakhor proposed Robust Multipath Source Routing Protocol (RMPSR) (Wei & Zakhor, 2004) that uses a per-packet allocation scheme to distribute video packets over two primary routes of two route sets, to support Multiple Description Coding (MDC) application over MANETs. And if one primary path is broken, it switches the transmission to another primary route.

Hsieh *et al.* (Hsieh *et al.,* 2007) presented an architecture supporting transmission of multiple of video streams in ad hoc networks by establishing multiple routing paths to provide additional video coding and transport schemes. In this framework, they have used on-demand multicast routing protocol to transport layered video streams.

However, these approaches do not consider any security measures.

3. Framework for Secure Multipath Multimedia Streaming over MANET

As the general architecture for the multipath transport of real-time multimedia applications depicted in the (Mao *et al.*, 2005) does not consider security measures, we have incorporated security enhancements in the framework for multipath multimedia streaming. The architecture for secure multipath multimedia streaming over wireless ad hoc network is shown in Fig. 2.

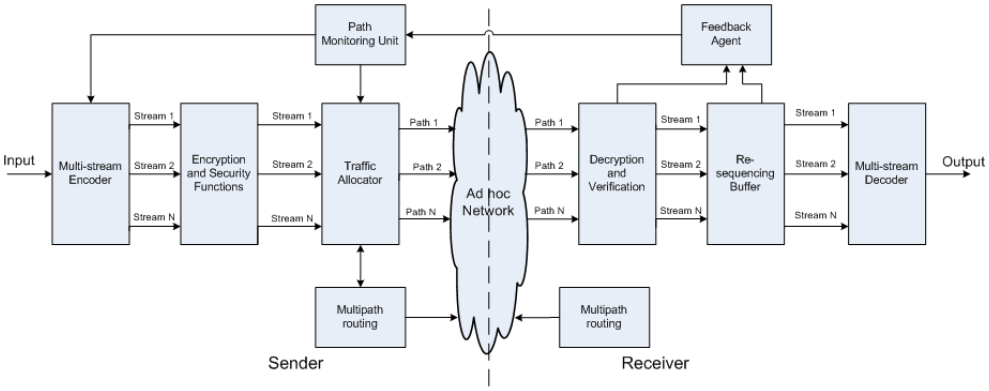


Fig. 2. Architecture of secure multipath multimedia streaming over wireless ad hoc network

We argue that proper selection of the active path set (APS) from the paths found by the multipath routing protocol can have a great impact on the usability of the found path-set in terms of both delivery ratio and delay and therefore will reduce not only the frequency of the costly route discovery process but also the overhead introduced to the network due to retry packets. Also, due to the dynamic topology of the network and existence of misbehaving nodes, which can change their behaviour through time, the paths will show time varying and non-stationary behaviour. Therefore, we maintain that any solution for improvement of the availability of end-to-end communication will not be successful unless it can adapt to the state of the paths and track their time varying behaviour.
For robust multiple routes discovery and data forwarding schemes in the presence of mobility and misbehaving nodes, the following assumptions have been considered:

- Source nodes and destination nodes are trustable, and between each pair of source node and destination node, there exists at least one route free of malicious nodes.
- Node-disjoint paths must be used to minimize the correlation between the paths.
- To control the overhead introduced to the network, an erasure coding or error-concealment source coding should be employed.
- It is assumed that the probability of the packets to be lost or modified is not the same for all paths.
- Due to variation of state of paths through time due to mobility or time varying misbehaviour of the malicious nodes; therefore, the system should be able to adapt to the current state of the paths.
- To guarantee the end-to-end confidentiality, end-to-end encryption must be used.

The function of each building block of the above-mentioned framework is presented as follows:

A. Multi-stream Coding Scheme
In the proposed framework, a multi-stream coding scheme is based on the Information Dispersal algorithm (IDA) (Rabin, 1989) to facilitate data redundancy. With this algorithm, the data and data redundancy is broken up into n streams, such that any m streams can be used to reconstruct data F, where n and m are positive integers, with m always less than or equal to n. If a sufficient number of streams are received at the destination, the destination proceeds to reconstruct the packets otherwise it waits.

B. Encryption & Security functions and Decryption & Verification
In the proposed framework, encryption & security functions at the sender side and decryption & verification at the receiver side are mainly for providing security for not only the route discovery but also for the real-time data transfer.

C. Traffic allocator
In this architecture, a traffic allocation scheme is used, which assigns the packets to the multiple paths using the weighted round-robin algorithm.
The list of active paths is maintained in the active path set (APS) table along with the path rating of each path. The path rating, r_s, is decreased by some constants each time failed transmission or unsuccessful transmission is reported, and it is increased by some constant for each successful reception. If the path rating falls below a threshold, the path is removed from the APS table.

D. Re-sequencing Buffer
One major concern when using multipath transport is the additional re-sequencing delay. Since packets sent on different paths suffer different delays, they may arrive at the receiver out-of-order. Thus the receiver needs to use a re-sequencing buffer to temporarily store the received fragments and put them in order using the sequence numbers.

E. Feedback agent
The main function of this agent is to collect the information of the received data and send a feedback message to the sender. The information of the received data can be either successful received, or unsuccessfully received due to modification or out-of-order, or never received because of packet loss due to congestion, wireless channel error, link breakage and/or misbehaving nodes.

F. Path Monitoring Unit
Monitoring scheme can adapt to the changes in the state of the paths. Depending upon the changes in the path, it would change the path cost which in turn change path rating. Here a path cost is the state of the path(s) describing the security and reliability related parameters of the path. For each path, we define two parameters reflecting availability and stability of the path. The parameters $a_i(t)$ and $s_i(t)$ denote the path availability and stability for all the paths respectively. These are stochastic representations of estimated reliability and stability of the i^{th} path among all available paths. Based on feedback message, the path cost is assigned with α^+, α^- and β^- for

successful delivery, packet loss, and modification respectively. It shows the path behavior in last packet transmission.

G. Multipath routing
The problem addressed by multipath routing protocols in the above-mentioned architecture is how to build multiple paths, in order to maximize the received video quality at the receive side. The key issue for the success of multipath streaming is to make packet loss over multiple paths as uncorrelated as possible. One natural metric for selecting multiple paths is to require them to be node-disjoint. Packet loss due to link failure or path breakage caused by nodes' movement are independent among node disjoint paths.

4. Key Distribution Mechanism

In this section, we describe key distribution mechanism used in the proposed framework that is based on self-certified key cryptography (Girault, 1991). In this regard, the self-certificate (Lee & Kim, 2000) for self-certified key is considered, which can provide the authenticity of self-certified key. Table 1 shows the notations used in key distribution mechanism.

Notation	Description
x_X	private key of node X
y_X	public key of node X
k_X	random number choosen by node X
CI_X	Certificate Information for node X
r_X	commitment of node X
ID_X	Identity of node X
CN	Certificate number
t_e	Certificate expiry time
s_X	signature parameter for node X
$h()$	hash function
$sCert_X$	Self-certificate generated by node X
$Sig_{x_X}(M)$	Message M digitally signed by node X

Table 1. Notations used in the key distribution mechanism

Initially, a centralized authority (CA) chooses random number x_{CA}, which is its private key and the corresponding public key is computed as $y_{CA} = g^{x_{CA}}$. It is assumed that y_{CA} is known to all the nodes presented in the network.
Prior to joining the network, the mobile node (MN) must connect to the trusted Certificate Authority (CA). This process is done offline. Suppose a node A joins the network, it will obtain the self-certified key from CA and then generate self-certificate, which is as follows.

1. CA chooses $\tilde{k}_A$.
2. CA computes $\tilde{r} = g^{\tilde{k}_A}$ and sends it to a node A.

3. The node A chooses *a* and computes $r_A = \tilde{r}_A g^a$
4. The node A sends ID_A, r_A to CA.
5. CA creates $CI_A = [ID_A \| r_A \| ID_{CA} \| y_{CA} \| CN \| t_e]$
6. CA computes $\tilde{s}_A = x_{CA} h(CI_A) + \tilde{k}_A$ and it to the node A.
7. The node A obtains its private key $x_A = \tilde{s}_A + a$
8. The node A verifies CA's signature by $y_A = g^{x_A} = y_{CA}^{h(CI_A)} r_A$
9. The node A then generates $sCert_A = Sig_{x_A}(CI_A, y_A)$

When node A presents $sCert_A$, any node in the network can explicitly verify the validity of y_A as follows.

1. Check the validity of node A's signature in $sCert_A = Sig_{x_A}(CI_A, y_A)$ by y_A.
2. Check the validity of CA's certification by checking $y_A = g^{x_A} = y_{CA}^{h(CI_A)} r_A$

5. Secure Multipath Routing Scheme

In this section, we propose a secure framework for multipath wireless mobile ad hoc networks that provides end-to-end security between the source-destination pair. The main goal of this framework is to provide security not only on the multipath routing protocol between the source and destination nodes but also on data transmission using these multiple routes.

This proposed framework is designed on based of source routing as such DSR. The proposed framework has three basic operations: route discovery, real-time data transmitting and route maintenance.

For provisioning security in route discovery phase, we have considered self-certified public key that can be used to generate self-certificate and digital signature, whereas used session keys to encrypt real-time data during real-time data transfer phase.

Before we discuss about the proposed framework further, we provide some of the assumption we have made for this scheme:

- We assume that a source node (S) and a destination node (D) share some secret information between them. For example, a source node knows a destination node's public key. For the key distribution, we have considered self-certified public keying technique (Girault, 1991; Lee & Kim, 2000), so any node in the network can compute another node's public key knowing public parameter, ID and CA's public key.
- We assume bidirectional communication on each link. This assumption is justified, since many wireless media access control (MAC-layer) protocols, including IEEE 802.11, require bidirectional communication.
- We assume that a mobile node can communicate with only neighbouring nodes and maintains the list of all its current immediate neighbouring nodes. However, secure neighbour discovery would only serve to strengthen the security of this scheme.

Table 2 shows the notations used in the proposed secure multipath route discovery for wireless ad hoc network.

Notation	Description
Sq	Unique ID assigned by S to RREQ
$Sign_{K_X-}(M)$	Message M digitally signed by node X
$sCert_X$	Self-certificate generated by node X
N_S	Nonce by S
SK_S, SK_D	Session keys generated by S & D

Table 2. Notations used in the multipath route discovery

5.1 Route Discovery

Route Discovery for multipath routing in wireless multihop ad hoc network is as follows: The route from source S to destination D will be obtained by flooding the network with route request (RREQ) packets.

When a node receives an RREQ packet with source address S and destination address D, it looks at its Intermediate node table. Intermediate node table maintains the list of recent most RREQ received for any source destination pair and the intermediate nodes for the request. If the packet arrived has a list of intermediate nodes that is a superset of what is there in the routing table, the packet is discarded else the node adds its own entry into the packet and rebroadcasts it.

Suppose an intermediate node 1 receives the RREQ directly from *S*. When the same RREQ packet with intermediate nodes {2} arrive from 2, *1* discards it. Upon receiving the RREQ, node *1* appends its address in route list and self-certificate, then rebroadcasts it.

In case of node *4*, it will accept RREQ from neighbors and 2, however discard that from node 5. Upon receiving the RREQ from node *1*, node *4* verfies the $sCert_1$, If it is valid, node *4* removes the signature of node *1* and signs the RREQ message with its K_{4-} and replaces $sCert_1$ with its $sCert_4$. And it appends its address in route list then rebroadcasts it.

Route request process is as follows:

$$S \Rightarrow *: \quad \left\langle Sign_{K_{S-}}(REQ,S,D,Sq),\ route\_list, E_{K_{D+}}(N_S,SK_S), sCert_S \right\rangle$$

$$1 \Rightarrow *: \quad \left\langle Sign_{K_{1-}}\left(Sign_{K_{S-}}(REQ,S,D,Sq),\ route\_list, E_{K_{D+}}(N_S,SK_S), sCert_S\right) sCert_1 \right\rangle$$

$$4 \Rightarrow *: \quad \left\langle Sign_{K_{4-}}\left(Sign_{K_{S-}}(REQ,S,D,Sq),\ route\_list, E_{K_{D+}}(N_S,SK_S), sCert_S\right) sCert_4 \right\rangle$$

$$7 \Rightarrow *: \quad \left\langle Sign_{K_{7-}}\left(Sign_{K_{S-}}(REQ,S,D,Sq),\ route\_list, E_{K_{D+}}(N_S,SK_S), sCert_S\right) sCert_7 \right\rangle$$

Route request process in Route Discovery is shown in Fig. 3.

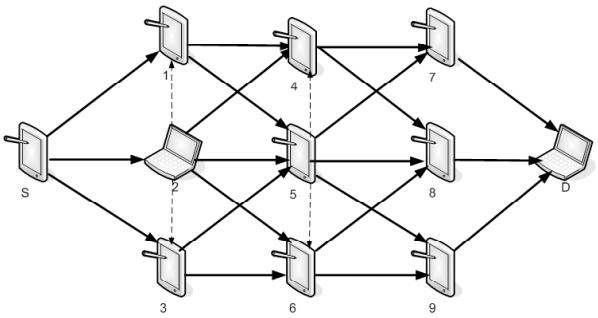


Fig. 3. Route Request Processing

When D receives RREQ packets from its neighbouring nodes, it is responsible for discovering multiple paths - primary path and node disjoint paths from all the received routes.
When receiving the first RREQ, the destination verifies all the signatures and caches the route list. It decrypts and stores session key from S.
Then D generates route reply (RREP) packet, which includes accumulated route as obtained from RREQ, a digital signature of the D on the entire message, and encrypted session key SK_D. The RREP is then sent back on the reverse route as given by the accumulated route in the RREQ. Each intermediate node on the reverse route verifies that its identifier as well as the predecessor and successor nodes' identifiers in the accumulated route.
If both tests are valid, the intermediate node signs the RREP and passes it to the next node in the path. As a result, the RREP reaches the source node. This node verifies whether it received the message from its neighbour and if this neighbour is the first node on the path. The path is then accepted to be valid if all the signatures are verified. It also decrypts and stores the session key from destination.
Route reply process is as follows:

$D \Rightarrow 7: \quad \langle Sign_{K_{D^-}}(REP, S, D, Sq, N_S, route\ list), E_{K_{S^+}}(S_{K_D}), sCert_D \rangle$

$7 \Rightarrow 4: \quad \langle Sign_{K_{7^-}}(Sign_{K_{D^-}}(REP, S, D, Sq, N_S, route\ list), E_{K_{S^+}}(S_{K_D}), sCert_D), sCert_7 \rangle$

$4 \Rightarrow 1: \quad \langle Sign_{K_{4^-}}(Sign_{K_{D^-}}(REP, S, D, Sq, N_S, route\ list), E_{K_{S^+}}(S_{K_D}), sCert_D), sCert_4 \rangle$

$1 \Rightarrow S: \quad \langle Sign_{K_{1^-}}(Sign_{K_{D^-}}(REP, S, D, Sq, N_S, route\ list), E_{K_{S^+}}(S_{K_D}), sCert_D), sCert_1 \rangle$

When the destination receives a duplicate RREQ, it will compare route path of RREQ to its route cache. If only source and destination nodes are same, a path is a node-disjoint path; otherwise it will discard the RREQ.

5.2 Route Maintenance

Whenever a route breaks because of node mobility, the neighbor of the node will send a route error to the source. The source will then discard that route from the routing table. If the source has another path to the destination, it can use it. When the source has no entry for the destination and the session is still active, it would initiate a new route discovery.
In order to authenticate the packet and ensure freshness, this scheme uses digital signature along with a nonce in route error messages.

5.3 Real-time Data Transmitting

Protecting the routing message from attacks is only one part of the security mechanisms of an ad hoc network. Often, a malicious node may behave normally during route discovery phase, however, during data forwarding phase, it either drops the segment packet or modifies the content of the packet and then forwards it.

The proposed scheme provides secure multipath data forwarding. In data forwarding, session keys SK_S and SK_D are used to encrypt and hash packets transmitted respectively. Apart from doing encryption and hashing, the packets would be divided in n fragments that would be sent to the destination on n different routes. The notations used in secure real-time data transfer are shown in Table 3.

Notation	Description
N	Sequence number
$h()$	hash function
TS	Timestamp
$E_{K_{X+}}(M)$	Encryption of message M with K_{X+}
$D_{K_{X-}}(M)$	Decryption of message M with K_{X-}
SK_S, SK_D	Session keys generated by S & D

Table 3. Notations used in secure real-time data transfer

To send multimedia data M from the source S to destination D,

1. S divides the packet in a set of n streams $\{g_0, g_1, ..., g_{n-1}\}$ with redundancy factor r. Thus the resultant length of each stream would be
$$len(g_i) = r \times \frac{len(M)}{n}; \quad 0 \le i < n$$
2. It encrypts each stream with SK_S i.e. it calculates encrypted stream as
$$E_{g_i} = E_{SK_S}(g_i \,||\, i \,||\, n \,||\, N \,||\, TS); \quad 0 \le i < n$$
3. It computes $h\left(E_{g_i} \,||\, SK_D\right)$
4. It sends $E_{g_i} \,||\, h\left(E_{g_i} \,||\, SK_D\right)$ to D on path i.
5. If S receives an acknowledgement from D with some successful delivered, some unsuccessfully delivered and some lost fragments, it resends the unsuccessful fragment and lost fragment on another path.

When D receives any packet from S from path i.

1. D generates $h\left(E_{g_i} \,||\, SK_D\right)$
2. It compares generated hash with the received one. If they match, the packet must be from S and not from any other node. If not, it informs the feedback agent.
3. It decrypts packet i.e. it calculates $g_i \,||\, i \,||\, n \,||\, N \,||\, TS = D_{SK_S}\left(E_{g_i}\right)$.
4. It checks timestamp.

5. As soon as D receives sufficient packets with sequence number *N* and different *i*'s, it reconstructs *M*.
6. If D doesn't receive enough packets to reconstruct *M* till the receiver timer expires, it sends the acknowledgement of the received fragments to S.

6. Security Analysis

A lot of attacks are possible in wireless mobile ad hoc networks that could threaten the security of the network (Pervaiz et al., 2007). We will evaluate the proposed scheme for passive attacks and active attacks.

A. Active attacks:

In active attacks, an attacker actively participates in disrupting the normal operation of the network services. A malicious host can create an active attack by modifying packets or by introducing false information in the ad hoc network. It confuses routing procedures and degrades network performance. Thus the attacks can carry the different detriments that focus on impersonation, modification, fabrication, replay, denial of service, and disclosure attacks (Mishra, 2008).

In this scheme, every intermediate node uses digital signature for authentication; source and destination nodes also use their certificates and session keys to authenticate and communicate securely. Therefore, this scheme is resilient to several active attacks such as attacks due to impersonation, modification and fabrication. For instance, a malicious node may cause fabrication attacks by falsifying route error (RERR) messages. This scheme defeats the RERR fabrication attack as it uses digital signature for protecting RERR messages and ensures authenticity and integrity of RERR messages. Similarly, a malicious node may try to impersonate a source node, however, this attempt will not be successful as the source node uses its digital signature to secure the non-mutable parts in RREQ message. Furthermore, since the scheme uses explicit self-certified public keys and self-certifications, malicious intermediate nodes would not be able to modify packets during the route discovery process. And this scheme uses nonce and timestamp to prevent from replay attack.

B. Passive attacks:

In passive attacks, an intruder snoops the data exchanged without altering it. The attacker mainly eavesdrop the data packets in the network without doing any active operations, or just refuse to execute the requested function. The goal of the attacker is to obtain information that is being transmitted, thus violating the message confidentiality. Since the activity of the network is not disrupted, these attackers are difficult to detect.

In this scheme, source node and destination node use their public key and session key to encrypt the communication data between them. Since the important information is encrypted during route discovery process, passive attackers cannot access the message without the knowledge of corresponding keys. Moreover, the use of an information dispersal technique makes difficult for passive attackers to get whole information.

7. Performance Evaluation

In order to evaluate the performance of the proposed framework in wireless ad hoc network, we have designed experimental model and simulated using OPNET Modeler (Opnet, 2008). We have modified DSR model to provide multipath routing protocols as per proposed framework.

7.1 Simulation Environment

In the simulation, the network coverage area is a 1000m x 1000m square with 50 mobile nodes, each having radio power range of 300m. The channel capacity is 2 Mbps. The IEEE 802.11 Distributed Coordination Function (DCF) is used as the MAC-layer protocol. The nodes are initially uniformly distributed throughout the network area and their movement is determined by the random waypoint mobility model. We have used a pause time of 1.0s for all the experiments. The speed of the nodes varies from 0m/s to 20m/s.
The traffic model of the audio streaming system considered employs G.729 codec for which the payload is 10 bytes and packet rate is 50 packets per second. Two simulatenous streamings can occur and the sources and destinations are chosen randomly with uniform probabilities. Each run executes 300 seconds of simulation time.

7.2 Simulation Scenarios and Metrics

For the simulation, two scenarios have been designed in wireless ad hoc environment - benign environment and adverse environment.
Under normal condition, we assume that there is no misbehaving nodes along the paths. And all the intermediate nodes are good behaving. Whereas, a second one is under adverse environment, there may be individual misbehaving nodes that can cause black hole attack. This attack is a selective data forwarding attack, in which adversaries only forward routing control packets, while dropping all data packets.
Several simulations were carried out with three schemes, namely, the proposed scheme, single-path DSR, and Mao's scheme with layered coding and selective Automatic repeat-request (ARQ) in both the environments.
While simulating audio streaming in wireless ad hoc network in above mentioned environments, we have considered following three key performance metrics to be evaluated:

- Packet Delivery Ratio (PDR): This is the ratio of the number of data packet successfully delivered to the destinations to the number of data packets generated by the constant bit rate (CBR) sources. It specifies the packet loss rate, which limits the maximum throughput of the network. The better the delivery ratio, the more complete and correct is the routing protocol.
- Average end-to-end delay: The end-to-end delay, network delay, indicates how long it took for a packet successfully delivery from the CBR source to the application layer of the destination. It represents the average data delay in the network.
- Normalized routing load: It is measured by the number of routing control packets transmitted per data packet delivered at the receiver. This is an important metric to compare the performance of different protocols since it can give a measure of the efficiency of protocols, especially in a low bandwidth and congested wireless environment.

7.3 Results and Analysis

In the simulation, we have examined the performance of the proposed secure scheme under normal and adverse environments.

Under normal condition, the single-path DSR, Mao's scheme (Mao et al., 2005) with layered coding and selective ARQ and the proposed scheme are compared. The speed of node is varied from 0 m/s to 20 m/s.

Fig. 4 shows the packet delivery rate plotted against node speed for the single path DSR, Mao's scheme and the proposed framework. It can be seen that the packet delivery rate of the proposed scheme is less than that of the single path DSR and Mao's scheme. Whereas Mao's scheme has the highest PDR for all the node speed.

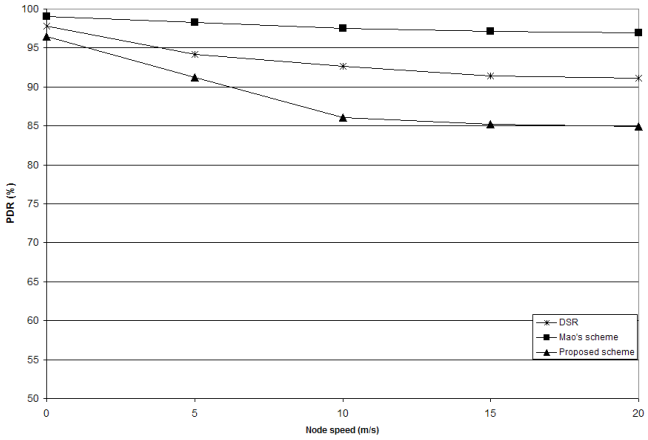


Fig. 4. Packet delivery ratio in normal environment

Fig. 5 shows the average end-to-end delay plotted against node speed for the single-path DSR, Mao's scheme and the proposed framework under normal condition. Due to cryptographic functions, the average delay of the proposed scheme is higher than that of the single path DSR and Mao's scheme. And it can be seen that Mao's scheme has the lowest average delay.

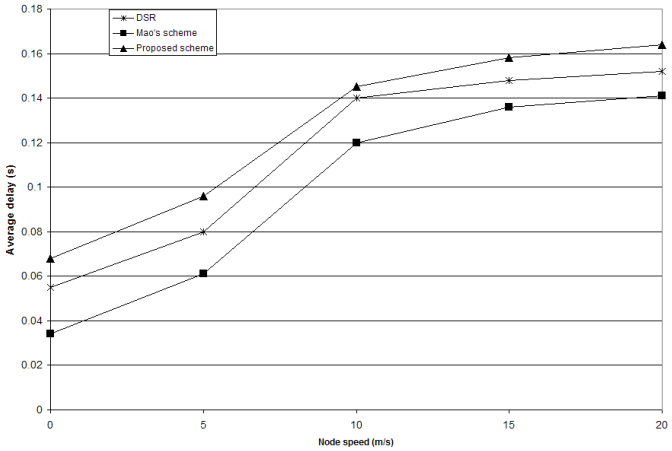


Fig. 5. Average end-to-end delay in normal environment

Fig. 6 shows the normalized routing load characteristics for single-path DSR, for Mao's scheme and proposed scheme. It can be seen that the normalized routing loads in Mao's scheme and the proposed scheme are much lower than that of DSR. With the increase of node speed, the normalized routing load in DSR increases more quickly than those in Mao's scheme and the proposed scheme. For two later schemes, it increases slowly with the increase of node speed. Since Mao's scheme and the proposed scheme can find multiple alternate route paths in a route discovery process, the protocols tremendously decrease the number of route rediscovery process. Whereas, since DSR encounters more link failures with the increase in mobility, it has to trigger more new route discovery process which causes more routing control packets.

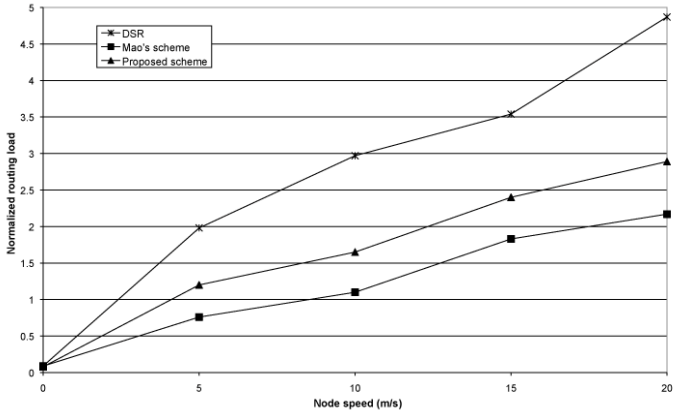


Fig. 6. Normalized routing load in normal environment

Under adverse environment, the proposed secure scheme is compared with the single-path DSR and Mao's scheme. The speed of node is at 10 m/s. And the percentage of misbehaving nodes in the network is varied from 0 to 20%.
Fig. 7 shows packet delivery rate plotted against number of misbehaving nodes for single-path DSR, Mao's scheme and the proposed secure framework. It can be seen that the packet delivery rate in network suffering different percentage of misbehaving nodes.
With the increase of misbehaving nodes in network, the packet delivery rate for the single DSR and Mao's scheme decrease dramatically. In case of the proposed secure scheme it is affected in a much lesser extent.

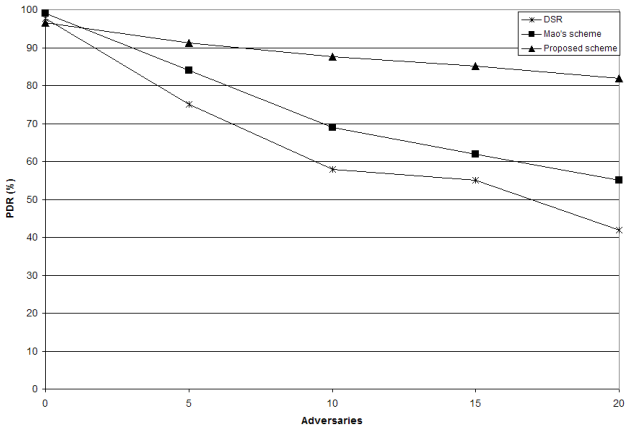


Fig. 7. Packet delivery ratio (adverse environment)

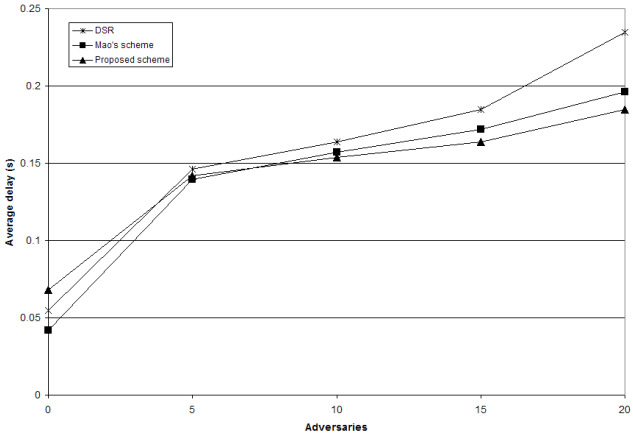


Fig. 8. Average end-to-end delay (adverse environment)

Fig. 8 shows average delay plotted against number of misbehaving nodes for single-path DSR, Mao's scheme and the proposed secure framework. It can be seen that the average delay in network increase with the increase in percentage of misbehaving nodes. At higher

percentage of malicious nodes, the average delay for the proposed scheme is lesser than single-path DSR, and Mao's scheme. With the increase of misbehaving nodes in network, the average delay for the single-path DSR increase significantly while that for Mao's scheme and proposed secure scheme increase steady.
Fig. 9 shows normalized routing load plotted against number of misbehaving nodes for single-path DSR, Mao's scheme and the proposed secure framework. It can be seen that the normalized routing loads for the single-path DSR, and Mao's scheme increase dramatically with increase of misbehaving nodes in network. Whereas, the normalized routing load for the proposed scheme is increased gradually.

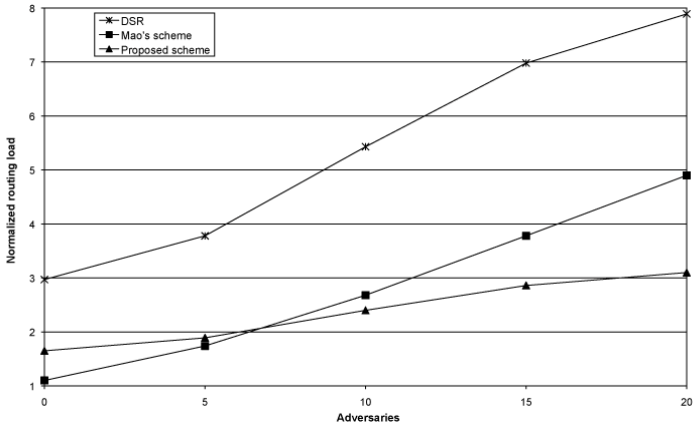


Fig. 9. Normalized routing load (adverse environment)

8. Conclusions

This chapter presents a secure and reliable framework for multipath audio streaming over wireless multihop network. It also shows the need of secure multipath routing protocol for multimedia streaming over wireless ad hoc network. While designing multipath routing scheme for a wireless ad hoc network, we have considered not only self-certified public keying technique and self-certificate but also digital signature and encryption technique for securing ad hoc routing as well as real-time data transfer. And we have considered Information Dispersal algorithm (IDA) in order to transmit real-time data through multiple paths. We have conducted security analysis for the proposed scheme and shown its robustness to various attacks. On implementing the proposed scheme in OPNET simulation, we have obtained various simulation results, which show that the performance of the proposed framework is better than the existing schemes under the adverse environment.
We can implement this scheme for video communication over multipath ad hoc network. Future work should consider on not only multipath but also multicast video streaming since multicast is an appealing technique for many applications, such as group video conferencing and video-on demand (VOD) services, and results in bandwidth savings as compared to multiple unicast sessions. We can also consider multiple video sources providing simultaneously service for multiple receivers.

9. Acknowledgements

This research was supported by the Plant Technology Advancement Program (07SeaHeroB01-03) funded by Ministry of Construction & Transportation, and the WCU Program (R31-2008-000-10026-0) by the Ministry of Education, Science, and Technology of Korean Government.

10. References

Berton, S.; Yin, H., Lin, C., & Min, G. (2006). Secure, Disjoint, Multipath Source Routing Protocol (SDMSR) for Mobile Ad-Hoc Networks, *Proceedings of Fifth International Conference on Grid and Cooperative Computing (GCC '06)*, pp 387--394, Oct. 2006

Girault, M. (1991). Self-certified public keys, *Proceedings of Advances in Cryptology (Eurocrypt'91)*, Springer, pp. 490-497, 1991

Hsieh, M. Y.; Huang, Y. M., & Chiang, T. C. (2007) Transmission of layered video streaming via multi-path on ad hoc networks, *Springer Multimedia Tools Applications*, Vol. 34, No. 2, 2007, pp. 155-177

Hu, Y. C.; Perrig, A., & Johnson, D. B. (2005) Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks, *Springer Wireless Network*, Vol. 11, No. 1-2, 2005, pp. 21-38

Johnson, D. B.; Maltz, D. A. & Broch, J. (2001). DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, In: *Ad Hoc Networking*, C. E. Perkins (Ed.), Chapter 5, pp. 139-172, Addison-Wesley

Lee, B. & Kim, K. (2000). Self-certificate: PKI using self-certified key, *Proceedings of Conference on Information Security and Cryptology (CISC 2000)*, Vol. 10, No. 1 pp 65-73, 2000.

Lee, S. J. & Gerla, M. (2000). AODV-BR: Backup Routing in Ad hoc Networks, *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2000)*, Vol. 3, pp. 1311–1316, September 2000, IEEE

Lee, S. J. & Gerla, M. (2001). Split multipath routing with maximally disjoint paths in ad hoc networks, *Proceedings of IEEE International Conference on Communications (ICC'01)*, Vol. 3, pp. 867-871, Helsinki, Finland, 2001, IEEE.

Leung, R.; Liu, J., Poon, E., Chan, A., & Li, B. (2001). MP-DSR: A QoS-Aware Multi-Path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks, *Proceedings of 26th IEEE Annual Conference on Local Computer Networks (LCN 2001)*, pp. 132-141, November, 2001

Lou, W.; Liu, W., Zhang, Y., & Fan, Y. (2009). SPREAD: Improving network security by multipath routing in mobile ad hoc networks. *Springer Wireless Networks*, Vol. 15, No. 3, 2009, pp. 279-294

Mao, S.; Lin, S., Panwar, S. S., Wang, Y., & Celebi, E. (2003). Video transport over ad hoc networks: multistream coding with multipath transport. *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 10, 2003, pp. 1721-1737

Mao, S.; Lin, S., Wang, Y., Panwar, S.S., & Li, Y. (2005). Multipath video transport over ad hoc networks. *IEEE Wireless Communications*, Vol. 12 No. 4, pp. 42-49, Aug. 2005

Marina, M. K. & Das, S. R. (2006). Ad hoc on-demand multipath distance vector routing. *Wiley Wireless Communications and Mobile Computing*, Vol. 6, No. 7, pp. 969–988, 2006

Marshall, J.; Thakur, V., & Yasinsac, A. (2003). Identifying Flaws in the Secure Routing Protocol, *Proceedings of 22nd International Performance, Computing, and Communications Conference (IPCCC 2003)*, Phoenix, Arizona, USA, April 9-11, 2003, pp. 167-174.

Mavropodi, R.; Kotzanikolaoua, P., & Douligerisa, C. (2007). SecMR- a secure multipath routing protocol for ad hoc networks, *Elsevier Ad Hoc Networks*, Vol. 5, Issue 1, January 2007, pp 87-99

Mishra, A. *Security and Quality of Service in Ad hoc Wireless Networks*, Cambridge University Press, 2008

Mohapatra, P. & Krishnamurthy, S. (2004). *Ad Hoc Networks: technologies and protocols*. Springer Science & Business Media, Inc., 2004

Mueller, S.; Tsang, R. P. & Ghosal, D. (2004). Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges, In: *Performance Tools and Applications to Networked Systems*, LNCS 2965, pp. 209-234

OPNET Modeler, URL: http://www.opnet.com. Accessed on July 2008

Papadimitratos, P. & Haas, Z. J. (2002). Secure Routing for Mobile Ad hoc Networks, *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, San Antonio, TX, USA, 2002

Papadimitratos, P. & Haas, Z. J. (2006). Secure Data Communication in Mobile Ad hoc Networks. *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 2, 2006, pp. 343-356

Perkins, C. E.; Belding-Royer, E., & Das, S.R. (2003). Ad hoc on-demand distance vector (AODV) routing. *IETF RFC 3561,* July 2003

Pervaiz, M. O.; Cardei, M., & Wu, J. (2007). Routing Security in Ad Hoc Wireless Networks, *Network Security*, S. Huang, D. MacCallum, & D. Z. Du (Eds.), 2007 Springer

Rabin, M. O. (1989). Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM*. Vol. 36, No. 2, pp 335-348, 1989

Sanzgiri, K.; LaFlamme, D., Dahill, B., Levine. B.N., Shields, C., & Belding-Royer, E.M. (2005) Authenticated Routing for Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 3, 2005, 598–610

Wang, L.; Shu, Y., Dong, M., Zhang, L., & Yang, O. W. W. (2001). Adaptive Multipath Source Routing in Ad Hoc Networks, *Proceedings of IEEE International Conference on Communications (ICC 2001)*, Vol. 3, pp. 867-871, Helsinki, Finland, June 2001

Wang, L.; Shu, Y., Zhao, Z., Zhang, L., & Yang, O. W. W. (2002). Load Balancing of Multipath Source Routing in Ad Hoc Networks, *Proceedings of IEEE International Conference on Communications (ICC 2002)*, Vol. 5, pp. 3197-3201, 2002

Wei, W. & Zakhor, A. (2004). Robust Multipath Source Routing Protocol (RMPSR) for Video Communication over Wireless Ad Hoc Networks, *Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2004)*, Vol. 2, pp. 1379-1382, Taiwan, June 2004

Ye, Z.; Krishnamurthy, S.V., & Tripathi, S.K. (2004). A routing framework for providing robustness to node failures in mobile ad hoc networks. *Elsevier Ad Hoc Networks Journal*, Vol. 2, No. 1, 2004, pp. 87–107

Zhou, L. & Haas, Z. J. (1999) Securing ad hoc networks, *IEEE Network*, Vol. 13, No. 6, pp. 24–30, 1999

16

Complete Video Quality Preserving Data Hiding for Multimedia Indexing

KokSheik Wong and Kiyoshi Tanaka
Shinshu University, Nagano
Japan

1. Introduction

Digital video has become a popular multimedia content in both online and offline environments thanks to the advancement of computer and portable devices, as well as broadband internet technologies. While digital video is still gaining popularity, it is important to consider ways to protect the contents from malicious use, efficient ways to search the desired contents in the database, secure ways to provide extra features for upgraded viewers, reliable ways to recover from transmission error for uninterrupted viewing, etc., for the future of digital video. To accomplish such tasks, data hiding is one of the fields that provides the solutions (Johnson et al., 2003; Katzenbeisser & Petitcolas, 2000).

In general, there are two types of data hiding for video: one that hides the video content itself (video encryption or scrambling) so that nobody understands what is being transmitted (Takayama et al., 2006; Wong et al., 2003; Zeng & Lei, 1999); the other that embeds external information into the video, hence utilizing video as the data host. We consider the latter in this chapter. Under this category, one of the basic requirements for a data hiding method is the ability to produce video of high image quality. On top of that, additional properties are desired, depending on the application in question. In case of watermarking, the information embedded into a video should be able to withstand some common image processing attacks such as re-compression at different bitrate, random video frame dropping, resizing, etc. (Cox et al., 2002). In case of steganography, the embedded information should stay undetectable with respect to steganalysis (Budhia et al., 2006), which is a process for revealing the existence of hidden information in a suspicious video. In applications such as annotation or indexing, even though it is not a compulsory property, it is usually preferable to achieve reversibility so that the embedded information could be removed to restore the original video. Other applications of data hiding could be found at (Katzenbeisser & Petitcolas, 2000; Kurosaki & Kiya, 2002; Yanagihara et al., 2005).

Some representative data hiding methods in video domain could be found at (Bodo et al., 2004; Kiya et al., 1999; Liu et al., 2004; Nakajima et al., 2005; Ni et al., 2006; Qiu et al., 2004; Sarkar et al., 2007; Xu et al., 2006; Zhang et al., 2001). For example, Kiya et al. embed information into an MPEG compressed video by modifying coefficients at selected location(s) in each 8×8 qDCTCs (quantized DCT coefficients) block (Kiya et al., 1999). The quantization table is further modified to suppress distortion. Nakajima et al. proposed a high carrier capacity data hiding method utilizing the idea of zerorun length coding in MPEG domain (Nakajima et al., 2005). After zigzag scanning, a dummy nonzero value is inserted at a location that is

z units away from the the original last nonzero qDCTC, where z depends on the data to be embedded. Zhang et al. utilize MV (motion vectors) in P and B-pictures as the data carriers for data embedding (Zhang et al., 2001). An MV is selected based on its magnitude, and its angle guides the modification operation. In these existing works, qDCTC or/and MV is/are usually utilized as the data carrier. Therefore, modifications done to the video during data embedding may alter the video bitrate. As a result, researches are carried out to maintain the original video bitrate for avoiding buffer overflow or underflow during video playback (Hartung & Girod, 1996; Pranata et al., 2004). For instance, Pranata et al. embed information into a frame by evaluating the combined bit lengths of a set of multiple watermarked VLC (variable length coding) codewords (Pranata et al., 2004). They successively replace the watermarked VLC codewords having the largest increase in bit length with their corresponding unmarked VLC codewords until a target bit length is achieved. Recently, a data hiding method using Mquant (i.e., the scaling factor in rate controller) as the data carrier is proposed, and this method always produces video with exactly the same bitrate as the original (compressed) video even after data embedding (Wong & Tanaka, 2007). Suboptimal histogram preserving modification scheme is also proposed to maintain the distribution of Mquant before and after data embedding.

In this work, we focus on complete image quality preservation, reversibility, and efficient data representation scheme as the fundamental research of data hiding in compressed video domain. Even though the aforementioned data hiding methods generally produce image/video of high quality regardless of their applications and data carrier in use, the image quality of the modified video is always lower than that of the original video. This is a critical drawback because these data hiding technologies cannot be utilized in applications where image quality degradation is not permitted. To solve this problem, we propose a novel data hiding method in the compressed video domain that completely preserves the image quality.

To the best of our knowledge, there is no data hiding method that completely preserves the image quality during data embedding, and this method is the first attempt of its kind. This method is also reversible, and it is applicable not only to the existing MPEG-1/2/4 or H.261/3 encoded videos but also applicable to the encoding process of MPEG-1/2/4 or H261/3 video from a sequence of raw pictures. The RZL (reverse zerorun length) data representation scheme is proposed to exploit the statistics of macroblocks for achieving high embedding efficiency while trading off with payload. We theoretically analyze that RZL outperforms matrix encoding (Crandall, 1998) in terms of payload and embedding efficiency for this particular data hiding method. The problem of video bitstream size increment as a result of data embedding is also addressed, and two independent solutions are presented to suppress this increment. Basic performance of this method is verified through experiments with various existing MPEG-1 encoded videos.

One of the possible applications of our data hiding method is video indexing where high image quality and reversibility are greatly desired because we can provide high quality video as well as additional specialized functions such as searching, playback control, hyper-linking with other media, etc. (Yanagihara et al., 2005). Here, we briefly introduce three practical scenarios using video indexing:

The first scenario is *educational use*. Instead of the traditional straight forward video (i.e., lecture) playback, we can provide interactive learning environment for the viewer. For example, the video could be played back at a specific speed suitable for the viewer, the parts of the video where important or complex ideas are presented could be easily accessed or repeated upon viewer's request, external resources such as presentation slides and related websites could

be hyper-linked to the frame of interest and accessed upon viewer's action, the area(s) in a frame where the viewer should pay attention to could be emphasized by marking or masking, frames in which the object of interest appear could be tagged for retrieval, etc. Note that the aforementioned features could be customized for each viewer based on his/her level of understanding on the material, and the embedded information could be removed when necessary. On top of that, the lecture to be viewed could be searched by using the information such as keyword that is embedded in it.
The second scenario is *servicing and maintainance business*. Instead of the traditional paper-based manual for an equipment, an interactive video manual could be authored using video indexing technology. Video based instruction along with audio description is extraordinarily informative especially when verbal or picture-based manual itself is incapable to provide detailed instructions. Suppose we author an interactive video manual for servicing computer printer. When a technician services the printer, he could watch and follow the interactive video instructions at his own pace. The technician could choose the area of the printer to be serviced (eg. printer cartridge, paper tray, firmware update, etc.) and the desired video will be played back. When there is a need for part replacement, the technician places the mouse cursor on the part to be replaced, and information such as URL to the order page, review and evaluation of the spare part manufactured by each company, current prices in the market, etc. could be displayed and accessed upon further action. This type of video could also be utilized for training new technicians.
The third scenario is *handling video database*. Instead of storing the classification information or unique tag of each video in a separate file, this information could be embedded into the video. By doing so, we could avoid managing two separate files (i.e., the record and the video itself) because the identifying information stays intact with the video and it could be decoded for purpose of indexing. Also, when two or more video databases are merged, ambiguity in video retrieval does not occur in our method. In case of using a pre-defined naming scheme for identifying the videos in place of data hiding technology, the ambiguity problem may occur because two videos, each from a different database, could have the exact same filename. From the ordinary user point of view, one could retrieve videos using query keywords, and browse these candidate videos having the same image qualities as the original unmodified videos. When the desired video is retrieved, the embedded indexing information could be removed (since our method is reversible) or it could be used for other purposes.

2. Review on video compression standard

2.1 MPEG-1 and MPEG-2 compression standards

In MPEG-1/2 compression standard, a sequence of pictures is segmented into GOP's (group of pictures). Pictures in each GOP are labeled as I, P or B, depending on the order in which they appear, and the interval between two consecutive P-pictures are determined by the M-factor parameter (Hanzo et al., 2007; Symes, 2004). Regardless of the picture type, each picture is divided into slices, and each slice is further divided into MBs (macroblocks). Each MB could be coded independently (INTRA mode), or motion compensated (INTER mode) where motion vectors and differential signals are selectively coded. Finally, each MB contains blocks of 8×8 qDCTCs for luminance and chrominance information. During video playback, DCT coefficients are reconstructed from the qDCTCs using Eq. (1). Here, x denotes the qDCTC, QT_1 and QT_2 are the default quantization table for INTRA and INTER-MB, respectively.

$$\mathrm{rec}[m][n] = \begin{cases} 2 \times x[m][n] & \times \quad \mathcal{Q}(k) \quad \times \quad \mathrm{QT}_1[m][n]/16, & \text{if INTRA} \\ (2 \times x[m][n] + \mathrm{sign}(x[m][n])) & \times \quad \mathcal{Q}(k) \quad \times \quad \mathrm{QT}_2[m][n]/16, & \text{otherwise} \end{cases} \tag{1}$$

For a specified video bitrate, MPEG utilizes Mquant (hereinafter referred as MQ) for distributing the available bits to code the pictures based on their spatial activities and similarity among index and reference frames. Each divisor in the default quantization table is scaled by MQ before the quantization operation is carried out. MQ assumes any integer in the range of $[1, 31]$, and its value is recorded in the header of each slice. This value is utilized by all MBs in the slice for encoding/decoding, but each MB can have its own MQ value. The acquirement of a new MQ value is indicated by the CODED_MQ flag in the header of each MB, and this newly coded MQ value is utilized by all tailing MBs until a new value is coded.

2.2 H.261, H263, and MPEG-4 compression standards

H.261, H.263, and MPEG-4 are similar to MPEG-1/2 in the sense that they are all cosine transform-based compression standards, and consist of many common entities such as MV, Mquant, qDCTCs, etc. On the other hand, these standards also differ in many aspects such as MV of size 8×8 pixels is only available in MPEG-4, three dimensional VLC in H.263 and MPEG-4, color schemes of 4:2:2 and 4:4:4 in MPEG-2, etc. Here, we only present the significant difference in these compression standards that is relevant to our data hiding method. In particular, during video playback, instead of using Eq. (1) to reconstruct the coefficients, Eq. (2) is utilized. The rest of the similarities and differences among MPEG-1/2/4 and H.261/3 could be found at (Hanzo et al., 2007; ITU-T H.261, 1990).

$$\mathrm{rec}[m][n] = \begin{cases} 0, & \text{if } x[m][n] = 0, \\ \mathrm{sign}(x[m][n]) \times (2\mathcal{Q}(k) \times |x[m][n]| + \mathcal{Q}(k)), & \text{if } x[m][n] \neq 0,\ \mathcal{Q}(k) \text{ is odd} \\ \mathrm{sign}(x[m][n]) \times (2\mathcal{Q}(k) \times |x[m][n]| + \mathcal{Q}(k) - 1), & \text{if } x[m][n] \neq 0,\ \mathcal{Q}(k) \text{ is even} \end{cases} \tag{2}$$

3. Methodology

We first present the basic idea using I-picture of MPEG-1 (i.e., all MBs are coded in INTRA mode, and no MB is skipped nor purely motion compensated), and extend our idea to handle INTER-MB that may occur in P and B-pictures. Modifications required to process MBs in MPEG-4 and H.261/3 are later justified.

3.1 INTRA-MB: *Excitement* and *promotion*

For any video decoder compliant to MPEG compression standard, the DCT coefficients (i.e., only AC components from INTRA-MB, and both DC and AC components from INTER-MB) are reconstructed by using Eq. (1), where $1 \leq m, n \leq 8$, respectively. However, DC components of INTRA-MB are reconstructed by the multiplication of a constant value (i.e., eight, and is irrelevant to the MQ value). To ease the discussion, let MB(j) denote the jth MB in a slice. Also, let $\mathcal{Q}(j)$ associate the CODED_MQ flag and the MQ value of MB(j) in the following manner:

$$\begin{aligned} \mathcal{Q}(j) = 0 \quad &\leftrightarrow \quad \text{CODED\_MQ = FALSE;} \\ \mathcal{Q}(j) > 0 \quad &\leftrightarrow \quad \text{CODED\_MQ = TRUE, where the coded MQ value is as specified.} \end{aligned} \tag{3}$$

Algorithm 1 *Exciting* an INTRA macroblock

1: $\mathcal{Q}(j_0) \leftarrow \alpha$
2: **for** Y, Cb, and Cr channel **do**
3: **for** all nonzero qDCTC$[m][n]$ **do**
4: qDCTC$[m][n] \leftarrow \beta \times$ qDCTC$[m][n]$
5: **end for**
6: **end for**

Algorithm 2 *Promoting* a macroblock

1: $\mathcal{Q}(j_0 + 1) \leftarrow \alpha \times \beta$
2: CODED_MQ flag $\leftarrow$ TRUE

Consider MB(j_0) such that $\mathcal{Q}(j_0) = \alpha \times \beta$ for some $\alpha, \beta \in \mathbb{N}$ and $\beta \neq 1$. Here, $\mathbb{N}$ denotes the set of natural numbers. Obviously, α and β are the factors of $\mathcal{Q}(j_0)$. If $\mathcal{Q}(j_0)$ is a prime, then the factors are unity and $\mathcal{Q}(j_0)$ itself. We *excite* MB(j_0) using **Algorithm 1**. Here, α is chosen as the new $\mathcal{Q}(j_0)$ value and β as the multiplying factor for all nonzero AC qDCTCs residing in MB(j_0). However, their roles could be interchanged as long as none of them is of value unity. Referring to the INTRA case of Eq. (1), the value of the reconstructed AC coefficient $rec[m][n]$ is exactly the same before (original) and after MB *excitement* (modified). In particular, the factor β that is "divided out" from the original $\mathcal{Q}(j_0)$ is compensated by the multiplication of β to all nonzero AC qDCTCs in MB(j_0) as performed in line 4 of **Algorithm 1**. Therefore, the image quality of MB(j_0) is completely preserved even after MB *excitement*.
As mentioned in **Section 2**, the last coded MQ value is utilized by the remaining MBs in the slice unless a new MQ value is coded. Since $\mathcal{Q}(j_0)$ is modified when MB(j_0) is *excited*, MB$(j_0 + 1)$ must be modified accordingly to achieve complete image quality preservation in the slice level. Note that modification is not required for MB$(j_0 + 1)$ if:

i. MB(j_0) is the last MB in the slice, i.e., $j_0 = N$, where N is the number of MBs in a slice, or
ii. $\mathcal{Q}(j_0 + 1) \neq 0$ in which case this value can never be $\alpha \times \beta$.

Otherwise MB$(j_0 + 1)$ is *promoted* using **Algorithm 2** to avoid the use of $\mathcal{Q}(j_0) = \alpha$ as the MQ value. Thus, the operation of MB *excitement*, followed by MB *promotion* when necessary, have no effect on the image quality of a slice of MBs, and hence a frame in a video. Note that, although the image quality of the video is completely preserved, it is achieved at the expense of an increase in the video bitstream size, which is further discussed in **Section 5**.

3.2 Complete video quality preserving data hiding

To embed information into a MPEG-1 compressed video utilizing MBs with MQ value of $\alpha \times \beta$, we propose ORS (ordinary representation scheme) which is defined as follows:

1. If the ith message bit $\mu(i)$ = "1", *excite* the MB, and *promote* the affected MB when necessary, or
2. If $\mu(i)$ = "0", do nothing to the MB.

The embedding flow is summarized in **Figure 1**. Instead of a fixed MQ value, we consider $\beta \in \S_1$ and the corresponding α's such that $\mathcal{Q} = \alpha \times \beta$ to increase the payload. Here,

$$\S_1 := \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31\}. \quad (4)$$

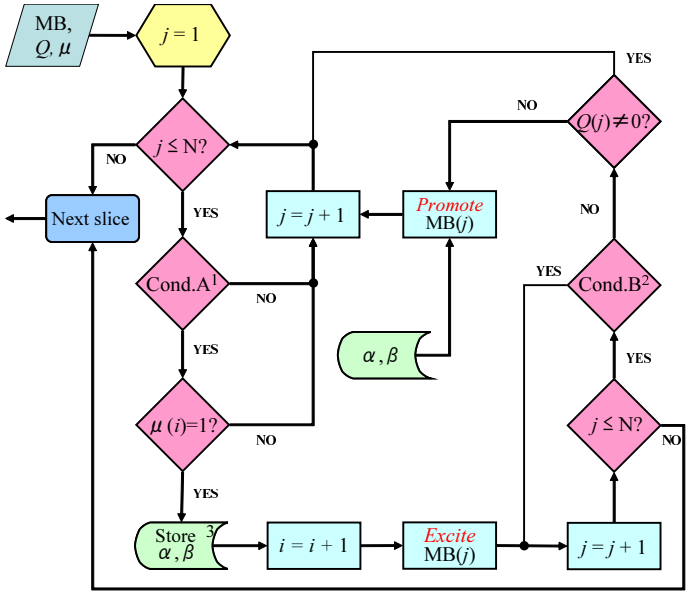


Fig. 1. Flowchart for data embedding. [1] **Cond.A**:= Is $\mathcal{Q}(j) > 0$ and $\text{mod}(\mathcal{Q}(j), \beta) = 0$ for at least one $\beta \in \S_1$?, [2]**Cond.B**:= Is MB(j) completely motion compensated without any qDCTC or skipped?, [3]This symbol denotes storage.

Thus, any MQ value in the range of $[2, 31]$ could be factored into α and β for at least one $\beta \in \S_1$. Note that each $\beta \in \S_1$ could also be utilized independently to realize multiple messages embedding (Wong et al., 2006). In particular, a maximum of 11 unique messages could be embedded into a video using INTRA-MBs in I-pictures while completely preserving the original image quality.

3.3 Data extraction and reversibility

To extract the embedded data, the modified video is parsed in the exact same order as MPEG decoder does. The flowchart for data extraction is summarized in **Figure 2**. When an MB with $\mathcal{Q} \neq 0$ is encountered, there are three interpretations:

(i) *excited* MB that holds the information bit "1", or

(ii) original MB that holds the information bit "0", or

(iii) original MB that holds nothing.

To resolve this ambiguity, we consider the condition

$$\text{mod}\ (x, \beta) = 0 \tag{5}$$

for each nonzero qDCTC $x \in$ MB. Case (i) occurs if condition (5) is true for a specific $\beta \in \S_1$ for all $x \in$ MB. We store the smallest value of $\beta \in \S_1$ that satisfies condition (5), update $\mathcal{Q} \leftarrow \mathcal{Q} \times \beta$, and yank an "1". On the other hand, case (ii) occurs if condition (5) fails but $\text{mod}\ (\mathcal{Q}, \beta) = 0$ for at least one $\beta \in \S_1$ (i.e., **Cond.D** holds true). In this case, "0" is yanked as

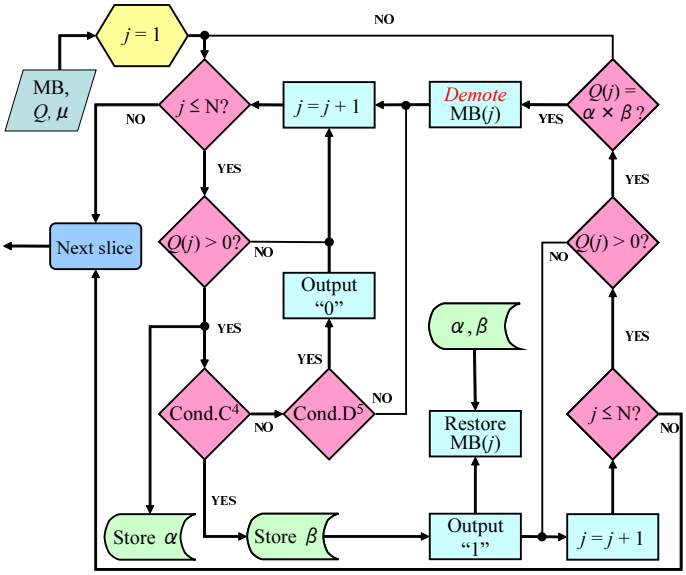


Fig. 2. Flowchart for data extraction and restoration. [4]**Cond.C**:= Is $\mod(x, \beta) = 0$ for all qDCTCs $x \in \text{MB}(j)$ for at least one $\beta \in \S_1$?, [5]**Cond.D**:= Is $\mod(\mathcal{Q}(j), \beta) = 0$ for at least one $\beta \in \S_1$?

the output. Case (iii) occurs if **Cond.D** fails, i.e., when the following holds true, and nothing is output:

$$\mathcal{Q} = 1 \tag{6}$$

Condition (5) or more generally, **Cond.C**, is valid for identifying *excited* MBs because it is very unlikely that all nonzero qDCTCs in an MB (from the original video) are divisible by a specific factor $\beta \in \S_1$, and this claim is verified through extensive experiments. Nevertheless, when such a case occurs, we can restrict data embedding to a set of MQ values. Suppose that there is at least one MB with $\mathcal{Q} = \alpha$ in the original video and all of its nonzero AC qDCTCs are divisible by β. We make sure that no other MQ values get mapped to α during data embedding, i.e., MB(j)'s with $\mathcal{Q}(j) = 2 \times \alpha, 3 \times \alpha, \cdots$ are ignored for data embedding. However, MB(j)'s with $\mathcal{Q}(j) = \alpha$ themselves could be utilized for data embedding since the value α will be replaced by $\hat{\alpha}$ ($\alpha = \hat{\alpha} \times \hat{\beta}$, and $\hat{\alpha}, \hat{\beta} \leq \alpha$) if they happened to be *excited* during data embedding. When MB(j) with $\mathcal{Q}(j) = \alpha$ is encountered during data extraction, we know that it must hold the information "0" since nothing gets mapped to it.

After encountering an *excited* MB, the next MB with a coded MQ value might be a *promoted* MB. If its MQ value is $\alpha \times \beta$, then it is either a *promoted* MB or an *excited* MB, depending on **Cond.C**. Otherwise, it is an MB that may hold "0", "1", or nothing, and hence condition **Cond.C** and **Cond.D** are considered again to determine its state. Note that a *promoted* MB never occurs by itself without a preceding *excited* MB. Therefore, we only resolve the ambiguity of reaching a *promoted* MB or an MB encoding "0" when its previous MB is *excited*. Suppose MB(j_0) is identified as holding "1", i.e., MB(j_0) is *excited* with $\mathcal{Q}(j_0) = \alpha$, and the information held by MB$(j_0 + 1)$ is to be determined. We list the interpretations of MB$(j_0 + 1)$ based on $\mathcal{Q}(j_0 + 1)$

$MB(j_0)$	$MB(j_0+1)$	Interpretation
α	$\neq \alpha \times \beta$	$MB(j_0)$ encodes "1", and $MB(j_0+1)$ encodes "0", "1" or nothing.
α	$\alpha \times \beta$	$MB(j_0)$ encodes "1", and $MB(j_0+1)$ is a *promoted* MB, or it encodes "1". It can never encode "0" because $\mathcal{Q}(j_0)$ and $\mathcal{Q}(j_0+1)$ can never be the same in the original video.
$\alpha \times \beta$	$\neq \alpha \times \beta$	$MB(j_0)$ encodes "0", and $MB(j_0+1)$ encodes "0", "1" or nothing.
$\alpha \times \beta$	$\alpha \times \beta$	$MB(j_0)$ encodes "0", and $MB(j_0+1)$ must encode "1" because $\mathcal{Q}(j_0)$ and $\mathcal{Q}(j_0+1)$ can never be the same in the original video.

Table 1. Distinguishing promoted MB and unmodified MB

Algorithm 3 Restoring an *excited* INTRA macroblock

1: $\mathcal{Q} \leftarrow \alpha \times \beta$
2: **for** Y, Cb, and Cr channel **do**
3: **for** all nonzero qDCTC$[m][n]$ **do**
4: qDCTC$[m][n] \leftarrow$ qDCTC$[m][n]$ / β
5: **end for**
6: **end for**

Algorithm 4 *Demoting* a *promoted* macroblock

1: $\mathcal{Q}(j_0+1) \leftarrow 0$
2: CODED_MQ flag $\leftarrow$ FALSE

in **Table 1**. For completeness of discussion, we also list the cases where $MB(j_0)$ encodes "0", i.e., $\mathcal{Q}(j_0) = \alpha \times \beta$.

With the procedures presented above, the *excited* and *promoted* MBs could be identified. Hence, we could restore an *excited* INTRA-MB to its original state by using **Algorithm 3**. Similarly, a *promoted* INTRA-MB could be restored to its original state by using **Algorithm 4**.

3.4 Example: Encoding and decoding information

An example is shown in **Figure 3** using two arrays of MBs (original at the top, modified at the bottom) together with their coded MQ values extracted from an I-picture. Here, the value of $\mathcal{Q}(j)$ is recorded in the array if $\mathcal{Q}(j) > 0$, or 'X' is marked otherwise. Suppose the message $\mu = 1010\cdots$. $MB(j_0)$ is *excited* to hold the first bit of μ (i.e., "1"), thus $\mathcal{Q}(j_0)$ is updated to $\alpha = 3$ and each nonzero AC qDCTC is scaled by the factor $\beta = 2$ as shown. Then, we check if $MB(j_0+1)$ needs to be *promoted*. Since $\mathcal{Q}(j_0+1) = 10 \neq 0$, $MB(j_0+1)$ is not *promoted*. Instead, $MB(j_0+1)$ is considered to encode the second bit of μ. $MB(j_0+1)$ is left as it is because an "0" is to be embedded. Note that $\mathcal{Q}(j_0+2) = 0$, but $MB(j_0+2)$ is not *promoted* since $MB(j_0+1)$ is not *excited*. Next, $MB(j_0+3)$ is *excited* to encode the third bit of μ (i.e., "1"). $MB(j_0+4)$ is *promoted* so that $\mathcal{Q}(j_0+4) = 13$. $MB(j_0+5)$ is left as it is to encode the forth bit of μ (i.e., "0"). The same process is repeated to embed the entire message.

To extract the embedded information, MBs are visited in the exact same order as in the encoding phase. DEC (message decoder) extracts "1" from $MB(j_0)$ because it satisfies **Cond.C** with $\beta = 2$. Since $\mathcal{Q}(j_0+1) > 0$ and $\mathcal{Q}(j_0+1) \neq \mathcal{Q}(j_0) \times 2$, DEC declares that $MB(j_0+1)$ is not a *promoted* MB. DEC extracts "0" from $MB(j_0+1)$ because it fails **Cond.C** but satisfies

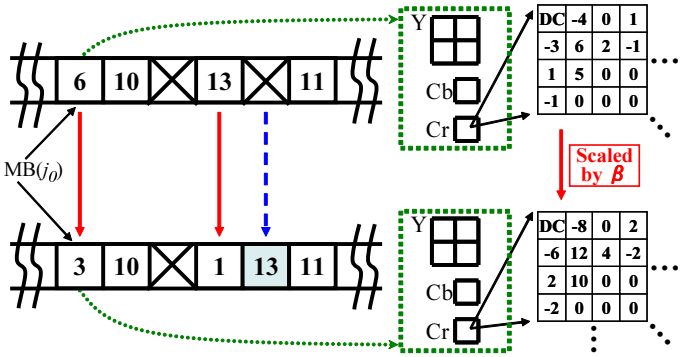


Fig. 3. Example: Encoding and decoding information

Cond.D with $\beta = 2$. DEC does not check MB(j_0+2) for possible *promotion* performed because MB(j_0+1) was not *excited*. DEC extracts "1" from MB(j_0+3) since it satisfies **Cond.C** with $\beta = 13$. DEC declares MB(j_0+4) as a *promoted* MB because $\mathcal{Q}(j_0+4) = \mathcal{Q}(j_0+3) \times 13$ and MB(j_0+4) fails **Cond.C**. DEC further extracts "0" from MB(j_0+5) because it fails **Cond.C** but satisfies **Cond.D** with $\beta = 11$. The decoding process continues until the entire message is extracted.

3.5 Handling INTER-MB

When dealing with INTER-MB which could occur in P and B-pictures, line 4 in **Algorithm 1** is modified as follows during MB *excitement*:

$$x \leftarrow \beta x + \text{sign}(x) \times y, \tag{7}$$

where x is a nonzero qDCTC in the MB (DC components are also included), and $y = (\beta - 1)/2$. The scaling equation given by Eq. (7) is derived in **Appendix**. Since all operations are carried out in integer mode in MPEG coding standard, y must be an integer. Hence, the multiplying factor β has to be an odd number, and 2 must be removed from $\S_1$ when processing an INTER-MB. In particular, we use $\beta \in \S_2$ when dealing with INTER-MBs where

$$\S_2 := \S_1 \setminus \{2\} = \{3, 5, 7, 11, 13, 17, 19, 23, 29, 31\}. \tag{8}$$

Next, an INTER-MB is *promoted* using the same operations as applied to *promote* an INTRA-MB, i.e., **Algorithm 2**. However, since an INTER-MB may be skipped or completely motion compensated, these MBs are ignored during the search for the next MB coded with qDCTCs (i.e., image signal in case of INTRA-MB or differential signal in case of INTER-MB) for necessary MB *promotion*. Refer to (Hanzo et al., 2007) for more information on MB type.
When extracting the embedded information from an INTER-MB, condition (5) is replaced by the following condition:

$$\text{mod}\ (x - \text{sign}(x) \times y, \beta) = 0. \tag{9}$$

Cond.C is also updated accordingly using $\S_2$ in place of $\S_1$. An INTER-MB is declared as encoding nothing if it fails **Cond.D**, and this happens when

$$\mathcal{Q} \in \{1, 2, 4, 8, 16\}. \tag{10}$$

Algorithm 3 is utilized to restore an *excited* INTER-MB, except that the division operation in line 4 is replaced by

$$x \leftarrow (x - \text{sign}(x) \times y)/\beta \tag{11}$$

for all nonzero qDCTC $x \in$ MB. Finally, a *promoted* INTER-MB is restored to its original state by using **Algorithm 4**.

3.6 Application to H.261, H.263, and MPEG-4

Since H.261 and H.263 reconstruct the DCT coefficients of an INTRA-MB using a linear equation similar to that of Eq. (1), **Algorithm 1** could be applied directly when *exciting* an INTRA-MB. Interestingly, similar to the case of MPEG-1/2, line 4 in **Algorithm 1** is also deduced to be replaced by Eq. (7) when dealing with an INTER-MB. Note that the reconstruction equation given by Eq. (2) differs depending on the value of MQ (i.e., odd or even), but Eq. (7) handles both cases simultaneously. This claim is justified in **Appendix**.

In case of MPEG-4, since the DC component of an INTRA-MB is reconstructed using MQ with a non-linear equation, the proposed method is restricted to INTER-MBs in MPEG-4. Similar to MPEG-1/2 and H.261/3, INTER-MB of MPEG-4 compressed video is *excited* using **Algorithm 1** and Eq. (7).

By *exciting* MB and *promoting* the affected MB, data embedding could be carried out while completely preserving the image quality of the original video. Procedure for decoding the embedded information, restoring the *excited* MB, and *demoting* the *promoted* MB could be easily derived from the aforementioned discussions and we omit the presentation here.

4. Reverse zerorun length

In this section, we propose RZL (reverse zerorun length) data representation scheme for achieving high embedding efficiency. For the rest of the chapter, embedding efficiency η refers to *the number of bits embedded per modification*. When embedding at a rate lower than the maximum carrier capacity (i.e., payload), more than k locations could be utilized to encode k bits of information while attempting to reduce the number of modifications. This idea is realized by exploiting the statistics of MB with respect to the proposed data hiding method.

4.1 Method

Suppose we treat each MB in its original state as "0" and the *excited* state as "1". The original/natural message induced by any video is thus an array of zeros. We exploit this statistic to encode a binary message μ while aiming to reduce the number of MB *excitements*.

Let μ be a binary message of length $|\mu|$, and let $k \in \mathbb{N}$ such that $k \geq 2$. μ is divided into segments of length k bits, and each segment $\mu(i)$ is processed one at a time for $i = 1, 2, \cdots, \lceil |\mu|/k \rceil$, where $\lceil z \rceil$ is the smallest integer greater than or equal to z. Unless specified otherwise, $|Z|$ denotes the cardinality of the set/array Z. Given $\mu(i)$ in ORS format as the input, the binary-to-RZL convertor (hereinafter referred as BIN2RZL) computes the decimal equivalent of $\mu(i)$ (denoted by $\mu_{10}(i)$), outputs $\mu_{10}(i)$ number of zeros, followed by an "1" to mark the end of the message segment. For example, suppose $\mu = 011101001$ and $k = 3$,

$$\begin{array}{lcll} \text{BIN2RZL}(011) & = & 0001 & \text{(i.e., three zeros followed by unity)} \\ \text{BIN2RZL}(101) & = & 000001 & \text{(i.e., five zeros followed by unity)} \\ \text{BIN2RZL}(001) & = & 01 & \text{(i.e., one zero followed by unity)} \end{array} \tag{12}$$

To embed the message μ using RZL, the output of BIN2RZL$(\mu(i))$ for all i are concatenated. The resulting sequence of zeros and ones are threated as a new message $\hat{\mu}$, and $\hat{\mu}$ is embedded

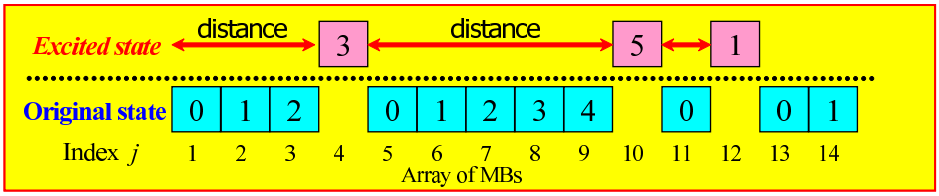


Fig. 4. Encoding and decoding example using RZL. Each MB is assumed to be usable for data embedding.

by *exciting* and *promoting* MBs as described earlier. In other words, RZL utilizes the distance between two consecutive "1"s output by BIN2RZL to encode information. Using the same example, $\hat{\mu} = 000100000101$ and **Figure 4** illustrates the resulting macroblocks after encoding $\hat{\mu}$ with RZL. Note that in this particular example, five MB *excitements* are required to encode $\mu = 011101001$ in case of ORS but only three MB *excitements* are required in case of RZL. Also note that given any binary sequence of length k, the output of BIN2RZL may be longer or shorter than k. Discussions on the number of *excitements* and locations required to encode $\mu(i)$ are presented in Section 4.2. For completion of discussion, RZL switches to ORS when $k = 1$.
To decode the message embedded in RZL format, $\hat{\mu}$ is first decoded by considering the state of each MB as described earlier. To ease the dicussion, let $\hat{\mu}(j)$ denote the jth bit in $\hat{\mu}$. The indices j_i where $\mu(j_i) = 1$ are stored in the array J in the order in which they appear. Note that $|J| \times k = |\mu|$. To decode $\mu(i)$, we set $j_0 = 0$ and consider the RZL-to-binary convertor (hereinafter referred as RZL2BIN) that converts $j_i - j_{i-1} - 1$ into binary representation of k digits, with zeros stuffing as the prefix when necessary. In particular,

$$\text{RZL2BIN}(\text{BIN2RZL}(\mu(i)), k) = \mu(i). \tag{13}$$

Note that the value k is required in RZL2BIN.

4.2 Performance analysis and discussion

The performance of RZL is compared to that of matrix encoding (ME) (Crandall, 1998). ME(k) is a data representation scheme that encodes k bits of information by utilizing $2^k - 1$ locations with at most one modification. In the context of the proposed data hiding method, location refers to MB with $\mathcal{Q} = \alpha \times \beta$ for $\beta \in \S_1$ or $\beta \in \S_2$ (depending on the MB type), and modification refers to MB *excitement*. ME is widely utilized because its payload could be traded for higher embedding efficiency. However, in order to store k bits of information, ME constantly requires $2^k - 1$ locations. On the other hand, the number of locations required by RZL varies depending on the message segment to be embedded. In particular, RZL occupies one location in the best case scenario (i.e., $\mu(i) = 000\cdots0$), and occupies 2^k locations in the worst case scenario (i.e., $\mu(i) = 111\cdots1$).
Since we may assume that the message μ is randomly distributed, i.e., $P(0) = P(1) = 0.5$, hence the expected number of locations required for encoding a k bit message segment in case of RZL is

$$\frac{2^k \cdot (2^k + 1)}{2} \times \frac{1}{2^k} = \frac{2^k + 1}{2} = 2^{k-1} + 0.5 \tag{14}$$

using the fact that $\sum_{i=1}^{n} i = n(n+1)/2$ for $n \in \mathbb{N}$. For embedding $\lceil |\mu|/k \rceil$ segments, it is difficult to formulate the estimated number of locations required. Nevertheless, $\lceil |\mu|/k \rceil \times$

k	1	2	3	4	5	6	7	8	9
ME	1	3	7	15	31	63	127	255	511
RZL	1	2.5	4.5	8.5	16.5	32.5	64.5	128.5	256.5

Table 2. Expected number of locations required

k	1	2	3	4	5	6	7	8
ME	1.000	0.667	0.429	0.267	0.161	0.095	0.055	0.031
RZL	1.000	0.800	0.667	0.471	0.303	0.185	0.109	0.062

Table 3. Expected payload for ME and RZL for various k [$\times 10^{-4}$ bits]

$(2^{(k-1)} + 0.5)$ gives a coarse estimation for the number of locations required to encode μ with segment size k when using RZL. **Table 2** records the expected number of locations required to encode a k bit message segment for $k = 1, 2, \cdots, 9$. For comparison purposes, the number of locations required for ME is also recorded in the same table. Obviously, RZL requires, in general, less locations (almost half) when compared to ME for encoding the same amount of information (i.e., k bits).

Next, we analyze the embedding efficiency η of ME and RZL (hereinafter referred as $\eta[\text{ME}(k)]$ and $\eta[\text{RZL}(k)]$, respectively). When ME is applied, an MB is *excited* with the probability of $1 - 1/2^k$, and hence $\eta[\text{ME}(k)] = k/(1 - 1/2^k)$. The subtraction of $1/2^k$ is due to the fact that an array of zeros (message segment) with length k occurs with probability $1/2^k$ in which case no modification is required. In other words, we always start with an array of zeros (be it of length $2^k - 1$ locations) and the only time we need not *excite* any MB is when the message segment to be embedded (k bits) is an array of zeros (which occurs with probability $1/2^k$). In case of RZL, we always need to *excite* an MB to mark the message, hence $\eta[\text{RZL}(k)] = k$, which is lower than $\eta[\text{ME}(k)]$. However, as recorded in **Table 2**, RZL requires less locations to encode the same amount of information when compared to ME, and hence a larger k could be utilized in case of RZL. As an example, suppose $|J| = 10000$, and we compute the payload of ME and RZL using various values of k. **Table 3** records the expected payload in fraction, i.e., actual payload divided by $|J|$. The results suggest that

$$\Omega[\text{RZL}(k+1)] \geq \Omega[\text{ME}(k)], \tag{15}$$

where $\Omega[Z(k)]$ denotes the payload for data representation scheme $Z \in \{\text{ME}, \text{RZL}\}$ with parameter k. Eq. (15) becomes more obvious for $k \geq 2$. Therefore, we may use a larger k for RZL to encode the same amount of information held by ME. Specifically, for a fixed message length $|\mu|$ and a fixed number of usable MBs, we expect $k_{\text{RZL}} \geq k_{\text{ME}}$. For the same reason, we expect higher embedding efficiency in RZL since

$$k/(1 - 1/2^k) \leq k + 1, \forall k \geq 1. \tag{16}$$

The aforementioned expectations are verified in **Section 6**. Thus, when embedding at any rate lower than the maximum carrier capacity, RZL could be utilized for achieving higher embedding efficiency, which in turn leads to smaller video bitstream size increment during data embedding.

5. Suppressing video bitstream size increment

When an MB is *excited* during data embedding, the magnitude of all nonzero AC components (including DC components in case of INTER-MB) are increased by a factor of at least β. Referring to the default VLC table specified in MPEG coding standard (Symes, 2004), it is observed that when the magnitude of a qDCTC increases, the length of its VLC increases even if its zerorun length remains the same. As a result, whenever an MB is *excited* (modified), the video bitstream size always increases. The simplest way to suppress video bitstream size increment is to code a new VLC table and utilize it in place of the original VLC table. However, the original VLC table in a compressed video could itself be a user-defined table or the default VLC table as specified by MPEG coding standard. Therefore, to achieve complete reversibility after data embedding, we must not code a new VLC table. Instead, we propose the following two independent solutions:

(i) Only small multiplying factors (eg. $\beta = 2$ or 3) are utilized so that the increase in code length (in VLC table) for each nonzero qDCTC is kept to the minimal. In particular, all prime factors no larger than $\phi \in \S_1$ is utilized for INTRA-MB, and all prime factors no larger than $\rho \in \S_2$ is utilized for INTER-MB. The restriction using ϕ and ρ reduces the payload since only a subset of β is considered for data embedding.

(ii) Instead of using all MBs with $\mathcal{Q} = \alpha \times \beta$ ($\beta \in \S_1$ and $\beta \in \S_2$ in case of INTRA and INTER-MB, respectively) for data embedding, only MBs that are made up of at most τ number of nonzero qDCTCs are utilized. This is a natural way to limit the bitstream size increment since less nonzero qDCTCs implies less multiplications by the factor β.

To ease the discussion, we define

$$\delta(V) = \text{filesize}(V') - \text{filesize}(V). \tag{17}$$

Note that $\delta(V) > 0$ since the modified video V' has larger filesize than the original video V. We use “filesize" and “video bitstream size" interchangeably for the rest of the chapter. Also, unless specified otherwise, “increase of video bitstream size" or $\delta(V_i)$ refers to the change of filesize for the entire video instead of “change of bitstream size per second" or such. The effect of each solution on payload and $\delta(V)$ are verified in **Section 6**. Last but not least, because higher embedding efficiency implies less modifications, which in turn leads to smaller $\delta(V)$, RZL proposed in **Section 4** could be utilized to further suppress $\delta(V)$ while trading off with payload.

6. Experimental results

Eight test videos are utilized to verify the performance of the proposed method. Each video is encoded by MPEG-1 compression standard using ISO/IEC TR 11172-5 (1998) at 1.5Mbps with 15 pictures in a GOP and picture type ratio of I:P:B = 1:4:10. More information on the test videos could be found in **Table 4**. Note that we need not evaluate the image quality of the modified video because when the modified video is fed into an ordinary decoder, it completely reconstructs the original video even compared at the bit-to-bit level. Nevertheless, we verified that each modified video has exactly the same PSNR value as its original counterpart. It is verified that the embedded information could be decoded and removed to restore the original video. Also, using Microsoft Windows Media Player (version 6.4.09.1130), it is verified that all modified videos could be played-back properly.

Video	Video description	Total Frames			Dimension	MB per frame
		I	P	B		
V_1	Coastguard	20	80	198	352×288	396
V_2	Container	20	80	198	352×288	396
V_3	Driving	20	80	198	352×240	330
V_4	Flower garden	10	40	98	352×240	330
V_5	Foreman	20	80	198	352×288	396
V_6	Hall monitor	20	80	198	352×288	396
V_7	Mobile calendar	20	80	198	352×288	396
V_8	A walk in the sqaure	30	120	298	352×240	330

Table 4. Information of standard test videos

6.1 Payload

First, we consider the payload offered by each $\mathcal{Q} = \alpha \times \beta$ for $\beta \in \S_1$ and $\beta \in \S_2$ in case of INTRA and INTER-MB[6], respectively. We record the payloads of V_1 :Coastguard as the representative example in **Table 5**. As expected, the payload is relatively low if we utilize only a specific pair of α and β for data embedding (eg. $\alpha = 5$ and $\beta = 3$ that provide merely 20-bits in INTRA-I). However, more than a pair of α and β could be utilized simultaneously to provide higher total payload. Note that the payload offered by $\mathcal{Q} = 1$ is always zero because $1 \notin \S_1$ and $1 \notin \S_2$. Also, in case of INTER-MB, the payload offered by $\mathcal{Q} \in \{2, 4, 8, 16\}$ is always zero since the only factor for these values is 2, which is not in the set $\S_2$. Similar results are observed for other test videos. For the rest of the chapter, payload refers to the sum of the payloads offered by each pair of α and β for $\beta \in \S_1$ or $\beta \in \S_2$ depending on the MB type.

Secondly, for each video, the average payload (bits per frame, hereinafter *bpf*) for I, P and B-picture are recorded in **Table 6**. When operating at full capacity, i.e., $(\phi, \rho, \tau) = (31, 31, 384)$, the payload of a video reaches its upper bound with respect to the proposed method. Using I-pictures of V_1 as the representative example, we elaborate the results. The value 4395-bits / 20 frames = 219.8*bpf* implies that, on average, out of 396 MBs in an I-picture, 219.8 MBs are equipped with coded MQ values that are each divisible by at least one $\beta \in \S_1$. Hence, on average, 219.8 bits could be embedded into each I-picture of V_1. The rest of the results are interpreted similarly.

In the full capacity mode, it is observed that the payload is influenced by the variation of spatial complexity (activity) in the video and similarity among frames. Regardless of the picture type, video of high variation in spatial complexity and low similarity among frames (eg. V_4 and V_7) offers high payload, and vice versa (eg. V_2 and V_6). On average, approximately 55.9%, 24.0%, and 23.6% of all MBs are usable for data embedding in I, P, and B-pictures, respectively. Also, for the same parameter setting, it is observed that regardless of the video, I-picture consistently yields the highest payload. This is because all MBs in I-picture are coded while only selected MBs in P and B-pictures are coded due to motion compensation.

Next, we investigate how the payload is influenced by solution (i) which is proposed in **Section 5** for suppressing filesize increase. When ϕ or ρ is reduced from 31, the payload reduces regardless of the picture type or video. Nevertheless, for each video, I-picture still yields the highest payload. When (ϕ, ρ) is reduced from $(31, 31)$ to $(13, 13)$, the reduction of payload is insignificant, i.e., an average drop of $\sim 2.6\%, \sim 4.1\%$, and $\sim 7.9\%$ are observed for I, P, and B-pictures, respectively. Similar results are observed for the reduction of (ϕ, ρ) from $(13, 13)$

$\mathcal{Q}$	INTRA-I	INTRA-P	INTER-P	INTRA-B	INTER-B
1	0	0	0	0	0
2	24	0	0	0	0
3[6]	282	1	0	0	0
4	601	6	0	0	0
5	763	18	3430	0	2672
6	727	22	0	0	0
7	568	17	2451	0	6503
8	411	15	0	0	0
9	330	9	0	0	0
10	211	5	664	0	3549
11	147	3	558	0	2618
12	116	3	0	0	0
13	105	3	340	0	1396
14	89	1	164	0	1088
15	20	1	46	0	931
16	1	1	0	0	0
17	0	0	7	0	750
18	0	0	0	0	0
19	0	0	0	0	260
20	0	0	0	0	145
21	0	0	0	0	62
22	0	0	0	0	18
23	0	0	0	0	13
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
Sum	4395	105	7660	0	20005

[6]Note that $\beta = 3$ is removed from $\S_2$ because there are INTER-MBs in some original videos in which case all of their qDCTCs are of the form $3z + \text{sign}(z)$ for integer z. This causes false detection of *excited* MB. Thus, in our experiments, $\S_2$ = {5,7,11,13,17,19,23,29,31} is considered.

Table 5. Distribution of usable MQ values for V_1 : Coastguard

to $(7,7)$. When (ϕ,ρ) is further reduced to $(2,5)$, the average payload of each video drops to approximately half of its original payload regardless of the picture type. With this relatively strict condition, i.e., $(\phi,\rho,\tau) = (2,5,384)$, we still achieve an average payload of ~ 104, ~ 38, and ~ 33*bpf* for I, P and B-pictures, respectively.

Now, suppose that data embedding is restricted to MBs with at most τ number of nonzero qDCTCs (i.e., solution (ii) in **Section 5**). The results for applying this restriction to the MBs extracted with $(\phi,\rho) = (31,31)$ and $(2,5)$ are recorded in **Table 7** for $\tau = 10, 4$ and 1. Payload for each type of picture generally decreases when τ is decreased. The results indicate that the reduction in payload is most severe in case of I-picture, followed by P and B-pictures. As an example, consider the payload for $(\phi,\rho) = (31,31)$. When τ is reduced from 384 to 10, the

Video	$(\phi,\rho)=(31,31)$			$(\phi,\rho)=(13,13)$			$(\phi,\rho)=(7,7)$			$(\phi,\rho)=(2,5)$		
	I	P	B	I	P	B	I	P	B	I	P	B
V_1	219.8	97.1	101.0	219.8	97.0	95.9	207.2	85.7	75.5	109.0	52.4	36.9
V_2	118.1	20.5	12.8	118.1	20.5	12.8	117.6	20.4	12.8	66.6	15.9	6.8
V_3	215.1	103.6	106.5	214.8	103.4	99.6	200.7	88.9	70.3	107.5	54.5	35.9
V_4	239.0	144.2	132.2	212.3	118.1	96.8	184.3	84.5	61.9	118.9	44.0	39.3
V_5	216.5	64.9	88.5	216.2	64.7	87.4	210.6	61.5	72.6	106.4	37.6	50.1
V_6	165.2	29.3	45.3	165.2	29.3	45.3	164.0	29.2	37.1	87.5	14.4	29.2
V_7	312.0	190.4	154.3	282.1	163.4	119.6	242.6	118.9	70.7	151.9	55.4	47.3
V_8	162.4	49.5	48.0	162.4	49.5	47.5	159.0	46.7	34.5	86.0	28.0	20.2

Table 6. Average payload per frame for each picture type with parameter $(\phi,\rho,\tau=384)$ and ORS $[bpf]$

Video	$(\phi,\rho)=(31,31)$								
	$\tau=10$			$\tau=4$			$\tau=1$		
	I	P	B	I	P	B	I	P	B
V_1	0.00	7.53	72.25	0.00	2.20	40.49	0.00	0.45	13.37
V_2	1.05	5.18	8.59	0.60	2.46	5.69	0.00	0.51	2.85
V_3	1.85	8.04	57.20	0.30	1.99	32.04	0.10	0.33	10.98
V_4	5.80	5.55	100.58	2.00	2.15	67.42	0.70	0.88	24.67
V_5	2.10	7.48	57.40	0.80	2.48	34.32	0.10	0.59	12.30
V_6	0.00	8.61	32.81	0.00	2.69	21.89	0.00	0.40	8.35
V_7	2.20	33.11	120.64	1.55	9.61	81.99	0.55	1.95	33.24
V_8	0.03	4.08	33.20	0.00	1.18	21.64	0.00	0.25	9.01
Video	$(\phi,\rho)=(2,5)$								
	$\tau=10$			$\tau=4$			$\tau=1$		
	I	P	B	I	P	B	I	P	B
V_1	0.00	4.28	27.66	0.00	1.19	15.71	0.00	0.28	5.29
V_2	1.00	4.01	4.76	0.55	1.85	3.22	0.00	0.38	1.72
V_3	1.10	4.93	19.93	0.25	1.11	11.40	0.10	0.19	3.74
V_4	3.20	1.28	29.53	1.00	0.55	20.26	0.20	0.23	7.50
V_5	1.25	4.53	33.34	0.40	1.54	20.31	0.05	0.36	7.22
V_6	0.00	4.09	20.20	0.00	1.55	11.87	0.00	0.33	3.61
V_7	0.55	9.46	36.63	0.35	3.01	24.74	0.20	0.59	10.00
V_8	0.00	2.56	14.08	0.00	0.86	9.23	0.00	0.21	4.03

Table 7. Average payload influenced by τ with respect to ORS $[bpf]$

payload is $\sim 1\%, \sim 14\%$ and $\sim 69\%$ of the original payload (i.e., when $\tau = 384$) of I, P and B-pictures, respectively. When τ is reduced to 4 and 1, the payload further decreases. Similar results are observed for $(\rho,\tau)=(2,5)$. Nevertheless, in case of $(\phi,\rho,\tau)=(2,5,1)$, B-pictures could still be utilized to embed information at the rate of $\sim 5.4 bpf$.
Finally, when RZL is applied to any combination of (ϕ,ρ,τ), we expect the resulting payload to be a fraction of the original payload achieved by (ϕ,ρ,τ). The fraction depends on the

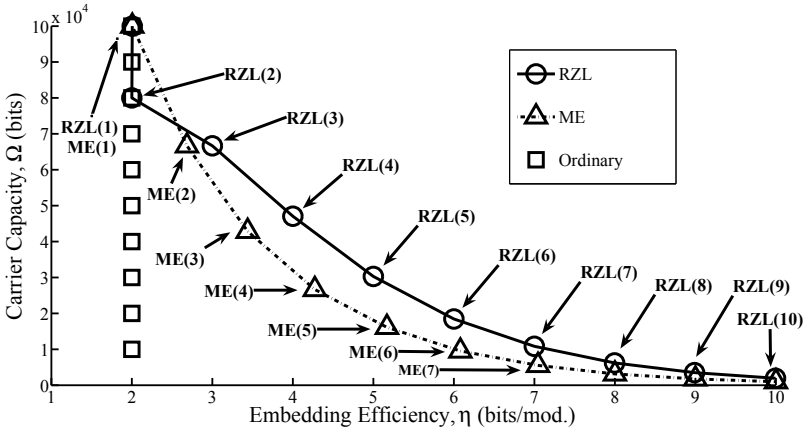


Fig. 5. Embedding efficiency vs. payload for ORS, ME, and RZL

parameter k of RZL, and it is recorded in **Table 3**. Nevertheless, when k increases, the payload decreases, and vice versa.

We conclude that the payload (*bpf*) for each type of picture is influenced by the parameters ϕ, ρ, and τ, as well as the application of RZL. Payload decreases whenever solution (i), (ii), or RZL is applied, but their roles in suppressing $\delta(V_i)$ become obvious in **Section 6.3**.

6.2 Performance of RZL

In this section, we simulate the embedding process using ME (Crandall, 1998) and RZL as the data representation scheme. For comparison purposes, we also consider ORS proposed in **Section 3.2**. As for the experiment parameters, $|J| = 100000$, and $k = 1, 2, \cdots, 7$. For each k, a message segment of length k bits is randomly generated and embedded. This process is repeated until all available locations ($|J|$ of them in total) are occupied. Next, the number of message segments embedded (ω) and the number of modifications (ν) are counted. This process is repeated for 1000 times, and the average of ω and ν are computed. The average embedding efficiency is computed as $\bar{\eta} = k \times \bar{\omega}/\bar{\nu}$. The aforementioned procedures are carried out using ORS, ME, and RZL. The results are shown in **Figure 5**.

As expected, the embedding efficiency for ORS is always 2 because an MB is *excited* with the probability of 0.5 when embedding a random binary message regardless of its length. Next, the results suggest that when we increase the parameter k from 1 to 2, $\eta[\text{RZL}(k)]$ stays the same, i.e., 2, as indicted by the vertical (solid) line in **Figure 5**. This is because every time we embed a message segment, we have to *excite* an MB regardless of its length k. Thus, $\eta[\text{RZL}(2)] = 2/1$.

For each k, it is observed that $\text{RZL}(k)$ is inferior to $\text{ME}(k)$ in terms of embedding efficiency. Rather, the results should be considered from the payload point of view. For instance, when we consider $\text{ME}(2)$ and $\text{RZL}(3)$, the carrier capacities are about the same, i.e., $\Omega[\text{ME}(2)] \sim \Omega[\text{RZL}(3)]$, but $\text{RZL}(3)$ achieves a higher embedding efficiency when compared to $\text{ME}(2)$, i.e., $\eta[\text{RZL}(3)] > \eta[\text{ME}(2)]$.

Ordinary representation scheme						
V_1	266.17	172.43	107.99	66.40	39.08	22.10
V_2	82.04	53.83	33.36	22.24	12.00	6.70
V_3	340.06	224.02	136.78	81.02	50.07	26.65
V_4	246.62	154.82	95.44	60.80	32.60	18.47
V_5	205.22	125.50	77.17	46.46	25.78	15.96
V_6	102.09	64.73	39.41	22.66	13.31	6.94
V_7	441.21	280.47	173.76	104.13	65.43	33.78
V_8	282.98	184.63	118.76	70.28	38.69	23.17

Matrix encoding with parameter k						
Video	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
V_1	203.74	104.05	51.46	24.66	12.81	6.60
V_2	57.65	29.34	15.39	7.37	3.20	2.20
V_3	258.22	129.78	64.47	32.44	16.06	9.08
V_4	180.97	94.27	46.16	24.15	11.56	4.99
V_5	151.58	73.92	37.72	19.00	8.26	4.86
V_6	76.63	40.68	19.54	10.12	4.86	2.48
V_7	343.39	165.56	86.15	42.59	21.37	10.67
V_8	210.62	103.03	53.39	25.81	12.60	5.76

Reverse zerorun length with parameter k						
Video	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
V_1	179.42	86.81	43.17	21.18	10.60	5.47
V_2	53.44	26.80	13.07	6.37	2.96	1.93
V_3	227.95	109.30	52.12	29.24	13.21	6.95
V_4	160.21	79.36	37.60	19.61	10.18	5.19
V_5	127.76	61.95	30.68	16.28	9.09	4.61
V_6	69.30	34.17	17.35	8.13	3.94	2.24
V_7	300.39	145.76	72.35	35.59	18.35	9.52
V_8	186.70	90.90	45.93	22.57	11.80	6.25

Table 8. Increase of video bitstream for ORS, ME(k), and RZL$(k+1)$ [Kbytes]

In fact, the embedding efficiency of RZL$(k+1)$ is always higher than that of ME(k) for $k \geq 2$. Moreover,

$$\Omega[\mathrm{ME}(k)] \leq \Omega[\mathrm{RZL}(k+1)] \tag{18}$$

and

$$\eta[\mathrm{ME}(k)] \leq \eta[\mathrm{RZL}(k+1)] \tag{19}$$

hold true simultaneously for $k \geq 3$. Since higher embedding efficiency implies less modifications (i.e., MB *excitements*), $\delta(V_i)$ is expected to be smaller when RZL$(k+1)$ is utilized as the data representation scheme as oppose to ORS or ME(k).

6.3 Suppression of video bitstream size increment

First, we compare the performance of ME (Crandall, 1998) and RZL in terms of $\delta(V_i)$ when embedding the same amount of information. Since solution (i) and (ii) are independent, it

Video	$(\phi,\rho)=(31,31)$				$(\phi,\rho)=(2,5)$			
τ	384	10	4	1	384	10	4	1
V_1	414.18	47.87	13.62	1.99	166.78	18.33	5.20	0.82
V_2	120.94	5.27	1.56	0.35	60.28	3.12	0.88	0.21
V_3	522.67	34.89	8.47	1.22	198.74	12.67	3.13	0.46
V_4	364.40	20.83	6.59	0.92	98.60	5.98	1.93	0.28
V_5	300.97	32.88	9.04	1.40	141.87	18.24	5.41	0.86
V_6	158.86	19.92	6.90	1.17	85.22	12.78	3.93	0.50
V_7	676.18	67.76	21.17	3.43	179.60	19.90	6.29	1.00
V_8	416.32	24.57	7.08	1.24	185.12	10.15	2.94	0.58

Table 9. Increase of video bitstream size with respect to (ϕ,ρ,τ) [kbytes]

is sufficient to consider a particular set of parameters and infer the results for other settings. In partcular, we consider $(\tau,\phi,\rho)=(384,31,31)$ which causes the largest $\delta(V_i)$ during data embedding. Since

$$\Omega[\mathrm{ME}(k)] \leq \Omega[\mathrm{RZL}(k+1)] \leq \Omega[\mathrm{ORS}], \tag{20}$$

we embed a message of length $\Omega[\mathrm{ME}(k)]$ bits into each video by using ORS, $\mathrm{ME}(k)$, and $\mathrm{RZL}(k+1)$. For fair evaluation purposes, we ensured that each type of picture (i.e., I, P and B) holds the exact same amount of information when using different representation scheme. The experiment is repeated for $k=2,3,\cdots,8$, and the resulting filesizes are recorded in **Table 8**. As expected, when either ME or RZL is applied, $\delta(V_i)$ is significantly suppressed. When embedding the same amount of information, $\mathrm{RZL}(k+1)$ consistently yields the smallest $\delta(V_i)$ when compared to ORS and $\mathrm{ME}(k)$ regardless of the value k. For example, using the filesize increased in case of ORS as the reference, the average reduction of $\delta(V_i)$ are 25% and 34% for $\mathrm{ME}(2)$ and $\mathrm{RZL}(3)$, respectively. However, when we consider $\mathrm{ME}(7)$ and $\mathrm{RZL}(8)$, the average reduction of $\delta(V_i)$ are 69% and 72%, respectively. In other words, the superiority of RZL over ME in suppressing $\delta(V_i)$ is more obvious for smaller k and become less obvious as k increases, which agree with the simulation results shown in **Figure 5**. We conclude that both ME and RZL are capable in suppressing $\delta(V_i)$, and RZL outperforms ORS and ME when embedding the same amount of information.

Next, we verify that solution (i) and (ii) proposed in **Section 5** are also capable in suppressing $\delta(V_i)$ caused by data embedding. For each test video, we consider $\delta(V_i)$ after embedding at the maximum rate with respect to parameter (ϕ,ρ,τ). For simplicity, we utilize ORS as the data representation scheme. The results are recorded in **Table 9**. Using V_1 and the parameter setting of $(\phi,\rho)=(31,31)$ as the representative example, we elaborate the results. When $\tau=384$, $\delta(V_1)=414.18$ kbytes, which is the upper bound of $\delta(V_1)$ for the proposed method. Once τ is reduced to 10, $\delta(V_1)$ reduces to 47.87 kbytes, or equivalently $\sim 1/9$ of $\delta(V_1)$ for $\tau=384$. When τ is further reduced from 10 to 4, $\delta(V_1)$ is suppressed to 13.62 kbytes. A negligible $\delta(V_1)$ of ~ 2 kbytes is observed when $\tau=1$, but the payload is also significantly reduced (refer to **Table 7**). The result for other videos and the setting of $(\phi,\rho)=(2,5)$ are interpreted similarly.

To better visualize the suppression of $\delta(V_i)$, we consider coding efficiency ε that is defined as *the number of message bits embedded per video bit increased*. Higher ε implies smaller $\delta(V_i)$, and vice versa. The results for ε are recorded in **Table 10**. On average, 0.0087 bits (from the message) are encoded per increased video bit in case of $(\phi,\rho,\tau)=(31,31,384)$. In other

Video	$(\phi,\rho)=(31,31)$				$(\phi,\rho)=(2,5)$			
τ	384	10	4	1	384	10	4	1
V_1	9.5	38.0	73.5	164.5	10.0	38.7	75.3	160.2
V_2	6.6	49.4	104.6	208.7	8.0	50.3	110.1	210.8
V_3	7.9	42.0	93.9	221.2	8.4	42.0	91.6	203.1
V_4	7.1	59.4	124.3	327.5	8.4	60.8	127.5	327.2
V_5	11.0	44.6	94.6	216.2	13.0	46.8	93.7	207.2
V_6	11.2	44.0	80.5	175.3	12.5	41.3	76.8	178.9
V_7	9.4	47.9	98.2	240.2	11.5	49.2	99.9	249.1
V_8	7.4	51.6	113.7	266.3	7.9	54.2	118.3	260.2

Table 10. Coding efficiency for various (ϕ,ρ,τ) at 1.5Mbps $(\times 10^{-3})$

words, embedding a single message bit causes an increment of 115 bits in the video bitstream. The rest of the results are interpreted similarly. The results show that ϵ increases when τ is reduced. In the best case scenario, an average increase of four bits in the video bitstream size is observed for every message bit embedded. Interestingly, $\varepsilon(31,31,\tau) \sim \varepsilon(2,5,\tau)$ for all τ considered. We conclude that both solution (i) and (ii) are capable in increasing the coding efficiency, and solution (ii) has greater impact in suppressing $\delta(V_i)$ than solution (i).
In conclusion, the results suggest that RZL, solution (i), and (ii) are capable in suppressing $\delta(V_i)$ while trading off with payload $\Omega(V_i)$.

6.4 Influence of video bitrate

In this section, we investigate the influence of video bitrate on the performance of our method by re-compressing V_i at 1.0, 2.0, 2.5 and 3.0 Mbps for $i = 1,2,\cdots,8$. Here, we only consider $(\phi,\rho,\tau)=(31,31,384)$ using ORS. The payload of V_i compressed at these bitrates are recorded in **Table 11**. The results indicate that, in general, the payload of I and P-pictures decrease as the video bitrate increases. A possible explanation to these observations is that I and P-pictures are finely quantized (with the same MQ value) when the bitrate is high since there are enough (available) bits to code them before reaching the bitrate restriction. In other words, the rate controller is frequently in idle state when coding at a higher bitrate, resulting in less coded MQ values when coding I and P-pictures. The remaining bits are utilized to code the B-pictures, and different MQ values are coded for achieving the specified bitrate. For that, the payload of B-picture shows no obvious trend as the video bitrate varies.
Next, we consider the influence of bitrate on $\delta(V_i)$. Since the payload and video length (i.e., number of frames) vary for each video, embedding at the maximum payload of each video does not lead to a conclusive result. Instead, to fairly evaluate our method, we embed information at a specific rate into each type of picture for all video bitrates. In particular, for each video V_i, we embed 55.8, 1.5, and 10.1 *bpf* into each I, P, and B-picture, respectively, for $i = 1,2,\cdots,8$. These values are selected (i.e., payload of V_2 at 3.0 Mbps) because these are the smallest payloads for all videos and for all bitrates considered. $\delta(V_i)$ after data embedding is recorded in **Table 12** in unit of kbytes. As expected, the results suggest that data embedding always leads to video bitstream size increment in the modified video. $\delta(V_i)$ generally increases as the (compression) bitrate increases because there are more nonzero qDCTCs in video compressed at higher bitrate, and vice versa. Again, we stress that the video bitstream increment

Video	1.0Mbps			2.0Mbps		
	I	P	B	I	P	B
V_1	251.7	127.1	105.5	196	65.8	98.8
V_2	142.9	31.9	22.8	95.2	11.9	14.2
V_3	233.6	130	106.7	194.8	73.5	102.1
V_4	246.8	172.6	67.7	227.1	130.6	133.9
V_5	241.3	113.5	93.6	195	47.4	64.3
V_6	181.9	50.9	77.2	162.7	28.8	31.5
V_7	324.3	235.6	96	300.4	166.4	148.3
V_8	237.1	112	60.2	184.4	48.8	58.4

Video	2.5Mbps			3.0Mbps		
	I	P	B	I	P	B
V_1	176.5	41.2	80.5	161.7	27.4	58.2
V_2	74.3	2.1	12.3	55.8	1.5	10.1
V_3	180.5	52.4	83.3	167.8	37.4	66.7
V_4	217.7	107.2	125.6	205.9	85.2	120.7
V_5	173.9	32.8	51.3	152.5	22.4	41.4
V_6	138.7	22.9	29.2	118.9	19.4	31.4
V_7	289.2	141.8	148.3	275.8	109	154.9
V_8	125.6	24.3	33.9	107.9	14.6	31.0

[7]Refer to **Table 6** for payload at 1.5Mbps

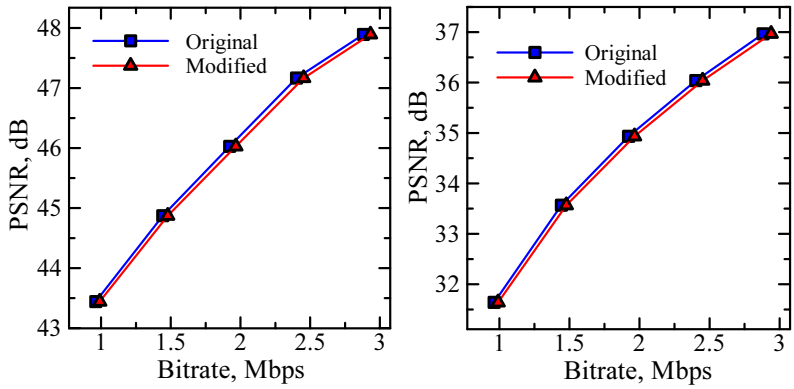
Table 11. Payload for video bitrate 1.0, 2.0, 2.5,and 3.0Mbps $[bpf]$

Bitrate [Mbps]	1.0	1.5	2.0	2.5	3.0
V_1	25.7	37.3	49.9	59.0	70.0
V_2	36.1	45.9	56.4	65.0	64.1
V_3	33.3	48.1	63.4	76.5	90.1
V_4	19.8	27.0	32.7	37.1	40.5
V_5	28.4	36.9	42.6	47.3	51.9
V_6	28.8	35.1	39.3	43.0	43.8
V_7	35.8	44.9	53.4	61.7	73.1
V_8	77.2	86.9	96.2	102.1	110.6

Table 12. Increase of video bitstream size for various bitrates [Kbytes]

$\delta(V_i)$ could be suppressed by tuning the parameters (ϕ, ρ, τ), and/or utilizing ME (Crandall, 1998) or RZL as the data representation scheme.

Last but not least, we utilize the results from **Table 12** and PSNR values to plot the graph of bitrate vs. PSNR in **Figure 6** for both the original and modified videos. V_2 and V_7 are shown here as the representative results. Instead of looking at a fixed bitrate and compare the PSNR values, we should consider a particular PSNR value and compare the bitrates because our method does not distort the image quality of the video during data embedding. For the same PSNR value, the modified video is of higher bitrate than the original video due to MB



(a) V_2: Container (b) V_7 : Mobile calendar

Fig. 6. Graphs of bitrate vs. PSNR

excitements and *promotions*. In other words, to achieve a specific PSNR value, MPEG-1 requires lower bitrate than our method (i.e., MPEG-1 coupled with the proposed data hiding method). In particular, when embedding 55.8, 1.5 and 10.5 *bpf* into each I, P, and B-picture, respectively, the video bitrate is increased by 2.3% and 4.2% for V_2 and V_7, respectively, which are very small increments. Similar results are observed for other videos. Thus, we stress again that our data hiding method preserves the image quality of the video at the expense of video bitstream size increment.

7. Conclusions

A novel data hiding method that completely preserves the image quality of the host video was proposed in the compressed video domain. During video playback, the modified video completely reconstructs the original video even compared at the bit-to-bit level. This method is reversible where the original video could be restored from the modified video. RZL (reserve zerorun length) data representation scheme was proposed to improve the embedding efficiency by trading off with payload. It was theoretically and experimentally verified that RZL outperforms matrix encoding in terms of payload and embedding efficiency. Basic performance of this method was evaluated using various MPEG-1 compressed videos. Results suggest that, approximately 55.9%, 24.0% and 23.6% of all MBs are usable for data embedding in I, P and B-pictures, respectively, which is sufficient for applications including indexing and annotation. The video bitstream size increases up to 40% when operating at full-capacity and this can be suppressed by RZL or any of the two independent solutions proposed. In the best case scenario, an average increase of four bits in the video bitstream size is observed for every message bit embedded.

As future works, we seek for possible extensions of our data hiding method to withstand hostile environment so that the embedded message could still be extracted after common image processing attacks. We should explore complete quality preserving data hiding in other domains such as still picture and audio. At the same time, we should succeed in suppressing video bitstream size increment due to data embedding without sacrificing payload. We also seek for the applications of RZL in other data hiding domains.

8. References

Bodo, Y., Laurent, N. & Dugelay, J.-L. (2004). Watermarking video, hierachical embedding in motion vectors, *IEEE Proc. of ICIP*, pp. 739–742.

Budhia, U., Kundur, D. & Zourntos, T. (2006). Digital video steganalysis exploiting statistical visibility in the temporal domain, *IEEE Trans. on Information Forensics and Security* **1**(4): 502–516.

Cox, I., Miller, M. L. & Bloom, J. A. (2002). *Digital Watermarking*, Morgan Kaufmann Publishers.

Crandall, R. (1998). Some notes on steganography.
URL: *http://os.inf.tu-dresden.de/westfeld/crandall.pdf*

Hanzo, L., Cherriman, P. & Streit, J. (2007). *Video Compression and Communications*, IEEE PRESS.

Hartung, F. & Girod, B. (1996). Digital watermarking of raw and compressed video, *Proc. SPIE Digital Compression Technologies and Systems for Video Communication*, Vol. 2952, Berlin, Germany.

ISO/IEC TR 11172-5 (1998). Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s – part 5: Software simulation, *Technical report*, ISO/IEC, Switzerland.

ITU-T H.261 (1990). Video codec for audiovisual service at p×64 kbit/s, *Technical report*, International Telecommunication Union, Geneva.

Johnson, N. F., Duric, Z. & Jajodia, S. (2003). *Information Hiding: Steganography and Watermarking - Attacks and Countermeasures*, Kluwer Academic Publishers.

Katzenbeisser, S. & Petitcolas, F. (2000). *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House Publishers.

Kiya, H., Noguchi, Y., Takagi, A. & Kobayashi, H. (1999). A method of inserting binary data into MPEG video in the compressed domain, *IEICE Trans. on fundamentals of electronics, communications and computer sciences* **82**(8): 1485–1492.

Kurosaki, M. & Kiya, H. (2002). Error concealment using a data hiding technique for MPEG video, *IEICE Trans. on fundamentals of electronics, communications and computer sciences* **E85-A**(4): 790–796.

Liu, Z., Liang, H., Niu, X. & Yang, Y. (2004). A robust video watermarking in motion vectors, *IEEE Proc. of International Conference of Signal Processing*, pp. 2358 – 2361.

Nakajima, K., Tanaka, K., Matsuoka, T. & Nakajima, Y. (2005). Rewritable data embedding on MPEG coded data domain, *IEEE Proc. of ICME*, pp. 682–685.

Ni, Z., Shi, Y.-Q., Ansari, N. & Su, W. (2006). Reversible data hiding, *IEEE Trans. on Circuits and Systems for Video Technology* **16**(3): 354–362.

Pranata, S., Wahadaniah, V., Guan, Y. L. & Chua, H. C. (2004). Improved bit rate control for real-time MPEG watermarking, *EURASIP Journal on Applied Signal Processing* **2004**(14): 2132–2141.

Qiu, G., Marziliano, P., Ho, A. T., He, D. & Sun, Q. (2004). A hybrid watermarking scheme for H.264/AVC video, *IEEE Proc. of ICPR*, pp. 865–868.

Sarkar, A., Madhow, U., Chandrasekaran, S. & Manjunath, B. S. (2007). Adaptive MPEG-2 video data hiding scheme, *SPIE Proc. of Security, Steganography, and Watermarking of Multimedia Contents IX*, pp. 489–497.

Symes, P. (2004). *Digital Video Compression*, McGraw-Hill.

Takayama, M., Tanaka, K., Yoneyama, A. & Nakajima, Y. (2006). A video scrambling scheme applicable to local region without data expansion, *IEEE Proc. of ICME*, pp. 1349–1352.

Wong, C., Yu, H. & Zheng, M. (2003). A DCT-based MPEG-2 transparent scrambling algorithm, *IEEE Trans. Consumer Electronics* **49**(4): 1208–1213.

Wong, K. & Tanaka, K. (2007). Mquant-based data hiding method in MPEG domain, *IIEEJ Proc. of Image Electronics and Visual Computing Workshop*.

Wong, K., Tanaka, K. & Qi, X. (2006). Multiple messages embedding using DCT-based Mod4 steganographic method, *LNCS Proc. of International Workshop on Multimedia Content Representation, Classification, and Security*, pp. 57–65.

Xu, C., Ping, X. & Zhang, T. (2006). Steganography in compressed video stream, *IEEE Proc. of International Conference on Innovative Computing, Information and Control*, pp. 269–272.

Yanagihara, H., Nakajima, Y., Matsuoka, T. & Tanaka, K. (2005). Advancement of streaming contents using watermark indexing (part iii) - development of software video player with watermark detection, *IIEEJ Proc. of 33rd Annual Conference*, pp. 77–78.

Zeng, W. & Lei, S. (1999). Efficient frequency domain video scrambling for content access control, *ACM Proc. of 7th International Multimedia Conference*, pp. 285–294.

Zhang, J., Li, J. & Zhang, L. (2001). Video watermark technique in motion vector, *IEEE Proc. of Brazilian Symposium on Computer Graphics and Image Processing*, pp. 179 – 182.

Appendix

[A]. Scaling equation for MPEG-1/2

We derive the scaling equation for INTER-MB. Assume that $x > 0$ (thus sign(x) = 1) and assume that the coded Mquant value $\mathcal{Q}$ is updated from $\alpha \times \beta$ to α, which is the same as $\mathcal{Q}/\beta$. Based on Eq. (1), we want to update x using $x \leftarrow \beta \times x + y$ and hence, we need to find y such that Eq. (21) holds true.

$$\begin{aligned} rec &= \frac{[2 \times x + \text{sign}(x)] \times \mathcal{Q} \times \text{QT}_2}{16} \\ &= \frac{[2 \times (\beta \times x + y) + \text{sign}(\beta \times x + y)] \times \mathcal{Q}/\beta) \times \text{QT}_2}{16}. \end{aligned} \tag{21}$$

For simplicity, we also assume that sign(x) = sign$(\beta \times x + y)$ since there is no restriction on the sign of y. After simplification and trimming of Eq. (21), we obtain:

$$[2 \times x + \text{sign}(x)] = [2 \times (\beta \times x + y) + \text{sign}(\beta \times x + y)]/\beta.$$

Since sign(x) = sign$(\beta \times x + y) = 1$, we have the following:

$$2 \times x + 1 \quad = \quad [2 \times (\beta \times x + y) + 1]/\beta$$

Simplifying the equation gives us $\beta = 2y + 1$, and thus $y = (\beta - 1)/2$ for $x > 0$. Similarly, we can derive that $y = (1 - \beta)/2$ for $x < 0$.

[B]. Scaling equation for H.261, H.263 and MPEG-4

Similar to MPEG-1/2, we assume that the Mquant value is $\mathcal{Q} = \alpha \times \beta$. First, we consider the case when $\mathcal{Q}$ is odd. Referring to Eq. (2), we want to find y so that the following equation holds true:

$$\text{sign}(x) \cdot [2\alpha\beta|x| + \alpha\beta] = \text{sign}(\beta x + y) \cdot [2\alpha|\beta x + y| + \alpha] \tag{22}$$

Again, we have the freedom for setting the sign of y, and we force y to have the same sign as x. Hence, Eq (22) could be simplified to

$$2\beta|x| + \beta = 2|\beta x + y| + 1. \tag{23}$$

Assume that $x > 0$, then we obtain

$$2\beta x + \beta = 2\beta x + 2y + 1. \tag{24}$$

Simplifying Eq. (24) leads us to $y = (\beta - 1)/2$ for $x > 0$. Similarly, we can derive that $y = (1 - \beta)/2$ for $x < 0$.
Now suppose $\mathcal{Q}$ is even. We want to find y so that the following equation holds true:

$$\text{sign}(x) \cdot [2\alpha\beta|x| + \alpha\beta - 1] = \text{sign}(\beta x + y) \cdot [2\alpha|\beta x + y| + \alpha - 1] \tag{25}$$

When we assume that x and $\beta x + y$ are both of the same sign, Eq. (25) simplifies to Eq. (23), and the aforementioned discussion could be applied directly. Therefore, $y = (\beta - 1)/2$ when $x > 0$ and $y = (1 - \beta)/2$ when $x < 0$ for both odd and even $\mathcal{Q}$.

Digital Watermarking Techniques for AVS Audio

Bai-Ying Lei[1], Kwok-Tung Lo[1] and Jian Feng[2]
[1] *The Hong Kong Polytechnic University, Hong Kong, China*
[2] *Hong Kong Baptist University, Hong Kong, China*

1. Introduction

One of the biggest technological events of the last two decades was the invasion by digital media in every aspect of our life. The proliferation of the Internet, which has enabled audio material (copyrighted or not) to be widely disseminated, has also made wiser the use of many compressed audio formats such as MP3 (MPEG Layer 3), AAC (Advanced Audio Coding), WMA (Windows Media Audio) and AVS (Audio Video Standard) at a global scale. AVS is an emerging coding standard fully developed by China (AVS Audio Expert Group, 2005). As digital data can easily be copied and distributed, protection of the copyright of media data becomes one of the most important topics in the Internet world. Digital watermarking has been identified as one of the feasible solutions for content protection and received considerable attention in recent years. Watermarking is the process of embedding hidden content protection information into digital data by making small modifications on the data. The biggest challenge of watermarking techniques is to embed the copyright information without affecting the human perception.

Digital audio watermarking technology helps to prevent or reduce unintentional and intentional copyright infringements by either notifying a recipient of any copyright or licensing restrictions or inhibiting or deterring unauthorized copying (He, 2008; Nedeljko & Seppanen, 2008). However, once the user downloads the music with the valid key, the embedded watermark can be easily removed by some software and hence it infringes on the purpose of audio watermarking. How to protect audio copyright efficiently and effectively has become a great challenge. The conflicts with MP3 copyrights make AVS audio copyright protection a problem which must resolve as AVS audio doesn't take copyright protection scheme into consideration. The development of effective watermarking schemes in the AVS compressed domain will remain an important research area in the future.

In order to achieve copyright protection of audio, the watermarking scheme needs to meet the following requirements:

- The watermark should be inaudible to human ears;
- Watermark detection should be done without referencing the original audio signals;

- The watermark should be undetectable without prior knowledge of the embedded watermark sequence;
- The watermark is directly embedded in the audio signals, not in a header of the audio;
- The watermark is robust to resist common signal processing manipulations such as filtering, compression, filtering with compression, and so on.

The organization of this chapter is as follows. Following the Introduction section, in Section 2, the emerging Chinese Audio Video Standard (AVS) for digital audio will be described. An overview of AVS audio coding will be given in this section. The state of the art audio watermarking techniques and related works are surveyed in Section 3. The psychoacoustic modelling for AVS and audio watermarking is described in Section 4. Section 5 introduces chaotic sequence and presents a comparative study with PN sequence in the field of watermarking. The new perception-based watermarking scheme for AVS audio is introduced in Section 6. The proposed algorithm embeds the chaotic signal as watermark in parts of the audio data that are masked or not perceptible because of psycho-acoustic laws. Finally, the chapter is concluded in Section 7.

2. Overview of AVS Audio

AVS is an emerging coding standard fully developed by China (AVS Audio Expert Group, 2005). AVS includes four main technical standards: system, video, audio and digital right management and support standard-consistency test. AVS has great local advantages with its own independent intellectual property rights in China. Currently, AVS is used by China Unicom as its IPTV standard. Meanwhile, it is also adopted in the CMMB (China Multimedia Mobile Broadcasting) and TD-SCDMA (Time Division - Synchronous Code Division Multiple Access) which is the third generation mobile telecommunication system developed by China and is currently deployed by China Mobile. It is evident that AVS audio will be given the first priority by IPTV and mobile TV operators in China. This indicates that the prospective future and broad development of the AVS standard.

The primary goal that the AVS Audio workgroup wanted to achieve, on the premise of developing their own intellect property, is to establish an advanced Chinese audio coding and decoding standard with general performance equivalent or superior to MPEG AAC. AVS Audio codec supports mono, dual and multichannel PCM audio signal and sampling rate of the input signal ranges from 8 KHz to 96 KHz, and the output bitrate is from 16kbps to 96kbps per channel. AVS audio can be applied in different fields of information industry such as high-resolution digital broad-cast, high-density laser and digital storage media, wireless broadband multimedia communication, internet broadband streaming media and portable media player. Compared with the MPEG AAC, AVS audio has its own characteristics and merits. For example, it introduces the Context-dependent Bit-plane Coding (CBC) for entropy coding, which has higher coding efficiency than the existing ones.

The AVS audio codec is based on a perceptual audio coding model, which has similar architecture and characteristics with most of perceptual audio coders. The lossy compression systems exploit both perceptual irrelevancy and statistical redundancy to

achieve the coding gain. But many new algorithms are designed for the core modules, which are different from other solutions. The block diagram of the AVS audio codec is shown in Fig. 1. As shown in Fig.1(a), the encoder manipulates digital audio signal and outputs the compressed bitstream. When the input audio samples enter the encoder, the window switch determines the length of analysis block depending on the transients. IntMDCT is introduced as the time-frequency analysis. Post Quantization Square Polar Stereo Coding (PQ-SPSC) is adopted to improve stereo audio signals' quality. CBC (Context-dependent Bit-plane Coding) performs entropy coding of quantized spectrum data and scale factors. Finally, the bitstream formatter outputs the bitstream in a suitable packet format. The decoding process is the inverse of encoding process as illustrated in the Fig. 1(b). The decoder recovers and decodes the quantized audio spectra of the bitstream with the process of the reconstructed spectra. At first, AVS audio bitstream is decoded in the CBC decoding part, and then the bitstream is fed into PQ-SPSC module before the dequantization module according to the header information. The frequency domain audio signal in the window switch module is transformed into the spatial domain in the Inversed IntMDCT module. At last, it outputs the PCM signal.

In order to meet different demands, AVS Audio coding technology adopted two profiles: main profile and scalable profile. Main profile is high quality and high complexity, while scalable profile has scalable bitrate and quality. In scalable profile, the coded bitstream is composed of base layers and some enhanced layers, and it can dynamically adapt variable bandwidth and the terminal decoding capability. The coding quality of AVS Audio main profile is equivalent or superior to that of MPEG 2 AAC LC profile. The coding and decoding complexity of AVS Audio main profile is higher than that of MPEG 2 AAC LC profile. Moreover, AVS Audio supports scalable coding.

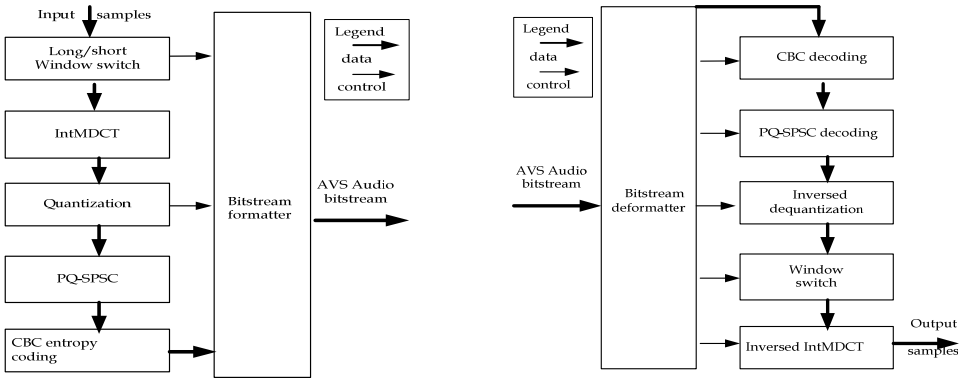


Fig. 1. AVS audio codec diagram (AVS Audio Expert Group, 2005)

3. Review on Audio Watermarking Techniques

Digital data protection and copyright issues have become more and more important in the face of today's technology. As a solution to copyright protection issues, digital

watermarking technology (Swanson et al., 1998; Wang 1998; Bassia et al., 2001; Podilchuk & Delp, 2001; He, 2008; Nedeljko & Seppanen, 2008) is gaining attention as a new method of protecting copyrights for digital data. The basic idea to avoid the unauthorized use and distribution of proprietary data is to sign every signal by means of an imprint characteristic of the source owner or distributor. This signature should be hidden in the signal and should not change its quality. The hiding procedure can take into account the human auditory imperfect detection. This is the basic idea of watermarking. Over the years, watermarking had been applied in various applications. In general, the applications areas of watermarking include the following:

- Copyright protection;
- To prove the digital media ownership;
- The tamper-proofing to check the data integrity;
- The covert communications;
- To exchange messages secretly embedded within multimedia data and the captioning;
- To embed descriptive and useful information within audio or video for applications like audio indexing.

For copyright-related applications, the embedded watermark is expected to be immune to various kinds of manipulations to some extent and acceptable in terms of perceptual quality. Therefore, watermarking schemes for copyright-related applications are typically robust (Seok et al., 2002), i.e. they are designed to ignore or remain insensitive to manipulations. Although a watermark is designed to be imperceptible to humans, the embedding is certainly intrusive and incurs distortion to the content. In some authentication applications (Zmudzinski & Steinebach, 2009) where any tiny changes to the content are not acceptable, the embedding distortion has to be compensated perfectly. In an attempt to remove the watermark so as to completely recover the original media after passing the authentication process, reversible watermarking schemes have been proposed in the last few years. Reversibility with media-independent embedding capacity will also be in the research agenda in the future for authentication applications. Although some perceptual models have been proposed to ensure low embedding distortion, how distortion and robustness could be optimized is still an open question and we expect new models will be proposed in the future. Requirements of digital watermarking vary across applications. The main requirements are low distortion, high capacity, and high security. However, meeting all the three requirements simultaneously is usually infeasible. Thus, trade-offs are frequently made to optimize the balance for each specific application. In many applications where original media is not available at the watermark decoder, blind detection of the watermark without any prior knowledge about the original is desirable. Although many solutions have been proposed by researchers in the past decade, psychoacoustic modelling needs to be further explored as it is the fundamental problems for most watermarking systems.

Depending on the embedding domain of the watermark, the existing audio watermarking techniques can be generally classified into two main categories: time domain techniques and compressed domain techniques. For time domain techniques, the watermark is inserted directly in original audio signal without a loss of sound quality. Common techniques

include the phase or amplitude modulation, quantization scheme and the echo hiding scheme. In (Bender et al. 1996), the information is embedded by modulating the phase of the audio signal. In (Bassia et al. 2001), it usually replaces the least significant bit (LSB) of the original audio signal with pseudorandom sequences, which can be viewed as a user identification number. They found a set of schemes that embed watermarks by adding predefined very weak noise signals to audio such that the changes are inaudible. The shortcoming of these sorts of watermarks is that they are fragile to most signal processing attacks including audio compression. Another technique exploits the insensitivity of the human ear to very short delay echoes. Although there are a few variations within this group, we collectively refer to them as echo hiding techniques (Gruhl et al. 1996; Seob et al. 2005; Chen et al. 2008). Echo hiding is the technique that embeds the watermark by introducing an echo which can effectively embed imperceptible information into an audio signal. One outstanding shortcoming of these approaches is that the watermark embedding success is signal and/or delay dependent. Moreover, resolving this problem tends to make the system unacceptably complex. But the watermark embedding process is signal dependent and detection rules are very lenient due to the fact that the information can be detected by anyone—even those without a special key. The echo hiding technique proposed by (Gruhl et al. 1996) is based on human insensitivity to audio distorted with a fused "echo" signal. Basically, the echo is defined with two delay times (offsets) to specify the embedding of binary "1" and "0". Their experimental tests showed that the relative volumes of the original and the echo signals control the rate of recovery of the watermark, and that the offset largely determines the perceptibility of the modifications. Echo hiding, in general, can be a guaranteed to an extent for a limited range of audio types. Further research is still under development to improve its performance on other types of audio data. The spread spectrum (SS) technique spreads the watermark all over the host signal and makes the attackers hard to locate the watermarks. The SS scheme (Kirovski & Malvar, 2003) requires psycho-acoustic adaptation for inaudible noise embedding. This adaptation is rather time-consuming. Another disadvantage of SS scheme is its difficulty of synchronization.

Since most multimedia products are distributed in compressed format, numerous methods in compressed domain have been reported. In (Lan et al. 2007), they proposed a perceptual based scalable AVS-DRM encryption model for AVS audio in order to protect audio content. The perception classification and multiple security levels are adopted and implemented in AVS audio codec. In (Li et al. 2006), a scalable lossless watermarking built on Advanced Audio Zip was developed with watermarking scalability and adaptiveness which address the problem of nonadptiveness of lossless watermarking and irreversible distortion problems. Tachibana, et al. (2002) proposed to use a two-dimensional pseudorandom array (PRA) as watermarking identification key to embed and extract watermarks in MPEG-2 AAC bitstream. The multiple watermarking scheme with PRA can not only achieve the synchronization, but also detect watermark correlating the amplitude, therefore, it is robust to cropping, pitch shifting and other attacks. A watermarking algorithm to embed watermark into low frequency MDCT coefficients during compression was proposed by Wang et al. (2004). In their scheme, the watermark is embedded into the MDCT-transformed permuted block with embedding locations chosen quite flexibly. A pseudorandom sequence is used as the watermark embedded by modifying the LSBs of AC coefficients. In the watermark extraction process, the stego audio is first transformed into the MDCT domain

and then the watermark can be obtained by the AC coefficients modulo 2. Embedding watermarks in the MDCT domain is very practical because MDCT is widely used in audio coding standards. The watermark embedding process can be easily integrated into the existing coding schemes without additional transforms or computation. Therefore, this method introduces the possibility of industrial realization in our everyday life. Petitcolas et al. (1999) exploited watermarking software named MP3Stego for MP3 audio. The software achieves watermark based on the parity of error length after audio quantizing and coding. But it cannot meet the real-time requirement and its capacity is too low. Digital watermark is embedded to the compressed domain of AAC audio in (Neubauer &Herre, 2000). In this scheme, watermark could be extracted with ancillary data calculated by psychoacoustic model. So it would increase computational complexity greatly. Qiao & Nahrstedt (1999) modified the scale factor of MPEG audio bitstream to embed watermark. However, the results for testing the robustness were not introduced in this scheme. Quan & Zhang (2006) employed wet paper codes and hide data directly in the MPEG audio bitstream by modifying the MPEG audio quantization process. The drawback is that the scheme could change the audio file size. However, as a recent developed China's standard, there is rarely report on watermarking schemes for AVS audio based on chaos in the literature.

Besides, there are some watermarking scheme based on audio content and HAS (Garcia, 1999). Boney et al. (1996) proposed an algorithm to make use of the MPEG psychoacoustic model in order to obtain frequency-masking values necessary to achieve prime imperceptibility, which generates watermarks by filtering a PN sequence with a filter that approximates the HAS frequency masking characteristics. An enhanced technique was developed by Swanson et al. (1998), in which they examine perceptual coding techniques in order to embed the watermark. Using only temporal masking or frequency masking will not be good enough for watermark inaudibility, thus both masks are used in order to achieve minimum perception distortion in the stego audio. Experimental results showed that this technique is both transparent and robust. The similarity values proved that the watermark can still survive under cropping, resampling and MP3 encoding. Therefore, this technique is an effective one under all the considerations for audio watermarking.

Cheng et al. (2002) proposed enhanced spread spectrum watermarking for compressed audio in MPEG-2 AAC format which embedded watermarks in the discrete cosine domain with the psychoacoustic model. A spectral envelope filter was introduced in the detection phase to reduce the noise variance in the correlation, thus improving the detection bit rate. They transformed the original host signal into a new host signal with smaller variance before adding the watermark and they used a heuristic estimation on perceptual weighting for embedding the watermark. All of these features result in both low structural and computational complexities. Seok et al. (2002) proposed an audio watermarking based on traditional direct sequence spread spectrum approach and achieved a watermark insertion rate of 8 bits per second. The inaudibility of the watermark was maintained by incorporating the psychoacoustic model derived from the MPEG I layer 1 audio coding standard. Their experiments showed the audio watermarking system to be robust to several signals transformations and malicious attacks.

In summary, the watermarking systems are designed to embed a hidden robust watermark into digital media. These systems have to satisfy two conflicting robustness and good fidelity requirements. To accomplish this task, variety of techniques has been exploited, and different domains are involved to enhance a certain application of watermarking and/or improve fidelity and robustness of watermarked signal. However, watermarking systems have a number of differences. These differences can be considered in evaluating the performance of watermarking systems and suitability of these systems for a specific application. Digital watermarking work mainly concentrated on the design of watermark algorithm, which is generally divided into watermark generation, embedding and detection in the spatial, frequency, cepstrum or mixed domain using symmetric and public key. Under this background, digital watermarking has received much attention recently and has been a focus in network information security. There is unprecedented development in the audio watermarking field. On the other hand, attacks against watermarking systems have become more sophisticated. In general, these attacks can be categorized into common signal processing techniques, such as AVS, MP3 and MPEG compression, low-pass filtering, noise addition, requantization, resampling and so on. Apart from security-oriented applications, which will continue to attract research interests, digital watermarking has been proved to be useful for broadcast monitoring and we believe that it can be useful for other non security-oriented applications such as error concealment and metadata hiding within multimedia content for legacy systems so that the metadata can survive format conversions. The latter is particularly useful for document identification as it allows us to re-associate medical images with patients' records and linking multimedia to the World Wide Web.

4. Psychoacoustic Modelling

Psychoacoustic modelling is important in audio coding and watermarking, which ensures that the changes of the original signal remain imperceptible. Compared with HVS, HAS is much more sensitive, which makes audio watermarking more challenging than image watermarking. HAS can detect signals with a range of frequency greater than 10^3 and with a power greater than 10^6. Understanding how HAS perceives sound is important for the development of a successful audio watermarking system. The main property of audio perception lies in the masking phenomena, which includes pre-masking and post-masking. Psychoacoustic modelling has made important contributions in the development of recent high quality audio compression methods and has enabled the introduction of effective audio watermarking techniques (Swanson et al., 1998; Robert& Picard, 2005). In audio analysis and coding, it strives to reduce the signal information rate in lossy signal compression, while maintaining transparent quality. This is achieved by accounting for auditory masking effects, which make possible to keep quantization and processing noises inaudible. In speech and audio watermarking, the inclusion of auditory masking has made possible the addition of information that is unrelated to the signal in a manner that keeps it imperceptible, transparent and can be effectively recovered during the identification process. Furthermore, no psychoacoustic model is available in the AVS compressed domain to enable the adjustment of the watermark to ensure inaudibility.

HAS can be modelled as a frequency analyzer consisting of a set of 25 bandpass filters (called critical bands) with logarithmically widening bandwidth for higher frequencies. The

critical band rate (CBR) specifies the correspondence between frequencies pooled for perception and locations on the basilar membrane. The mapping between CBR and frequency has been approximated as:

$$z = 13\arctan(0.76) + 3.5\arctan(f/7.5)^2 \quad (1)$$

where z is CBR in Bark and f is frequency in kHz. In psychoacoustics, the intensity of a sound is measured in terms of Sound Pressure Level (SPL). The HAS cannot perceive the sound if SPL is below a threshold. Such threshold is called absolute threshold of hearing(ATH), which determines the energy for a pure tone that can be detected by HAS in noiseless environment. ATH is a nonlinear function approximated by (2) and is illustrated in Fig. 2.

$$T(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 (dB) \quad (2)$$

where T is the SPL in dB and f is the frequency in Hz.

The basic idea underlying perceptual watermarking schemes is to incorporate the watermark into the perceptually insignificant region of an audio signal in order to ensure transparency. An important characteristics of HAS is auditory masking that has been applied in the area of perception-based compression audio coding and in the watermarking field too. If a sound has any frequency components with power levels below the ATH, then these components can be discarded without degrading the perceptual quality of the sound. The AVS audio algorithm compresses audio data in large part by removing the acoustically irrelevant parts of the audio signal. In fact, it takes advantage of the HAS inability to hear quantization noise under conditions of auditory masking. This masking occurs whenever the presence of a strong audio signal makes a temporal or frequency neighbourhood of weaker audio signals imperceptible. The psychoacoustic model analyzes the audio signal and computes the amount of noise masking available as a function of frequency. The encoder uses this information to decide how best to represent the input audio signal with its limited number of code bits. The psychoacoustic model defined in AVS audio algorithm exploits the frequency masking: let two simultaneously occurring signals be close together in frequency, the lower-power frequency components may be inaudible in the presence of the higher-power frequency components. Note that the frequency-masking model defined in AVS audio is to obtain the spectral characteristics of a watermark based on the inaudible information of the HAS. The perceptual model is based on the psychoacoustic phenomena of masking.

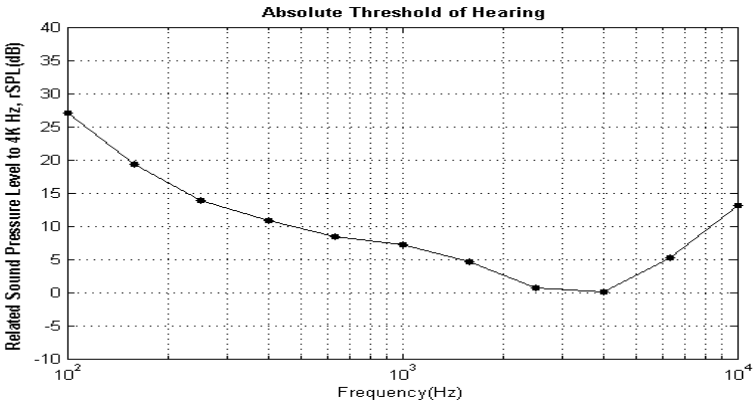


Fig. 2. Absolute threshold of hearing

The masking model estimates the amount of (noise-like) energy that can be added to the original audio signal without becoming perceptible. This energy varies over frequency. Therefore, the implementation of the masking model introduces frequency band partitions. The maximum energy allowed in a particular band is referred to as the masking threshold. ATH at any frequency is the minimum intensity of the sound at that frequency a normal human ear can perceive which is similar to overall threshold. A portion of an audio with 44.1 kHz sampling rate, which is expressed by the power spectrum, ATH, overall threshold and the audio power spectrum density (PSD) is illustrated in Fig.3. In the psychoacoustic model, the frequency masking threshold for each frequency component is computed from specific audio signal. Besides, modifications to the audio frequency components whose magnitudes are less than the masking threshold create no audible distortions to the audio piece by normalizing the added chaotic signal based on these threshold values.

Perceptual weighting in audio watermarking algorithm considers the human audio perception by means of adapting the introduced watermark energy to the current masking threshold. On the one hand, this can lead to audible distortion for the case that the amount of embedded watermark energy is too large. On the other hand, from a psycho-acoustical view, in some cases, more noise energy would be allowed. In other words, less watermark energy than possible is introduced and only suboptimal extraction performance is achieved. Psychoacoustic modelling strives to reduce the signal information rate in lossy signal compression while maintaining transparent quality. This is achieved by accounting for auditory masking effects, which makes it possible to keep quantization and processing noise inaudible. In speech and audio watermarking, the inclusion of auditory masking has made possible the addition of information that is unrelated to the signal in a manner that keeps it imperceptible. Perceptual weighting of the added chaotic sequence is performed with the frequency characteristic similar to the threshold in quiet curve of the HAS.

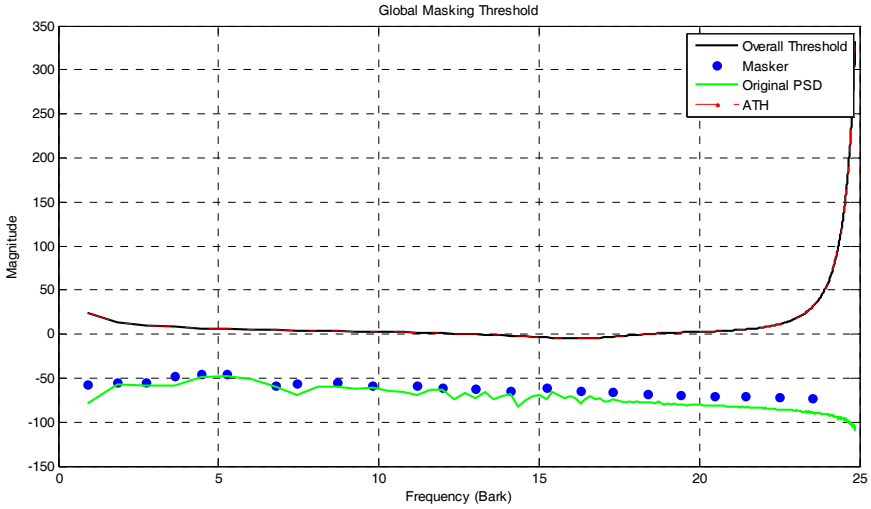


Fig. 3. ATH, overall threshold and audio PSD

5. Chaotic Watermarking Scheme for Audio Signal

5.1 Properties of Chaotic Function

Mathematically, chaos means deterministic behavior which is very sensitive to its initial conditions, in other words, infinitesimal perturbations of initial conditions for a chaotic dynamic system lead to large variations in behavior. A chaotic map exhibiting some sort of chaotic behavior is fully described by (3).

$$\{x_n : x_n = F(x_{n-1,} r)\} \tag{3}$$

where *r* is a function seed and *F()* is a nonlinear transformation that maps scalars to scalars condition. A discrete-time chaotic signal x_n can be generated from a chaotic system with a single state variable by applying the recursion:

$$x_n = F(x_{n-1}) = F^n(x_0) \tag{4}$$

where x_0 is the system initial condition. A chaotic sequence may be easily reproduced given the same initial conditions and initial value x_0. A slight change in the initial conditions of a chaotic function will lead to significant changes in the resultant mapping but their correlation may change slightly which can be seen in Fig. 4. Chaotic maps often occur in the study of dynamical systems and often generate fractals. A fractal may be constructed by an iterative procedure studied as sets rather than in terms of the map that generates them. This is often because there are several different iterative procedures to generate the same fractal. The simplest chaotic map is logistic map which is defined as:

$$x_{n+1} = rx_n(1 - x_n) \tag{5}$$

where r is the function seed and x_n is the current value of the mapping at time n with an initial value x_0. For the logistic map, there are two distinct regimes, namely, the periodic or bifurcation regime and the chaotic regime. When r=3, x_n oscillates between two values and never converges. As r increases, x_n goes through bifurcations and eventually becomes chaotic. When $r \approx 3.5699$, x_n becomes random. When $r > 3.5699$, the behavior is in chaotic state. A bifurcation diagram summarizes this in Fig. 5. The horizontal axis shows the values of the parameter r, while the vertical axis shows the possible long-term values of x.

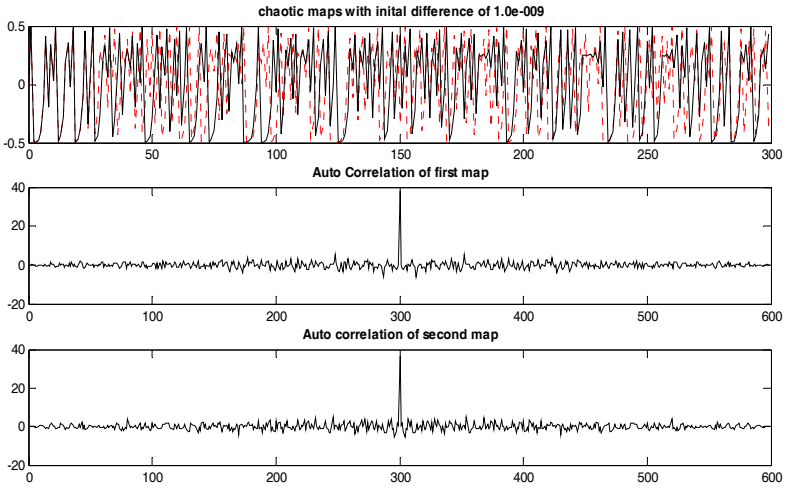


Fig. 4. The chaotic sequence with slight change and auto correlation

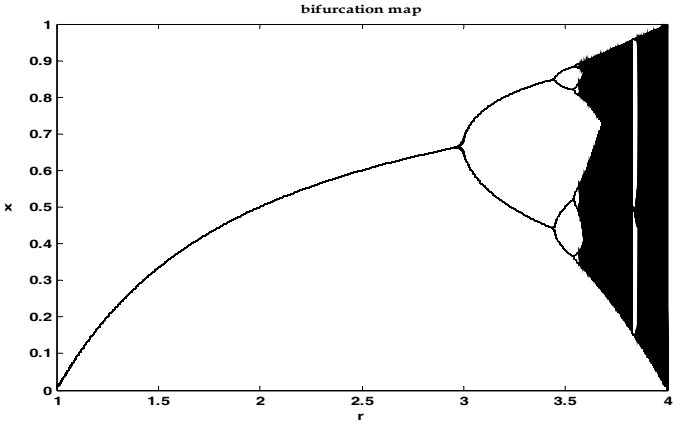


Fig. 5. Bifurcation diagram of logistic map

To date, a number of chaotic functions have been proposed in the literature for the purpose of watermark generation (Giovanardi et al., 2003; Tsekeridou et al.,2001), the most prominent and popular ones are the tent map, Bernoulli map, logistic map and quadratic map. Pseudo random sequences generated by chaotic dynamic system can be used to randomize coefficients in the watermarking field. A chaotic function can produce almost uncountable random sequences that have good auto and cross correlation characteristics and are extremely sensitive to the initial secret keys. From Fig. 6, we can see different time series of different chaotic sequences and their auto and cross correlation properties, where ACF denotes the auto correlation and CCF means cross correlation.

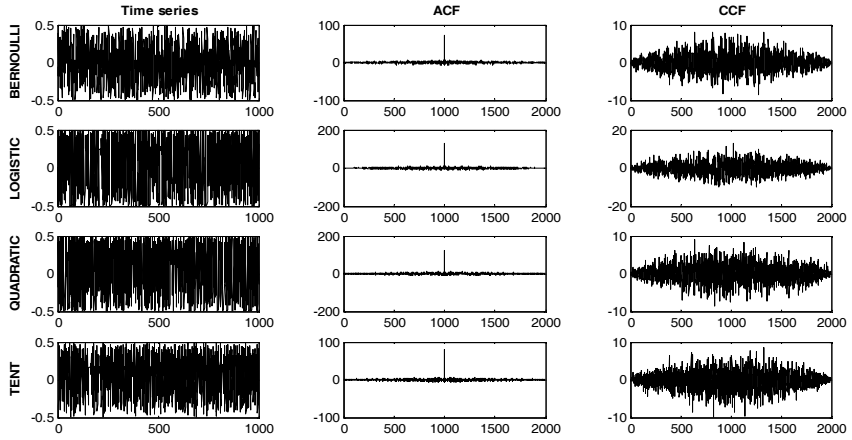


Fig. 6. Time series, ACF and CCF of different chaotic maps

The degree of the randomness of any map, *F(x)*, can be seen from the Lyapunov Exponent variation of the map with respect to the value of r. We can calculate the Lyapunov Exponent λ for the chaotic function as follows:

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \log \left| \frac{d}{dx} F^n(x) \right|_{x_0} \tag{6}$$

By applying the chain rule, the term reduces to

$$\lambda(x_0) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N} \log(|F'(x_i)|) \tag{7}$$

After N iterations, if Lyapunov Exponent is negative, then the orbits converge in time(periodic), and if Lyapunov Exponent is positive, then the distance between nearby orbits grows exponentially in time, and the system exhibits sensitive dependence on initial conditions(chaotic). The dynamics of Lyapunov exponents in terms of time values are displayed in Fig. 7.

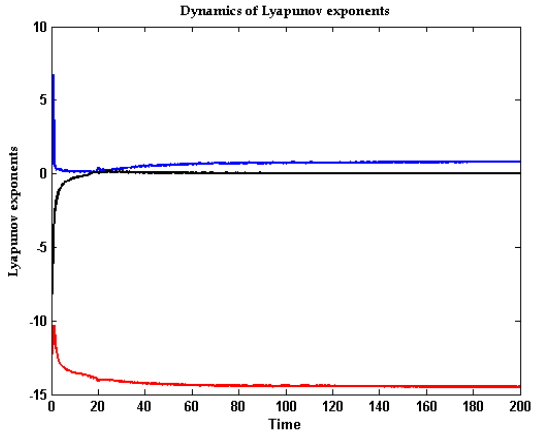


Fig. 7. Dynamics of Lyapunov exponents

5.2 Advantages of Chaotic Watermarks over Pseudorandom Watermarks

The chaotic map has been shown to produce lowpass watermarks and also white watermarks. These white watermarks are similar to those generated by a PN generator. Since there are irrational numbers close to every rational number, the map exhibits sensitive dependence on initial conditions. As iteration times increase, chaotic sequences are characterized by white spectrums. The control of spectral properties of the generated sequences of these chaotic functions offers a distinct advantage over the sequences generated by the pseudorandom generators. For example, if we know that the watermarked audio will be subjected to manipulations which are lowpass in general, we can generate a lowpass watermark which will be more robust to these attacks. By simply altering the function seed in these chaotic functions, one can generate watermarks with different spectral properties. In applications where no severe distortions are expected, e.g. in captioning or indexing applications, highpass spectrum watermarks can be used since they guarantee superior performance. Watermark signals generated by iterating of a chaotic function have an advantage over signals generated by coloring white noise in that these signals are much easier to create and re-create. Rather than have to seed a pseudorandom number generator and then apply a filter to the resultant signal to generate colored noise, a single seed can determine the properties of the generated sequence from the chaotic function. Highpass sequences are typically less robust to lowpass filtering and small geometric deformations of the image than lowpass sequences. Moreover, we can see the differences and advantages from their auto and cross correlation shown in Fig. 8 too. Chaotic have better auto and cross correlation characteristics than PN sequence. These characteristics make chaotic maps excellent candidates for watermarking and have been shown to have superior robustness than the widely used PN sequences in watermarking applications (Giovanardi et al., 2003; Tefas et al., 2003; Tsekeridou et al., 2001). Therefore, chaotic maps have recently been used for digital watermarking to increase the security.

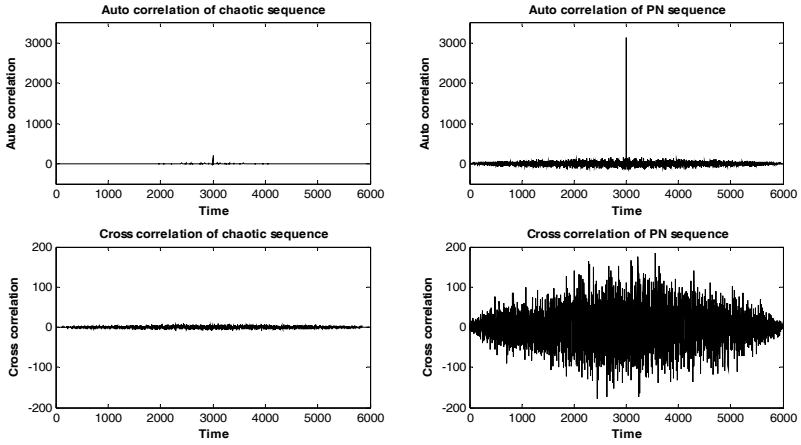


Fig. 8. Comparison of auto and cross correlation of chaotic sequence and PN sequence

Therefore, on the basis of the characteristics of the human auditory system (HAS) and the techniques of chaotic map, in next section, we present a perception-based audio watermarking algorithm for AVS audio files that works in the compressed domain, makes its manipulations in the frequency domain. Our algorithm overcomes the problem of the algorithms that operate in the uncompressed domain, which are vulnerable to compression/recompression attacks, as it embeds the chaotic signal as watermark in parts of the audio data that are masked or are not perceptible because of psycho-acoustic laws.

6. New Perceptual Watermarking Scheme for AVS Audio

6.1 Introduction

In this section, a new perceptual audio watermarking approach is developed for AVS audio based on the HAS and chaos. The copyright information is embedded into the IntMDCT (integer modified discrete cosine transform) coefficients in the AVS compressed audio data based on perceptual model of frequency masking. The proposed AVS-WM scheme is shown in Fig. 9.The IntMDCT is an integer approximation of MDCT with perfect reconstruction and has good characteristics of energy compaction. Chaotic sequences are adopted to improve the security of the proposed watermarking scheme. The low frequency components after IntMDCT transform are chaotically spread and encrypted to protect the audio copyright. The good property after various attacks demonstrates that the proposed audio watermarking scheme based on chaos is able to provide a feasible and effective copyright protection scheme for AVS audio signal efficiently.

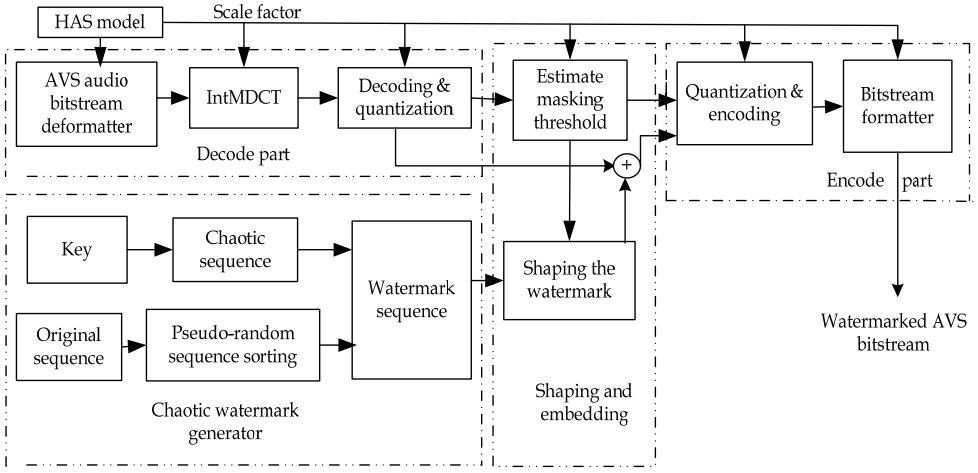


Fig. 9. The diagram of AVS-WM scheme

6.2 Chaotic Watermark Generation

Chaotic watermarks have been proposed as an alternative to the more commonly used pseudo random watermarks. The process of generating a watermark derived from a chaotic map involves several steps. A value for the function seed and an initial starting value must be first selected. The chaotic function is iterated n times. To increase the watermark security, we introduced a chaotic sequence generated by chaotic dynamic system to randomize the coefficients. A chaotic function can produce almost uncountable random sequences that have good autocorrelation characteristics and are extremely sensitive to the initial secret keys. The copyright holder can use the arbitrary real numbers between 0 and 1 as the initial secret keys in order to get satisfactory auditory quality in embedding. Without correct key, knowing the algorithm itself is far from enough to extract watermarks. The sequence x_n generated by the chaotic map is composed of real numbers, so the output sequence $c(n)$ is quantized into binary stream by the perceptual auditory masking threshold T:

$$c(n) = \begin{cases} 1, & x_n \geq T \\ 0, & x_n < T \end{cases} \tag{8}$$

where T is chosen as the perceptual masking threshold in assigning the binary values. Then we use Exclusive OR operation to generate the watermark $w(n)$:

$$w(n) = \sum_{n=1}^{N_w} b(n) \oplus c(n) \tag{9}$$

where N_w is the size of the watermark, $b(n)$ is the copyright information, $\oplus$ denotes the Exclusive-OR operation. Use the chaotic maps to generate chaotic digital watermark signal

with zero mean and one variance. The embedded watermark sequence $I'(n)$ can be obtained as follows.

$$I'(n) = I(n) + a(n)w_k(n) \tag{10}$$

where $I(n)$is the original audio sequence and a is strength parameter that can be controlled by characteristics of HAS or some variables of audio signal such as the average power of amplitude of signal for ensuring inaudibility and robustness of watermarked signal. By varying a, the embedded watermark intensity can be modulated and hence the hidden effects can be adjusted. An appropriate a is chosen to enhance the robustness of the watermark and renders the watermark imperceptible and yet with good watermark quality.

6.3 Watermark Embedding

This section proposes an enhanced watermarking scheme that not only embeds the watermark below masking threshold area, but also takes advantage the HAS properties and insert the watermark into the audible spectrum areas and still keep the introduced distortion imperceptible. The details of the watermark embedding process are as follows:

Step 1. The input original audio is segmented into overlapped frames with N samples long. Each frame is decomposed into 25 subbands.

Step 2. Compute the power spectrum of the audio segment. Each segment of the signal $s(n)$ is weighted with a Hanning window. The maximum is normalized to a reference sound pressure level of 96 dB.

Step 3. The special and noticeable audio blocks to the embedders with larger energy are selected as salient features blocks. Salient features where the audio signal energy is climbing fast to a peak value are robust against attacks. Psychoacoustic model is applied to determine the masking thresholds for each subband. High threshold value is suitable for audio with sharp energy variation. Such selection may generate audible high frequency noise. Careful shaping can reduce the noise to a hardly audible level.

Step 4. Calculation of the individual masking thresholds. An embedding frequency point in the selectable frequency range is chosen based on a chaotic secrete key. The interleaved data is spread by the chaotic sequence.

Step 5. After segmenting and choosing the frequency points, we quantize the frequency coefficients to embed the watermark. The digital watermark in the watermark generation section can be embedded into audio signal by quantizing the selected transformation coefficients.

Step 6. For each selected audio block, IntMDCT transform is performed. It is common to utilize the HAS masking effects for keeping good inaudibility and robustness. The watermark is only embedded into salient features of audio blocks which have a high energy value in our scheme. The data to be embedded (hidden data) is spread by chaotic sequence and interleaved to enhance watermarking robustness.

Step 7. The available coefficients are used to form the audio blocks by inverse IntMDCT. Finally, the watermarked audio blocks combine with the unselected audio blocks to form the watermarked audio signal.

6.4 Watermark Extraction

The extraction process should be implemented before reconstructing the IntMDCT coefficients because the information is embedded in the quantized coefficient. The incoming audio is first segmented into overlapped frames. The block diagram of extracting hidden watermark is the reverse processing of watermark embedding. The watermark detection procedure, the extraction rule and the detailed step are as follows:

Step 1. Perform segmentation of the watermarked signal. The frame is decomposed into 25 subbands.

Step 2. Find the first frame that contains watermark. According to the embedding algorithm, the first bit of watermark is embedded in the first non-zero frame. Therefore, the first non-zero frame should be the first detection candidate.

Step 3. Merge two neighboring frames to one frame. Apply Gaussian distribution analysis on all the macro frame for watermark detection. Watermark is extracted based on Gaussian distribution analysis function on watermarked frame.

Step 4. Carry out the inverse transformation on segmented watermarked signal by secret key to get the coefficients

Step 5. The same HAS model is applied on the data in frequency domain to determine the masking thresholds.

Step 6. The appropriate data are de-spread and de-interleaved in order to detect and recover any hidden data. Extracted watermark by quantization rule in the chosen frequency point. Calculate the cross-correlation coefficients ρ and τ and compare ρ and τ to get watermark $\hat{w}$

$$\rho = \sum_{i=1}^{N_w} w(i) * \hat{w}(i)) \Bigg/ \left[\sum_{i=1}^{N_w} | w(i) |^2 * \sum_{i=1}^{N_w} | \hat{w}(i) |^2 \right]^{1/2} \tag{11}$$

$$\tau = \sum_{i=1}^{N_w} \hat{w}(i) * \sqrt{N_w} \Bigg/ \left[\frac{1}{N_w} \sum_{i=1}^{N_w} (\hat{w}(i) - \hat{\mu}_{\hat{w}})^2 \right]^{1/2} \tag{12}$$

$$\begin{cases} \hat{w} = 1, & if\ \rho \geq \tau \\ \hat{w} = 0, & if\ \rho < \tau \end{cases} \tag{13}$$

6.5 Experimental Results and Performance Analysis

As to test files, in this experiment, the 19 well known SQAM files are selected. The files have a sampling frequency of 44.1 kHz and are 16 bit quantized. In our experiments, the files are reduced to a mono signal, For ease of expression hereafter, the host audio signals are marked with a number in ascending order, i.e. 1) Music: 1.Bach, 2.Pop, 3.Rock, 4. Jazz; 2) Percussive instruments: 5.Hihat, 6.Castanets, 7.Glockenspiel1, 8.Glockenspiel2; 3) Tonal instruments: 9. Harpsichord, 10.Violoncello, 11.Horn, 12.Pipes, 13.Trumpt, 14.Electronic tune; 4) Vocal: 15.Sopranor, 16.Bass, 17. Quartet; 5) Speech: 18.Female speech, 19.Male speech.

6.5.1Time Domain Waves and Spectrum

Tests were run to evaluate whether the watermarked data in an AVS audio can be detected through more qualitative methods, and the characteristics of the changes. This study only analyzed the sound waveforms in the time and spectrums in the frequency domains. The original AVS audio waveforms are compared with the watermarked audio containing the hidden data. The resulting waveforms in the time domain and spectrums in the frequency domain are shown in Fig.10 and 11 respectively. Both analyses do not show any distinguishing differences between the original audio and the watermarked audio. However, there seems to be a slight amount of signal loss and a slight increase in distortion due to the hidden data, but the overall distortion is acceptable. Besides, from the time waveform and spectrum of the watermarked and unwatermarked signal and their residue difference, it is obvious that the scheme is imperceptible and inaudible.

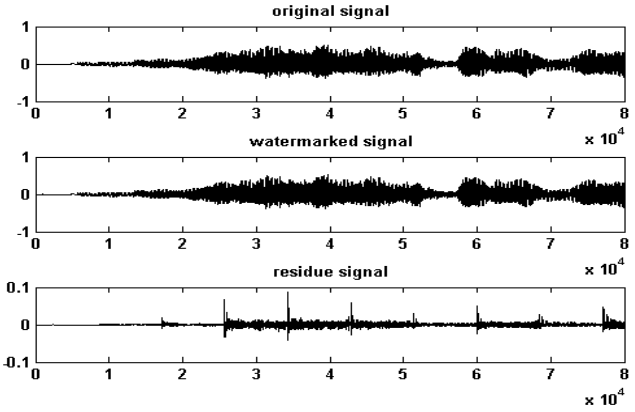


Fig. 10. Original, watermarked and residue audio waveform in time domain

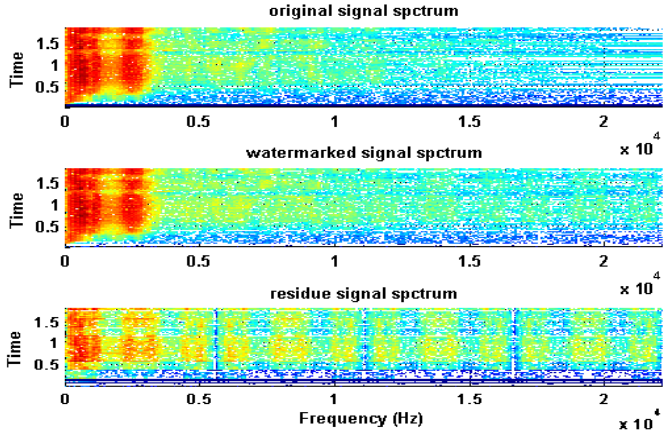


Fig.11. Spectrum of original, watermarked and residue signal

6.5.2 Robustness test

In order to illustrate the robustness nature of our watermarking scheme, in our experiment, common audio signal processing include re-quantization, re-sampling, additive noise, low-pass filtering, echo addition, equalization, mp3 compression, pitch shifting, time-scale modification and jittering are used to estimate the robustness of our scheme. Table 1 summarizes the watermark detection results against various common signal processing attacks. Watermark detection results after the attacks described above are shown in Table 1. It is evident that correlation values after attacks are very high which indicate the robustness of the audio watermarking scheme. The attacks are described in detail as follows:

- *Additive noise:* White noise with 10% of the power of the audio signal is added.
- *Amplitude variation*: The watermarked signal is attenuated up to 120% and down to120%.
- *AVS compression*: The coding/decoding is performed using a software implementation of the AVS Audio coder with several different bit rates (32kbps, 48 kbps, 64 kbps and 96 kbps).
- *Echo addition:* An echo signal with a delay of 50 ms and a decay of 50% is added to the original audio signal.
- *Expanding*: Expand the watermarked signal with increment of 3 dB and -3dB respectively.
- *Jittering*: Jittering is a random cropping performed evenly.
- *Low-pass filtering*: The low-pass filter with the 4 kHz cutoff frequency is applied to watermarked audio signal.
- *Pitch shifting:* Tempo-preserved pitch shifting is a difficult attack for audio watermarking algorithms as it causes frequency fluctuation. In this experiment, the pitch is shifted 1 degree higher and 1 degree lower.
- *Random cropping*: 10% samples are cropped at each of three randomly selected positions (front, middle and back).
- *Re-quantization*: A 16-bit watermarked audio signal is quantized 8-bit and back to 16-bit.
- *Re-sampling*: The original audio signal is sampled with a sampling rate of 44.1 kHz. Watermarked audio signal is down-sampled to 11.025 kHz, 22.05 kHz, and then up-sampled back to 44.1 kHz; up-sampled to 88.2 kHz, and then down-sampled back to 44.1 kHz.
- *Reverse amplitude*: Reverse the plus or minus of amplitude of samples.
- *Smoothness filtering*: Smoothly filter the watermarked audio signal.
- *Time-scale modification*: The watermarked audio signal is lengthened by 4% while preserving the pitch.

Attacks	Corr.value	Attacks	Corr.value
Without attack	1	Jittering	0.95
Additive noise	0.96	Low-pass filtering	0.93
Amplitude variation	0.86	Pitch shifting	0.92
AVS (96kbps)	0.96	Random cropping	0.82
AVS (48kbps)	0.93	Re-sampling 22.05kHz	0.83
AVS(64kbps)	0.89	Re-sampling 11.025kHz	0.79
AVS(32kbps)	0.87	Re-sampling 88.2kHz	0.73
Echo addition	0.83	Smoothness filtering	0.94
Expanding	0.88	Time-scale modification	0.86

Table 1. Robustness test results

6.5.3 StirMark Benchmark for Audio (SMBA) Test

SMBA is a standard and common robustness evaluation tool for examining audio watermarking technique(Lan et al.,2005). In this experiment, different attacks are applied to the watermarked and un-watermarked test set by using the Stirmark software. The parameters of the benchmark software were the default parameters included in the version of the tool available on the web. In addition, only the left channel has been marked in the experiments, thus stereo attacks do not apply here either. The attacks considered for this test are summarized in table 2. In this table, a total of thirty six different attacks are performed. From the results of the SMBA test, the high correlation values of after attacks denote that the watermarked can be extracted and this scheme is robust and secure.

Attacks	Corr. value	Attacks	Corr. value	Attacks	Corr. value
AddBrumm	0.98	Normalizer1	0.87	FFT Invert	0.88
AddNoise	0.94	Normalizer2	0.84	CopySample	0.49
AddSinus	0.76	Compressor	0.95	FlippSample	0.51
AddFFTNoise	0.58	BassBoost	0.92	CutSample	0.54
NoiseMax	0.77	RC-HighPass	0.93	ZeroCross	0.67
Denoise	0.92	RC-LowPass	0.90	ZeroLength1	0.71
LSBZero	0.85	FFT HLPass	0.71	ZeroLength2	0.69
Echo	0.89	Stat1	0.89	ZeroRemove	0.85
Exchange	0.54	Stat2	0.85	PitchScale	0.70
Resampling	0.52	FFTStat1	0.82	DynamicPitchScale	0.68
ExtraStereo	0.95	Smooth1	0.67	TimeStretch	0.62
VoiceRemove	0.57	Smooth2	0.76	DynamicTimeStretch	0.59

Table 2. Correlation values for SMBA test

6.5.4 Subjective Listening Tests

In order to evaluate the audibility of the watermarked audio, we have selected the "Double blind, triple stimulus, with hidden reference" test methodology according to ITU-BS.1116 listening test(ITU,1993). In this test, all clips and trials are randomized. This test method is

used for evaluating systems that cause only small degradations in audio quality. Ten listeners both experienced and familiar with the set of critical audio items participated in the test. The SQAM test items contain excerpts of single and multiple instruments, speech and complex sound sources. This test set was also extensively used for assessment of the subjective audio quality in the AVS audio development process. Fig. 12 presents the results from the ITU-BS.1116 listening test. It can be seen that the quality degradation of the bit stream watermarking system is very small for the vast majority of the test items. For all items the confidence intervals of both signals overlap which indicates that there is no significant distortion introduced by this scheme. The test results indicate that there is no statistical difference between the audio quality of the watermarked items and the original audio quality.

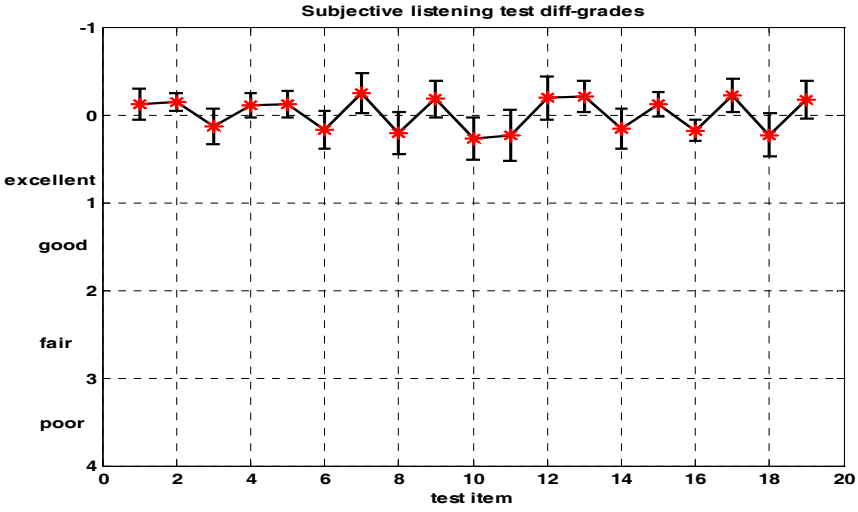


Fig. 12. Subjective listening test results of AVS audio watermarking scheme

From the above test results, we can come to some performance analyses:

1) Comparative robustness against common signal processing operations: the frequency domain selectable frequency range is chosen as the tradeoff of inaudible and robustness against normal operations. The embedded frequency points are all from this range in the AVS audio watermarking scheme. Consequently, this watermarking algorithm has the comparative robustness against common signal processing operations;
2) High robustness against malicious attack: the selectable frequency range brings the possibility to be hidden; and the chosen embedding point based on chaos secret key determines hidden and random embedding position. The two measures have guaranteed the privacy of the embedding position, which has higher robustness against malicious attack algorithm than fixed embedding positions. Therefore, the intruder without secret key must be difficult to implement malicious attack on the audio watermarking scheme.
3) High systematic security: chaotic systems have high security due to extreme sensitivity to initial value. In the proposed method, chaotic map and HAS model is used to choose

embedding position, the initial value of map is considered as systematic secret key. Therefore, the security of whole system only relies on secret key, which makes it have higher systematic security.

7. Conclusion

In this chapter, a literature review on audio watermarking techniques is first given. Different issues related to audio watermarking are described. A robust audio watermarking scheme with high robustness against malicious attack is then presented for AVS audio. Salient features are adopted to select embedding points for random and hidden embedding position, then the selected data are spread by chaotic sequence to increase robustness against watermark attacks during the embedding process. Besides, the masking characteristics of psychoacoustic model are used to avoid introducing audible noise in the AVS compression. Consequently, the proposed method has higher robustness against malicious attack. The performance analyses and simulation results show that the proposed scheme not only guarantees robustness against common operations, but also keeps the watermark imperceptible and inaudible.

References

AVS Audio Expert Group (2005). Information Technology —Advanced Audio Video Coding Standard Part 3: Audio, Audio Video Coding Standard Group of China (AVS).

Bassia, P., Pitas, I., & Nikolaidis, N. (2001). Robust audio watermarking in the time domain. *IEEE Transactions on Multimedia*, vol. 3, pp. 232-241.

Bender, W., Gruhl, D., Morimonto, N. & Lu, A. (1996). Techniques for data hiding, *IBM System Journal*, vol. 35, no. 3&4, pp. 313-336.

Chen, B., & Wornell, G. W. (2001). Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, vol. 47, pp. 1423-1443.

Chen, B., Zhao, J., & Wang, D. (2008). An Adaptive Watermarking Algorithm for MP3 Compressed Audio Signals, *Proceedings of IEEE Conference on Instrumentation and Measurement Technology*, pp. 1057-1060.

Chen, O. T. C., & Wen-Chih, W. (2008). Highly Robust, Secure, and Perceptual-Quality Echo Hiding Scheme. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 629-638.

Chen, X.-S., Yang, Y.-T., Zhang, H., & Lu, X.-J. (2008). An audio blind watermark algorithm in wavelet domain based on chaotic encryption, *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition*, pp. 470-473.

Cheng, S., Yu, H., & Xiong, Z. (2002). Enhanced spread spectrum watermarking of MPEG-2 AAC audio, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. IV/3728-IV/3731.

Chung, T.-Y., Hong, M.-S., Oh, Y.-N., Shin, D.-H., & Park, S.-H. (1998). Digital watermarking for copyright protection of MPEG2 compressed video, *IEEE Transactions on Consumer Electronics*, vol. 44, pp. 895-901.

Daniel Gruhl, Anthony Lu, & Bender, W. (1996). Echo hiding, *Lecture Notes in Computer Science*, vol. 1174, pp. 295-315.

Garcia, R. A. (1999). Digital watermarking of audio signals using a psychoacoustic auditory model and spread spectrum theory, *Audio Engineering Society*.

Giovanardi, A., Mazzini, G., & Tomassetti, M. (2003). Chaos based audio watermarking with MPEG psychoacoustic model I, *Proceedings of the 2003 Joint Conference of the Fourth Pacific Rim Conference on Multimedia & the Fourth International Conference on Information, Communications and Signal Processing*, vol. 1603, pp. 1609-1613.

Gruhl, D., Lu, A., & Bender, W. (1996). Echo hiding, *Information Hiding, ser. Spring Lecture Notes in Computer Science,* vol. 1174, pp. 295–315.

He, X. (2008). *Watermarking in Audio: Key Techniques and Technologies*, Cambria Press.

ITU-R. Recommendation BS.1116 (1993). Methods for the Subjective Assessment of Small Impairments in Audio Systems including Multichannel Sound Systems, *ITU Technical report*.

Kirovski, D., & Malvar, H. S. (2003). Spread-spectrum watermarking of audio signals, *IEEE Transactions on Signal Processing*, vol.51, pp.1020-1033.

Lan, J., Huang, T., & Qu, J. (2007). A perception-based scalable encryption model for AVS audio, *Proceedings of the IEEE International Conference on Multimedia and Expo,* pp. 1778-1781.

Lang, A., Dittmann, J., Spring, R., & Vielhauer, C. (2005). Audio watermark attacks: from single to profile attacks. *Proceedings of the 7th workshop on Multimedia and Security*.

Li, Z., Sun, Q., & Lian, Y. (2006). Design and Analysis of a Scalable Watermarking Scheme for the Scalable Audio Coder, *IEEE Transactions on Signal Processing*, vol. 54, pp. 3064-3077.

Nedeljko, C., & Seppanen, T. (2008). *Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks*. Information Science Reference.

Neubauer, C., & Herre, J. (2000). Audio watermarking of MPEG-2 AAC bitstream, *Proceedings of 108th AES Convention*.

Petitcolas, F. A. P., Anderson, R. J., & Kuhn, M. G. (1999). Information hiding-a survey, *Proceedings of the IEEE,* vol. 87, pp. 1062-1078.

Podilchuk, C. I., & Delp, E. J. (2001). Digital watermarking: algorithm and application. *IEEE Signal Processing Magazine*, vol. 18, pp. 33-46.

Qiao, L., & Nahrstedt, K. (1999). Non-invertible watermarking methods for MPEG encoded audio, *Proceedings of the International Society for Optical Engineering,* pp. 194-202.

Quan, X., & Zhang, H. (2006). Data Hiding in MPEG Compressed Audio Using Wet Paper Codes, *Proceedings of 18th International Conference on Pattern Recognition,* pp. 727-730.

Robert, A., & Picard, J. (2005). On the use of masking models for image and audio watermarking, *IEEE Transactions on Multimedia*, vol. 7, pp. 727-739.

Seob, K.B., Nishimura, R., & Suzuki, Y. (2005). Time-spread echo method for digital audio watermarking, *IEEE Trans. Multimedia*, vol. 7, no. 2 pp. 212-221.

Seok, J., Hong, J., & Kim, J. (2002). A Novel Audio Watermarking Algorithm for Copyright Protection of DigitalAudio, *ETRI Journal*, vol.24.

Song, X., Su, Y., Hu, J., & Ji, Z. (2008). Information hiding in AVS compressed stream. *Proceedings of International Conference on Audio, Language and Image Processing*, pp. 988-992.

Swanson, M. D., Zhu, B., Tewfik, A. H., & Boney, L. (1998). Robust audio watermarking using perceptual masking, *Signal Processing*, vol. 66, pp. 337-355.

Tachibana, R., Shimizu, S., Kobayashi, S., & Nakamura, T. (2002). An audio watermarking method using a two-dimensional pseudo-random array, *Signal Processing*, vol. 82, pp. 1455-1469.

Tefas, A., Nikolaidis, A., Nikolaidis, N., Solachidis, V., Tsekeridou, S., & Pitas, I. (2003). Performance analysis of correlation-based watermarking schemes employing Markov chaotic sequences, *IEEE Transactions on Signal Processing*, vol. 51, 1979-1994.

Tsekeridou, S., Solachidis, V., Nikolaidis, N., Nikolaidis, A., Tefas, A., & Pitas, I. (2001). Bernoulli shift generated chaotic watermarks: Theoretic investigation, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1989-1992.

Wang, C. T., Chen, T. S., & Chao, W. H. (2004). A new audio watermarking based on modified discrete cosine transform of MPEG/audio layer III, *Proceedings of IEEE International Conference on Networking, Sensing and Control*, pp. 984-989.

Wang, Y. (1998). A new watermarking method of digital audio content for copyright protection, *Proceedings of Fourth International Conference on Signal Processing*, pp. 1420-1423.

Wang, X.-Y., & Zhao, H. (2006). A novel synchronization invariant audio watermarking scheme based on DWT and DCT, *IEEE Transactions on Signal Processing*, vol. 54, pp. 4835-4840.

Xiang, S., Kim, H. J., & Huang, J. (2008). Audio watermarking robust against time-scale modification and MP3 compression, *Signal Processing*, vol. 88, pp. 2372-2387.

Zmudzinski, S., & Steinebach, M. (2009). Perception-based authentication watermarking for digital audio data, *Proceedings of the International Society for Optical Engineering*.

18

Klaman Assisted LMS Methods

Nastooh Avesta
University of Ottawa
Canada

1. Introduction

The Least Mean Square (LMS) algorithm is a widely used adaptive method for identification, predication and noise suppression of Finite Impulse Response (FIR) filters (Haykin, 2002). Furthermore, it is extensively employed in various components of multimedia systems. Multimedia specific applications of LMS include, but certainly are not limited to, traffic routing (Xinyu et al, 2004), decision feedback channel equalization (Hoshino et al, 2000), signal classification (Bauckhage, 2005), echo cancelation (Sristi et al, 2001) and shape reconstruction (Mal-Rey, 2001). Main reasons for the popularity of LMS are its conceptual simplicity and ease of implementation. Although, these features are advantageous in both theoretical and practical developments; however, they tend to obscure the fine details of the systems under consideration. To address the shortcomings of a typical LMS algorithm, combined Klaman filter and LMS methods are proposed.

Klaman filter provides the optimal, in bayesian sense, state estimate of a linear system, based on degraded and noisy observations. It does so by taking into account the prescribed system model and simultaneously utilizing the difference between the predicated state and the observed one. The former, i.e., prescribed system model, is what LMS is incapable of taking into account, i.e., there are no means by which LMS can take into account partial observations or incomplete state measurements. Thus, under such circumstance, which are all too prevalent under any practical situation, LMS performance degrades rapidly. Although, recently, there have been other attempts to merge Kalman filtering and LMS algorithm (Lopez et al, 2007) and (Kohli & Mehra, 2007); however, these solutions relay on explicit assumptions on the observed system model. Methods described in this chapter are fundamentally different, as none make any implicit or explicit, assumptions on the observed system model. Here, we present full and partial Kalman filter based LMS update algorithms, where the traditional LMS algorithm is modeled as a noisy observation plant of a Kalman filter, with Kalman filter states representing the update coefficients. Nothing that multimedia applications often necessitate the use of long adaptive filters, requiring excessive computations to calculate the filter output and to update all coefficients, many researchers have addressed the impact of reducing the necessary computations through using shorter filters or updating only a subset of coefficients at every iteration. These partial update algorithms include periodic LMS update (Douglas, 1997), where every L-th coefficient is updated and the M-MAX algorithm (Mayyas & Aboulnasr, 2004), where the M coefficients with the most potential to reduce error are updated in a given iteration.

However, these partial update algorithms typically use the same update term that would have been used by the full update algorithm. Furthermore, no post processing or additional real-time processing is included to improve the overall estimation. Given the limited nature of partial updates, in this paper we propose an improved estimation of the full length filter, based on partial updates and observations.
The outline of this chapter is as follows. In Section 2, a brief description of gradient based adaptive methods, highlighting a number of performance criteria and stability requirements, is given. Section 3, provides an overview of Kalman filtering and its respective performance criterion. Klaman assisted methods for full and partial weight update are developed in Section 4 and Section 5. Finally, Section 6 provides a summary and an overview of the current and future line of research.

2. Gradient Based Adaptation

In this section, we consider two gradient based methods for adaptive filtering: Steepest Descent (SD) and Least Mean Square (LMS). We note that although SD method does not lend itself to practical implementation; however, it is capable of reaching the theoretical optimum solution. On the other hand, we also note that although LMS, readily, lends itself to practical implementation; however, its final solution may be far from the optimal solution. Developments of this section closely follow (Haykin, 2002).

2.1 Steepest Descent Method

At time instance n, let the output of an N tap adaptive filter, in response to input vector $\boldsymbol{x}(n)$, be $y(n)$, (1). In (1), the i^{th} element of $\boldsymbol{x}(n)$, Nx1, is the tap delayed input $x(n-i)$ and $\boldsymbol{w}(n)$, 1XN, represents the gain of each tap at instance n.

$$y(n) = \boldsymbol{w}(n)\boldsymbol{x}(n) \tag{1}$$

Let the desired response, at time n, be indicated by $d(n)$ and its difference with $y(n)$ by $e(n)$.

$$e(n) = d(n) - y(n) \tag{2}$$

Our aim is to minimize the Mean Square Error (MSE), (3), by judicial adjustment of the weight vector $\boldsymbol{w}(n)$, where, in (3), E is the expectation operator.

$$J(n) = E\left[e^2(n)\right] \tag{3}$$

Given that $J(n)$ is a quadratic function of $\boldsymbol{w}(n)$, i.e., $e(n)$ is a linear function of $\boldsymbol{w}(n)$ and elements of $\boldsymbol{w}(n)$ are assumed independent, a simple update policy is to move in the opposite direction of gradient vector of (3). Expanding (3) and differentiating it with respect to $\boldsymbol{w}(n)$, (4), the gradient vector is obtained, where $\boldsymbol{r}_{xd}$ is the cross correlation function of $\boldsymbol{x}(n)$ and $d(n)$; and $\boldsymbol{R}_{xx}$ the auto-correlation matrix of $\boldsymbol{x}(n)$.

$$\nabla J(n) = -2\boldsymbol{r}_{xd} + 2\boldsymbol{R}_{xx}\boldsymbol{w}(n) \tag{4}$$

The update policy may now be stated as (5), that is, the optimal solution is obtained by moving in the opposite direction of gradient vector, in step sizes equal to μ.

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\left[\boldsymbol{r}_{xd} - \boldsymbol{R}_{xx}\boldsymbol{w}(n)\right] \tag{5}$$

It can be shown, e.g. see (Haykin, 2002), that for $\boldsymbol{w}(n)$ to converge to optimum solution as $n \to \infty$, μ must satisfy (6), where λ_{max} is the largest eigenvalue of $\boldsymbol{R}_{xx}$.

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{6}$$

It is noted that for the SD method to reach to the optimum solution, both $\boldsymbol{R}_{xx}$ and $\boldsymbol{r}_{xd}$ must be known a priori.

2.2 Least Mean Square Algorithm

In this section, we provide an overview of the LMS algorithm, along with its performance measure characteristics. Let the noise induced observation, at time n, be indicated by $d(n)$, where the observation noise is characterized as a zero mean white Gaussian noise with variance σ_o^2, $o \sim N\left(0, \sigma_o^2\right)$.

Adjusting $\boldsymbol{w}$ to minimize the instantaneous squared error, $e^2(n)$, we proceed by minimizing $e^2(n)$ as a function of $\boldsymbol{w}$.

$$\begin{aligned} &\frac{de^2(n)}{dw_i} = 0 \quad i = 0 \cdots N-1 \\ &-2e\frac{dy(n)}{dw_i} = 0 \end{aligned} \tag{7}$$

Noting that $y(n) = \sum_{k=0}^{N-1} w_k x(n-k)$ and hence $\frac{dy(n)}{dw_i} = x(n-i)$, the update equation (8) is obtained. That is, $\boldsymbol{w}$ is adjusted by moving in the opposite direction of instantaneous error surface gradient, in step sizes equal to μ.

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu e(n)\boldsymbol{x}(n) \tag{8}$$

For LMS algorithm to converge to its final solution, μ must satisfy the bounds of (9), where σ_x^2 is the variance of the input signal.

$$0 < \mu < \frac{2}{N\sigma_x^2} \tag{9}$$

The performance of LMS is greatly influenced by its design parameter μ, for example, the minimum achievable MSE is directly proportional to μ.

$$J(\infty) = \sigma_o^2\left(1 + \frac{\mu}{2} N\sigma_x^2\right) \tag{10}$$

The performance criteria of LMS is measured based on Mean Square Error (MSE), as defined in (3), and Mean Square Deviation (MSD), as defined in (11), where $\boldsymbol{\varepsilon} = \boldsymbol{w}^{\boldsymbol{Optimum}} - \boldsymbol{w}$, $\boldsymbol{w}^{\boldsymbol{Optimum}}$ is the optimum weight vector and $\|.\|^2$ is the Euclidean norm.

$$D(n) = E\left[\|\boldsymbol{\varepsilon}(n)\|^2\right] \tag{11}$$

Furthermore, to characterize the transient behavior and convergence of the LMS, we identify the following parameters: mean square excess, misadjustment, and average time constant. Both mean square excess, (12), and misadjustment, (13), are measures of LMS optimality.

$$J_{ex}(n) = J(n) - \sigma_o^2 \tag{12}$$

$$M = J_{ex}(\infty)/\sigma_o^2 \tag{13}$$

Average time constant is a measure of convergence time of LMS. Average time constant can be estimated as in, (14), where λ_{av} is the average value of eigenvalues of $\boldsymbol{R}_{xx}$.

$$\tau_{mse,\,av} = \frac{1}{2\mu\lambda_{av}} \tag{14}$$

Noting that misadjustment may also be expressed as, (15), it becomes obvious that in dealing with traditional LMS a choice has to be made with respect to convergence time verses the final error, i.e., for faster convergence, larger μ, a larger error will be incurred and vice versa.

$$M = \frac{\mu N\lambda_{av}}{2} \tag{15}$$

Under system identification setup, **Fig. 1**, our main interest is to minimize MSD, irrespective of the observation noise.

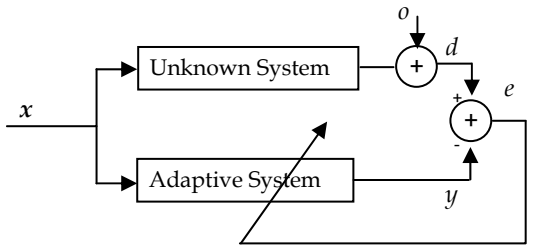


Fig. 1. System Identification Setup

3. Kalman Filter

Let the state evolution of a system be described by (16), where $\boldsymbol{x}_k$, $\boldsymbol{u}_k$, $\boldsymbol{\omega}_k$,all Nx1, are state of the system, the deterministic control and the observation noise; $\boldsymbol{A}_k$, $\boldsymbol{B}_k$, and $\boldsymbol{C}_k$, all NxN, are drift, diffusion and colouring matrices, respectively, at instance *k*,.

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k + \boldsymbol{C}_k \boldsymbol{\omega}_k \tag{16}$$

Given the observation plant (17), where z_k , and $\boldsymbol{v}_k$, both Mx1, are the observation vector and observation noise; $\boldsymbol{D}_k$, MxN, degradation matrix, respectively, then it could be shown, (Lewis, 1986), that an optimal linear estimate of $\boldsymbol{x}_k$ can be obtained through Kalman filtering.

$$z_k = \boldsymbol{D}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{17}$$

In particular, for $\boldsymbol{x}_0 \sim N\left(E\left[\boldsymbol{x}_0\right], \boldsymbol{P}_{\boldsymbol{x}_0}\right)$, $\boldsymbol{\omega}_k \sim N\left(0, \boldsymbol{Q}_k\right)$, and $\boldsymbol{v}_k \sim N\left(0, \boldsymbol{R}_k\right)$, i.e., all Gaussian statistics, this estimate is unconditionally optimal. The Kalman algorithm is described in (18), where P_k ,NxN, is the posteriori error covariance, P_k^- , NxN, is the a priori error covariance, I ,NxN, is the identity matrix, K_k , NxM, is the Kalman gain matrix, $\hat{\boldsymbol{x}}_k$, Nx1, is the posteriori state estimate and , Nx1, $\hat{\boldsymbol{x}}_k^-$ is the a priori state estimate, at time instance k .

Initialization

$$\boldsymbol{P}_0 = \boldsymbol{P}_{\boldsymbol{x}_0} \quad \hat{\boldsymbol{x}}_0 = E\left[\boldsymbol{x}\right]$$

Time Update

$$\begin{aligned} \boldsymbol{P}_{k+1}^- &= \boldsymbol{A}_k \boldsymbol{P}_k \boldsymbol{A}_k^T + \boldsymbol{C}_k \boldsymbol{Q}_k \boldsymbol{C}_k^T \\ \hat{\boldsymbol{x}}_{k+1}^- &= \boldsymbol{A}_k \hat{\boldsymbol{x}}_k + \boldsymbol{B}_k \boldsymbol{u}_k \end{aligned} \tag{18}$$

Measurement Update

$$\begin{aligned} \boldsymbol{K}_{k+1} &= \boldsymbol{P}_{k+1}^- \boldsymbol{D}_{k+1}^T \left(\boldsymbol{D}_{k+1} \boldsymbol{P}_{k+1}^- \boldsymbol{D}_{k+1}^T + \boldsymbol{R}_{k+1}\right)^{-1} \\ \boldsymbol{P}_{k+1} &= \left(I - \boldsymbol{K}_{k+1} \boldsymbol{D}_{k+1}\right) \boldsymbol{P}_{k+1}^- \\ \hat{\boldsymbol{x}}_{k+1} &= \hat{\boldsymbol{x}}_{k+1}^- + \boldsymbol{K}_{k+1} \left(z_{k+1} - \boldsymbol{D}_{k+1} \hat{\boldsymbol{x}}_{k+1}^-\right) \end{aligned}$$

In the context of system identification, the main goal is the ability to identify the state of the system at any given time. In terms of Kalman filter, this means that the system described by (16) and (17) should be observable, i.e., the observability matrix (19) should be of rank N.

$$O = \left[\boldsymbol{D}_k \;\; \boldsymbol{D}_k \boldsymbol{A}_k \ldots \boldsymbol{D}_k \boldsymbol{A}_k^{N-1}\right]^T \tag{19}$$

The algorithm of (18) is appropriate for on-line applications, where the optimal estimate at time instance *k* is strictly based on observation up to *k*. However, for off-line applications it is advantageous to employ all observation points to provide an estimate for all time instances. This formulation is referred to as smoothing Kalman filter, and is described in

(20), where $\boldsymbol{P}_k^s$,NxN, $\boldsymbol{x}_k^s$, Nx1, and $\boldsymbol{F}_k$, NxN, are smoothing error covariance, estimated vector and Kalman gain matrices, at instance k. It is noted that before smoothing can be employed, Kalman filter is applied for the duration L and optimal state estimates along with their respective posteriori covariance matrices are recorded.

$$
\begin{aligned}
&\textit{Initialization}\\
&\quad \boldsymbol{P}_L^s = \boldsymbol{P}_L \quad \boldsymbol{x}_L^s = \hat{\boldsymbol{x}}_L\\
&\textit{Update } k \in \{L-1 \cdots 0\}\\
&\quad \boldsymbol{x}_k^s = \hat{\boldsymbol{x}}_k + \boldsymbol{F}_k\left(\boldsymbol{x}_{k+1}^s - \hat{\boldsymbol{x}}_{k+1}^-\right)\\
&\quad \boldsymbol{F}_k = \boldsymbol{P}_k \boldsymbol{A}_k \left(\boldsymbol{P}_{k+1}^-\right)^{-1}\\
&\quad \boldsymbol{P}_k^s = \boldsymbol{P}_k - \boldsymbol{F}_k\left(\boldsymbol{P}_{k+1}^- - \boldsymbol{P}_{k+1}^s\right)\boldsymbol{F}_k^T
\end{aligned}
\tag{20}
$$

4. Full Update Kalman Least Mean Square Algorithms for System Tracking

In this section, we present two Kalman filter based Least Mean Square (LMS) update algorithms, for both on-line and posthumous system tracking applications. In both cases, the traditional LMS algorithm is modeled as a noisy observation plant of a Kalman filter, with Kalman filter states representing the update coefficients. It is shown that the proposed update algorithms achieve superior performance compared to the traditional LMS and RLS, under diverse conditions, without compromising the convergence time. In particular, it is shown that the proposed on-line algorithm is capable of achieving lower Mean Square Deviation (MSD) bound than LMS. Fig. 2 describes the proposed combined algorithms setup.

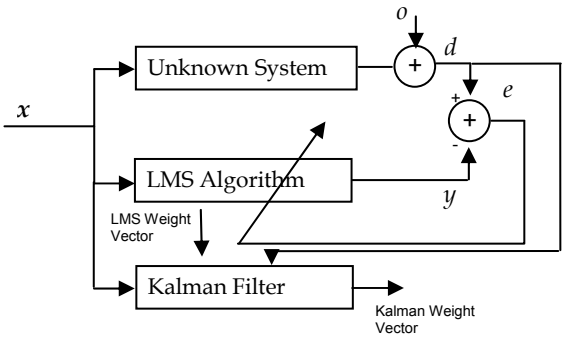


Fig. 2. Combined Kalman LMS Setup

4.1 Time-invariant Identification

Consider a regular system identification setup using an FIR filter. Let the coefficients of the adaptive LMS FIR filter at instant k be given by $\boldsymbol{w}_k^{LMS}$ and let the coefficients of the unknown time invariant system be given by $\boldsymbol{w}_k^{Opt}$. We can then express LMS coefficients as observation vector of a Kalman filter, whose state equation describes the unknown system

coefficients,(21), where (21.a) is the state evolution plant, similar to (16), (21.b) is the observation plant, similar to, (17), and ε_k^w is a zero mean Gaussian noise with variance σ_w^2 .

$$\begin{aligned} \boldsymbol{w}_{k+1}^{Opt} &= \boldsymbol{w}_k^{Opt} && \text{a)} \\ \boldsymbol{w}_k^{LMS} &= \boldsymbol{w}_k^{Opt} + \varepsilon_k^w && \text{b)} \end{aligned} \tag{21}$$

σ_w^2 may be calculated as follows. Firstly, it is noted that MSD of LMS is bounded by (22), (Haykin, 2002), where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of the input signal correlation matrix.

$$\frac{J_{ex}(k)}{\lambda_{max}} \le MSD_k \le \frac{J_{ex}(k)}{\lambda_{min}} \tag{22}$$

Secondly, (13) may be rewritten as (23).

$$J_{ex}(\infty) = M\sigma_o^2 \tag{23}$$

Thirdly, for unit variance white noise, (24) holds.

$$\lambda_{\min} = \lambda_{\max} = \sigma_x^2 = 1 \tag{24}$$

Combining (22) through (24), expression (25) for σ_w^2 is obtained.

$$\sigma_w^2 = \frac{\mu N \sigma_O^2}{2} \tag{25}$$

It can be shown, (Lewis, 1986), that in time-invariant case, Kalman filtering amounts to a simple weighted average of the observed LMS weight vector. That is, assuming a typical Kalman update, the independence of the weights and an initial Gaussian distributed weight vector, $\boldsymbol{w}_0$, of mean $\overline{\boldsymbol{w}}$ and variance $\sigma_{w_0}^2$, after k iterations the following updates are obtained.

$$\begin{aligned} P_k^{i^-} &= \frac{\sigma_{\boldsymbol{w}_0}^2}{1 + (k-1)\left(\frac{\sigma_{\boldsymbol{w}_0}^2}{\sigma_{\boldsymbol{w}}^2}\right)} \\ P_k^i &= \frac{\sigma_{\boldsymbol{w}_0}^2}{1 + k\left(\frac{\sigma_{\boldsymbol{w}_0}^2}{\sigma_{\boldsymbol{w}}^2}\right)}; \quad K_k^i = \frac{\frac{1}{\sigma_{\boldsymbol{w}}^2}}{\frac{1}{\sigma_{\boldsymbol{w}_0}^2} + \frac{k}{\sigma_{\boldsymbol{w}}^2}} \\ \hat{w}_k^i &= \frac{1}{1 + k\left(\frac{\sigma_{\boldsymbol{w}_0}^2}{\sigma_{\boldsymbol{w}}^2}\right)}\left[\hat{w}_0^i + \left(\frac{\sigma_{\boldsymbol{w}_0}^2}{\sigma_{\boldsymbol{w}}^2}\right)\sum_{j=1}^{k} w_k^{i\,LMS}\right] \\ &\forall i \in \{1 \cdots N\} \end{aligned} \tag{26}$$

where $P_k^{i^-}, P_k^i, K_k^i, \hat{w}_k^i, \sigma_{w_0}^2$ and $\hat{w}_0^i$ are the i^{th} element of the a priori error covariance, posteriori error covariance, Kalman gain, optimal estimate, initial error covariance, and initial weight estimate, respectively. For a typical system tracking and identification, we

may assume zero prior knowledge, *i.e.* $\sigma^2_{\boldsymbol{w}_0} = \infty$ and $\hat{w}^i_0 = 0$, in which case the Kalman optimal weight estimate simplifies to (27).

$$\begin{aligned} &\hat{w}^i_k = \hat{w}^i_{k-1} + \frac{1}{1+k}\left[w^{i^{LMS}}_k - \hat{w}^i_{k-1} \right] \\ &\forall i \in \{1 \cdots N\} \end{aligned} \tag{27}$$

4.2 Time Variant System Tracking

Based on (27), it is clear that measurement contributions are ignored as *k* increases. Hence, system model of (21) would not be able to accommodate tracking of time-varying systems. To address this shortcoming, we choose the update scheme of (28) based on Steepest Descent (SD), (5), to replace (21.a). Under the system identification assumptions input signal correlation matrix, input vector and observation scalar are available. However, the input-observation cross correlation function, $\boldsymbol{r}^k_{\boldsymbol{xd}}$, must be estimated. For ease of implementation, let this estimate be a simple recursive averaging, as in (29), with α between 0 and 1.

$$\begin{aligned} \boldsymbol{w}^{SD}_k &= \boldsymbol{w}^{SD}_{k-1} + \mu\left[\boldsymbol{r}^k_{\boldsymbol{xd}} - \boldsymbol{R}^k_{\boldsymbol{xx}} \boldsymbol{w}^{SD}_{k-1}\right] \\ &= \left(I - \mu \boldsymbol{R}^k_{\boldsymbol{xx}}\right)\boldsymbol{w}^{SD}_{k-1} + \mu \boldsymbol{r}^k_{\boldsymbol{xd}} \end{aligned} \tag{28}$$

$$\hat{\boldsymbol{r}}^k_{\boldsymbol{xd}} = \alpha \hat{\boldsymbol{r}}^{k-1}_{\boldsymbol{xd}} - (1-\alpha)\boldsymbol{x}(k)d(k) \tag{29}$$

Similarly, let the variance of $\boldsymbol{r}^k_{\boldsymbol{xd}}$, $\sigma^2_{\boldsymbol{x}^k d^k}$, be approximated by (30), with β between 0 and 1.

$$\sigma^2_{\boldsymbol{x}^k d^k} = \beta \sigma^2_{\boldsymbol{x}^{k-1} d^{k-1}} + (1-\beta)\left[\boldsymbol{x}(k)d(k)\right]^2 \tag{30}$$

It is noted that (Taylor, 1997)

$$\hat{\boldsymbol{r}}^k_{\boldsymbol{xd}} = \boldsymbol{r}^k_{\boldsymbol{xd}} + \overline{\varepsilon^{\boldsymbol{xd}}_k} \tag{31}$$

where $\overline{\varepsilon^{\boldsymbol{xd}}_k}$ is approximated as a zero mean Gaussian noise with variance $\sigma^2_{\overline{\varepsilon^{\boldsymbol{xd}}_k}}$. $\sigma^2_{\overline{\varepsilon^{\boldsymbol{xd}}_k}}$ may be approximated by the first differential of (29) as in (32):

$$\begin{aligned} \sigma^2_{\overline{\varepsilon^{\boldsymbol{xd}}_k}} &= \left(\frac{\partial \hat{\boldsymbol{r}}^k_{\boldsymbol{xd}}}{\partial \boldsymbol{x}^k d^k} \sigma_{\boldsymbol{x}^k d^k} \right)^2 \\ &= (1-\alpha)^2 \left(\beta \sigma^2_{\boldsymbol{x}^{k-1} d^{k-1}} + (1-\beta)\left[\boldsymbol{x}(k)d(k)\right]^2 \right) \end{aligned} \tag{32}$$

Based on the above, the following state-space description is obtained.

$$\begin{aligned} \boldsymbol{w}^{SD}_k &= \left(I - \mu \boldsymbol{R}^k_{\boldsymbol{xx}}\right)\boldsymbol{w}^{SD}_{k-1} + \mu \boldsymbol{r}^k_{\boldsymbol{xd}} \\ \boldsymbol{w}^O_k &= \boldsymbol{w}^{SD}_k + \mu \overline{\varepsilon^{\boldsymbol{xy}}_k} \end{aligned} \tag{33}$$

where $\boldsymbol{w}^O_k$ is the observed weight vector. However, it is noted that, unlike (21), the Kalman setup of (33) does not take into account the observation noise. To remedy this, we combine the setup of (21) and (33) as in (34), where, by comparing (5) and (8), it is noted that $\boldsymbol{w}^{LMS}_k$, the observation vector of this Kalman LMS-SD model, also requires less computational effort than that of a typical SD algorithm.

$$\begin{aligned} \boldsymbol{w}_k^{SD} &= \left(I - \mu \boldsymbol{R}_{\boldsymbol{xx}}^k\right)\boldsymbol{w}_{k-1}^{SD} + \mu \boldsymbol{r}_{\boldsymbol{xd}}^k + \mu \varepsilon_k^{\overline{xd}} \\ \boldsymbol{w}_k^{LMS} &= \boldsymbol{w}_k^{SD} + \varepsilon_k^{w} \end{aligned} \tag{34}$$

Furthermore, it is noted that since for a stable LMS both the drift matrix, $\boldsymbol{A}_k = \left(I - \mu \boldsymbol{R}_{\boldsymbol{xx}}^k\right)$,and degradation matrix, $\boldsymbol{D}_k = I$, are full rank, the proposed Kalman filter is indeed observable. The setup of (34) can be employed either on-line, employing typical Kalman filter, or posthumously, using smoothing Kalman filter.

4.3 Verification and Simulation Results

The performance of the system described by (34) has been studied for unity input variance, N=50 randomly generated taps, β=α, simulation length of 2000 steps, various observation noises and step sizes. For time varying systems, the optimum coefficients are assumed to be a constant, modulated by a sine wave, with a randomly generated frequency of variances 0.5 and 5, corresponding to slowly and rapidly varying systems, respectively. To gain further understanding of the performance of on-line Kalman LMS-SD method (KLMSSD) and posthumous Kalman LMS-SD method (PKLMSSD), comparison against both traditional LMS and RLS method are included. The results are summarized in Table . In the following simulations α, β and λ_{RLS}, RLS forgetting factor, are selected, in order to minimize the MSE of their respective adaptive algorithm. Fig. 3 shows a typical performance comparison of LMS, KLMSSD, PKLMSSD and RLS algorithms. It is noted that while Kalman based algorithms retain the tracking capability of LMS, unlike RLS, they do so without introducing excessive noise in their output. Here, the vertical lines indicate the instances that LMS, KLMSSD, PKLMSSD and RLS have detected the zeros crossing of the 2 representative system coefficients. The significant delayed response of the RLS is noted. The poor performance of RLS can be readily identified in its state-space form, (Haykin, 2002), (35).

$$\begin{aligned} \boldsymbol{w}_{k+1}^{RLS} &= \lambda_{RLS}^{-\frac{1}{2}} \boldsymbol{w}_k^{RLS} \qquad 0 \le \lambda_{RLS} \le 1 \\ y_k &= \boldsymbol{x}_k^T \boldsymbol{w}_k^{RLS} + \lambda_{RLS}^{-\frac{k}{2}} \eta_k \end{aligned} \tag{35}$$

where T is the transpose operator, x is the Nx1 input vector and η is the zero-mean Gaussian observation noise, with variance σ_o^2 .

Algorithm	σ_O=1 High SNR						σ_O=10 Low SNR					
	Time Invariant		Slowly Varying		Rapidly Varying		Time Invariant		Slowly Varying		Rapidly Varying	
	MSE	MSD	MSE	MSD	MSE	MSD	MSE	MSD	MSE	MSD	MSE	MSD
LMS	1.82 μ=0.012	122	19.8 μ=0.028	144	1859 μ=0.028	142	165 μ=0.014	118	490 μ=0.028	136	2351 μ=0.028	138
RLS	1.09 λ=1	141	11.7 λ=0.95	141	609 λ=0.9	137	109 λ=1	148	248 λ=0.95	144	1235 λ=0.9	139
KLMSSD	1.30 α=0.995	128	4.36 α=0.995	133	176 α=0.99	136	73 α=0.95	118	145 α=0.995	126	1036 α=0.99	123
PKLMSSD	0.84	116	1.65	129	122	134	67	111	61	107	307	106

Table I LMS, RLS, KLMSSD and PKLMSSD Performance Table

Comparing the system plant processes of (35) and (34) it is evident that RLS has no direct means by which it would be able to track changes in the unknown system. Its state-space form essentially assumes that the current state is that of the optimal state plus observation noise; whereas, KLMSSD is able to detect and track changes in the unknown system due to changes in $\boldsymbol{r}_{\boldsymbol{xd}}^{k}$.

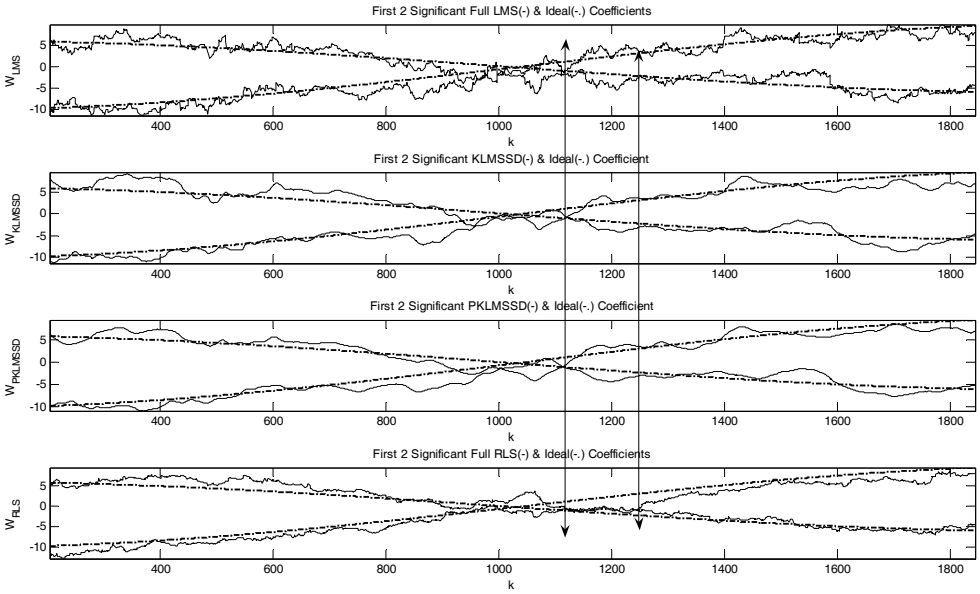


Fig. 3. LMS, Kalman LMS, Smoothing Kalman LMS and RLS Comparison for Slowly Varying System Identification

4.4 KLMSSD MSD BOUND

In this section, we provide an existence proof to show that MSD bound lower than LMS is achievable via KLMSSD. To investigate the MSD obtained by the KLMSSD, the covariance error matrix of (34) is analyzed for the wide sense stationary case. It is noted that as $k\to\infty$ then $\boldsymbol{P}_{k^-}\to \boldsymbol{P}_{k^+} \to \boldsymbol{P}$, where $\boldsymbol{P}_{k^-}$ is the a priori covariance error and $\boldsymbol{P}_{k^+}$ is the posteriori covariance error of the Kalman filter. For a stabilized system, $\boldsymbol{P}$ may be expressed as (36), where Q, the expected value of (32), is given in (37).

$$\boldsymbol{P}=\left\{\left(I-\mu\boldsymbol{R}_{\boldsymbol{xx}}\right)\left[\boldsymbol{P}-\boldsymbol{P}\left(\boldsymbol{P}-\frac{\mu N\sigma_o^2}{2}\right)^{-1}\boldsymbol{P}\right]\left(I-\mu\boldsymbol{R}_{\boldsymbol{xx}}\right)^{T}\right\}+\mu Q\mu \tag{36}$$

$$\begin{aligned} Q &= E\left[(1-\alpha)^2\left(\beta\sigma_{xd}^2+(1-\beta)\left(x^k d^k\right)^2\right)\right] \\ &= (1-\alpha)^2\left(\beta\sigma_{xd}^2+(1-\beta)E\left[\left(x^k d^k\right)^2\right]\right) \\ &= (1-\alpha)^2\left(\beta\sigma_{xd}^2+(1-\beta)\sigma_{xd}^2\right) \\ &= (1-\alpha)^2\sigma_{xd}^2 \end{aligned} \tag{37}$$

Noting that $\boldsymbol{P}$ is a diagonal matrix with diagonal element p_i, (36) simplifies to (38), which can further be simplified to (39).

$$\begin{aligned} p_i &= \left(1-\mu\sigma_x^2\right)^2\left[p_i-\frac{{p_i}^2}{\left(p_i-\frac{\mu N\sigma_o^2}{2}\right)}\right]+\mu^2\left(1-\alpha^2\right)\sigma_{xd}^2 \\ &= \frac{\left[\left(1-\mu\sigma_x^2\right)^2 p_i\left(p_i-\frac{\mu N\sigma_o^2}{2}\right)-\left(1-\mu\sigma_x^2\right)^2{p_i}^2+\left(p_i-\frac{\mu N\sigma_o^2}{2}\right)\mu^2\left(1-\alpha^2\right)\sigma_{xd}^2\right]}{p_i-\frac{\mu N\sigma_o^2}{2}} \end{aligned} \tag{38}$$

$\forall\, i\in\{1\cdots N\}$

$$\begin{aligned} &{p_i}^2-p_i b_i+c_i=0 \\ &\forall\, i\in\{1\cdots N\} \end{aligned} \tag{39}$$

where

$$\begin{aligned} b_i &= \frac{\mu N\sigma_o^2}{2}\left(1-\left(1-\mu\sigma_x^2\right)^2+\mu^2\left(1-\alpha^2\right)\sigma_{xd}^2\right) \\ c_i &= \frac{\mu N\sigma_o^2}{2}\mu^2\left(1-\alpha^2\right)\sigma_{xd}^2 \end{aligned} \tag{40}$$

Given the stability condition of (9), it follows that $\mu\sigma_x^2<1$, for N >2.

Thus $\left(1-\mu\sigma_x^2\right)<1$ and therefore b_i>0. Similarly, it is also noted that c_i>0. Quadratic solutions of (39) are given by (41)

$$\begin{aligned} &p_i^n=\frac{1}{2}\left[b_i-\sqrt{{b_i}^2-4c_i}\right] \quad p_i^p=\frac{1}{2}\left[b_i+\sqrt{{b_i}^2-4c_i}\right] \\ &p_i^n\le p_i^p \end{aligned} \tag{41}$$

Furthermore, noting that $b_i\ge\sqrt{{b_i}^2-4c_i}$, for all real solutions, from (41), (42) follows.

$$p_i^n\le p_i^p\le b_i \tag{42}$$

Given that the first factor of b_i in (40) is the same MSD bound as LMS, for a bound lower than LMS to hold, the second factor of (40) should be less than unity. Equation (43) identifies the requirement for this assertion.

$$\begin{aligned} &1-\left(1-\mu\sigma_{\boldsymbol{x}}^2\right)^2+\mu^2\left(1-\alpha^2\right)\sigma_{\boldsymbol{xd}}^2<1 \\ &-\left(1-\mu\sigma_{\boldsymbol{x}}^2\right)^2+\mu^2\left(1-\alpha^2\right)\sigma_{\boldsymbol{xd}}^2<0 \\ &\sigma_{\boldsymbol{xd}}^2<\frac{\left(1-\mu\sigma_{\boldsymbol{x}}^2\right)^2}{\mu^2\left(1-\alpha^2\right)}<\frac{\left(1-\dfrac{2\sigma_{\boldsymbol{x}}^2}{N\sigma_{\boldsymbol{x}}^2}\right)^2}{\mu^2\left(1-\alpha^2\right)}=\frac{\left(1-\dfrac{2}{N}\right)^2}{\mu^2\left(1-\alpha^2\right)} \end{aligned} \tag{43}$$

Based on (44), this requirement may be expressed in terms of observation noise,(45), where the last line of (44) has been obtained under the assumption of white noise input and independence of weights. Since both μ and α are independent design parameters, they may be chosen in such manner to unconditionally satisfy any range of σ_o^2, *e.g.* for α close to unity, the allowable range is ∞.

$$\begin{aligned} \boldsymbol{\sigma}_{\boldsymbol{xd}}^2 &= E\left[\boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{X}^T\right] \\ &= E\left[\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{W}^{Opt}+\boldsymbol{O}\right)\left(\boldsymbol{W}^{Opt}+\boldsymbol{O}\right)^T\boldsymbol{X}\boldsymbol{X}^T\right] \\ &= \boldsymbol{R}_{ww}=\delta_{ij}w_iw_j\begin{cases}w_j^2+\sigma_o^2 & j=i\\ \sigma_o^2 & j\neq i\end{cases} \end{aligned} \tag{44}$$

$$\sigma_o^2<\frac{\left(1-\dfrac{2}{N}\right)^2}{\mu^2\left(1-\alpha^2\right)}-\max\left(w_i^2\right)_{i\in\{1\cdots N\}} \tag{45}$$

Hence, $\begin{cases}\exists\alpha \text{ and } \mu \mid p_i^n\le p_i^p\le b\le LMS_{MSD}\\ \forall\, i\in\{1\cdots N\}\end{cases}$, that is, an MSD bound, lower than LMS bound is possible.

5. Partial Update Kalman Least Mean Square Algorithms for System Tracking

This section presents a novel partial update algorithm for FIR adaptive filters based on a Kalman background engine. In the proposed system, a Kalman filter is setup with the coefficients of the full adaptive filter being the states to be estimated based on the observation of a subset of filter coefficients being updated. It is shown that this setup allows for an improved estimation of the full set of filter coefficients, despite the partial update. We propose two methods for postmortem improvements on an ordinary M-Tap periodic update LMS. We also propose a Kalman feedback method, in conjunction with a 1-Tap periodic update LMS, which has a similar performance as a full length LMS, for non-stationary system identification.

5.1 Length Constrained Adaptive FIR Kalman Filter (LCAFKF)

Let the ideal, vector form, full length LMS update be described as in (46).

$$\boldsymbol{w}_{k+1}=\boldsymbol{w}_k+\boldsymbol{\mu}_k\boldsymbol{x}_k e_k \tag{46}$$

where $\boldsymbol{w}_k$ is the Nx1 weight vector. The length-constrained adaptive filter, M<N, can be described as (47)

$$\boldsymbol{w}_k^O = \boldsymbol{D}_k \boldsymbol{w}_k \tag{47}$$

where $\boldsymbol{w}_k^O$ is the observable length-constrained weight vector of length Mx1, M≤N, and $\boldsymbol{D}_k$, MxN, is the matrix determining which coefficients of $\boldsymbol{w}_k$ are present in $\boldsymbol{w}_k^O$ and is given by:

$$\boldsymbol{D}_k = \left[\boldsymbol{I} \middle| \boldsymbol{0}\right] \tag{48}$$

where $\boldsymbol{I}$ is the MxM identity matrix and $\boldsymbol{0}$ is M x (N-M) all-zero matrix. Our objective is to estimate $\boldsymbol{w}_k$ based on the limited knowledge obtained from $\boldsymbol{w}_k^O$. Considering the above definitions, we now propose to formulate the problem as a Kalman filtering system. A typical Kalman filter is described as in (49):

$$\begin{aligned} \boldsymbol{w}_{k+1} &= \boldsymbol{w}_k + \boldsymbol{x}_k \xi_k \\ \boldsymbol{w}_k^O &= D_k \boldsymbol{w}_k \end{aligned} \tag{49},$$

where ξ_k is a Gaussian noise with variance $\mu^2 J(\infty)$.
With the formulation complete, we now consider the feasibility of the solution to this problem. The observability matrix of this Kalman filter described above is given by:

$$O = \left[D_k I \ldots D_k I^{N-1}\right]^T \tag{50}$$

However, we can see that the rank of O in (50) is M<N. It then follows that with the current Kalman formulation, the optimal estimate of the full weight vector $\boldsymbol{w}_k$ cannot be obtained, based on observations of a length-constrained adaptive filter $\boldsymbol{w}_k^O$. This result is intuitive and the details are included here as an introduction for the next system.

5.2 Partial Update Adaptive FIR Kalaman Filter (PUAFKF)

Given that the original objective is to estimate the full length coefficient vector using fewer observations, we now consider other forms of observation matrix, along with their accompanying assumptions. Of particular interest is the partial coefficient update case, where the observation vector is indeed full length Nx1, but the observation is limited by the fact that only a subset of M coefficients is periodically updated per iteration. That is, in the first time step, only the first M taps are updated, in the second time, taps M+1 through 2M are updated and so on. The updated coefficients will be accurately observed with zero observation error while the remaining coefficients that were not updated will have an observation error equivalent to the missing update term with negative sign. The observed coefficients $\boldsymbol{w}_k^O$ can then be expressed as:

$$\boldsymbol{w}_k^O = \boldsymbol{w}_k - \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \sum_{i=0}^{(m+1)\left\lceil \frac{N}{M} \right\rceil} \mu_k e(k-i)x(k-i-M) \\ \vdots \\ \sum_{i=0}^{m\left\lceil \frac{N}{M} \right\rceil} \mu_k e(k-i)x(k-i-2M) \\ \vdots \\ \mu_k e(k)x(k-N-M) \\ \vdots \\ \mu_k e(k)x(k-N-1) \end{bmatrix} \tag{51}$$

where $\boldsymbol{0}$ is an Mx1 zero vector , and $m = \left\lceil \frac{N}{M} \right\rceil$ is the number of M-blocks in an N tap filter. Assuming LMS update and grouping the error terms together, with (51) can be rewritten as:

$$\boldsymbol{w}_k^O = \boldsymbol{w}_k - \mu \left\{ \begin{array}{l} e(k) \begin{bmatrix} \boldsymbol{0} \\ x(k-M) \\ \vdots \\ x(k-N-1) \end{bmatrix} + \cdots + \\ \\ + \ldots + e(k-m-2) \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ x(k-m-M) \\ \vdots \\ x(k-m-2M) \\ \boldsymbol{0} \end{bmatrix} \end{array} \right\} \tag{52}$$

Comparing (52) with the observation equation in Eq (49), it follows that the observation matrix for this partial update case is an identity matrix of size NxN. Considering the observability matrix, (19), it is clear that O is of rank N, *i.e.* the system is observable and we can indeed get an estimate for the full length filter coefficients based on a partially updated set of coefficients. To complete the necessary formulation, a stochastic description of observation noise and update terms in (52) is required. Proceeding under the typical LMS setup assumption, the problem simplifies to finding the mean and correlation matrix of the $e(k)x(k\text{-}i)$ process, for $i \in \{1,\ldots,N\text{-}1\}$. The mean of this process is estimated as in (53).

$$\begin{aligned} E\left[e(k)x(k-i)\right] &= E\left[\left(d(k) - \boldsymbol{x}_k^T \boldsymbol{w}_k\right)x(k-i)\right] \\ &= E\left[d(k)x(k-i)\right] - w_k^i \sigma_x^2 \\ &= w_i^W \sigma_x^2 - w_k^i \sigma_x^2 \\ &= \Delta w_k^i \sigma_x^2 \end{aligned} \tag{53}$$

Where $\boldsymbol{w}^W$ is the optimal solution, and the third line follows from Wiener-Hopf equation and the white Gaussian assumption of the input. Noting that

$\Delta w_{k+1}^i = \left(1-\mu\sigma_x^2\right)\Delta w_k^i - \mu x(k-i)e^O(k)$, (Haykin, 2002), where e^O is the additive observation noise, then (53) simplifies to (54)

$$\begin{aligned} \Delta w_k^i r_0^x &= \left[-\mu e_k^O x(k-i) * \left(1-\mu\sigma_x^2\right)^k U(k)\right]\ \sigma_x^2 \\ E\left[e(k)x(k-i)\right] &= \Delta w_k^i \sigma_x^2 \xrightarrow{k\to\infty} 0 \end{aligned} \tag{54}$$

where $U(.)$ is the step function, and * the convolution operation. The second line of (54) follows from the exponential decay of the convolution term. The variance of this process is estimated as:

$$\begin{aligned} &E\left[e(k)x(k-i)e(k)x(k-j)\right] = \\ &\quad E\left[e(k)e(k)\right]E\left[x(k-i)x(k-j)\right] \\ &+ E\left[e(k)x(k-i)\right]E\left[e(k)x(k-j)\right] \\ &= J(\infty)\sigma_x^2 \end{aligned} \tag{55}$$

The second line follows from the Gaussian moment factoring theorem, (Haykin, 2002), and the third line from (54).

5.3 Performance Evaluation through Simulations

The proposed system described above is composed of a regular adaptive FIR filter of length N, where M coefficients are being updated sequentially, at every iteration. The observed coefficient vector is fed into a Kalman filter as the observation vector to produce an improved estimate for the full length adaptive filter. We consider two scenarios. The first scenario, the Standard Algorithm (SA), the partial updates FIR are fed into the Kalman engine which provides the estimate for the full length filter coefficients in a postmortem fashion. This corresponds to a case where a limited complexity remote system does the preliminary work and then feeds the result to the Kalman filter for improved estimation. In the second scenario, LMS Kalman Feedback Algorithm (LKFA), the output of the adaptive filter is fed to the Kalman filter, as before, furthermore, the Kalman filter output is then fed back to update all the FIR filter coefficients before the next iteration.

To verify the accuracy of the assumptions leading to (55), we estimate the statistics of the observation noise on line as in (56). We refer to this method as Online Observation Plant Noise Variance Estimation Algorithm (OPNVEA).

$$r_i^{ex}(k) = \gamma r_i^{ex}(k-1) + (1-\gamma)\left[e(k)x(k-i)\right]^2 \qquad i\in\{1\cdots N\} \quad 0<\gamma<1 \tag{56}$$

We now provide the simulation results for SA, LKFA and OPNVEA methods, under time invariant, slowly varying and finally rapidly varying system tracking. As points of reference, the MSE of a full length LMS filter and 1-Tap periodic update LMS, *i.e.* M=1, are also provided. The unknown system has 50 taps. Its first 3 significant coefficients and the accompanying SA identification are depicted in Fig. 4, Fig. 5 and Fig. 6 through Fig. 8. The input variance is 3, observation noise variance is 1, and the full LMS step size is $8\text{x}10^{-5}$. For the time varying systems of Fig. 5 through Fig. 8, each coefficient is modulated by a sine wave, with a randomly generated frequency of variances 0.5 and 15, respectively. The coefficients of the unknown system are also randomly generated, with amplitude variance of 10.

Results of the simulations are summarized in Table 2. The first column of the Table 2 refers to the time invariant system identification, the next column to the slowly varying system and the final column to the rapidly varying system. The step size for the 1-Tap periodic update LMS (M=1) is $4x10^{-3}$, the forgetting factor γ for OPNVEA (Scenario 1) is 0.1. As seen in Table 2, all proposed algorithms lead to improvements, compared to 1-Tap periodic LMS algorithm. These improvements become more pronounced in non-stationary system estimation. It is further noted that the performance of the OPNVE is similar to SA, even in the non-stationary case, confirming the validity of the assumptions leading to (55). It is particularly worthwhile to note that the performance, in non-stationary cases, of the full LMS and 1-Tap LKFA are quite similar. With respect to the temporal complexity of the LKFA, it should be noted that since the correlation matrices of the system and observation plants are known ahead of time, the most time consuming portions of the Kalman filter, *i.e.* the post covariance matrix and Kalman gain, can be calculated off-line, thus significantly reducing the real-time constraints on the LKFA.

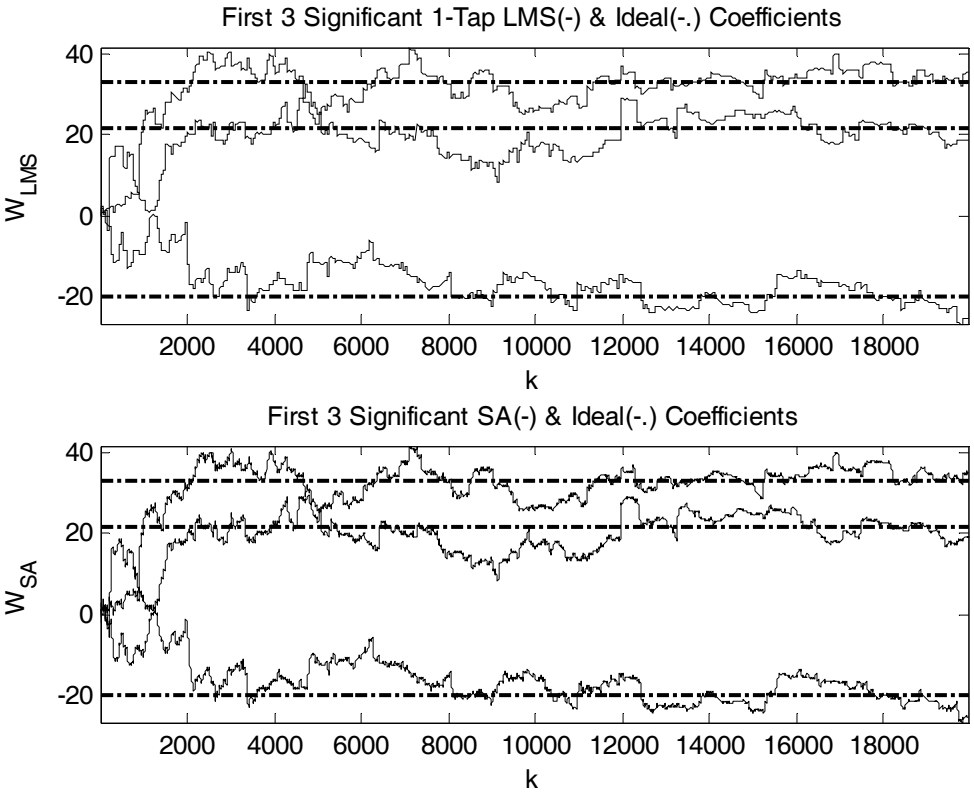


Fig. 4. Stationary System Identification

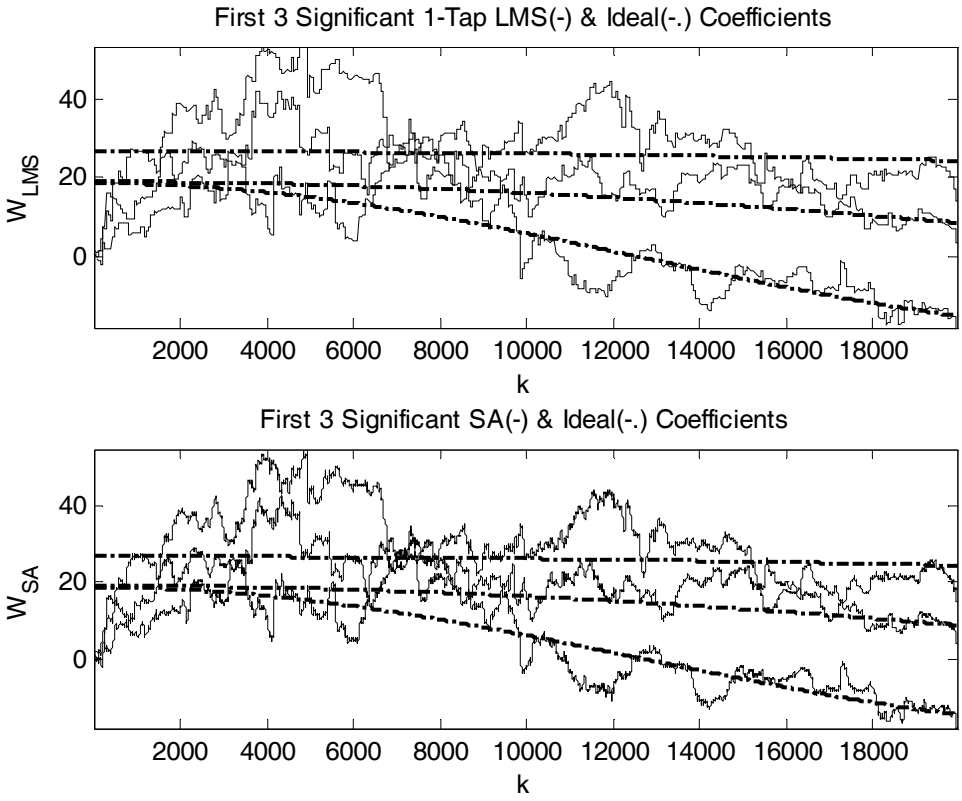


Fig. 5. Slowly Varying System Identification

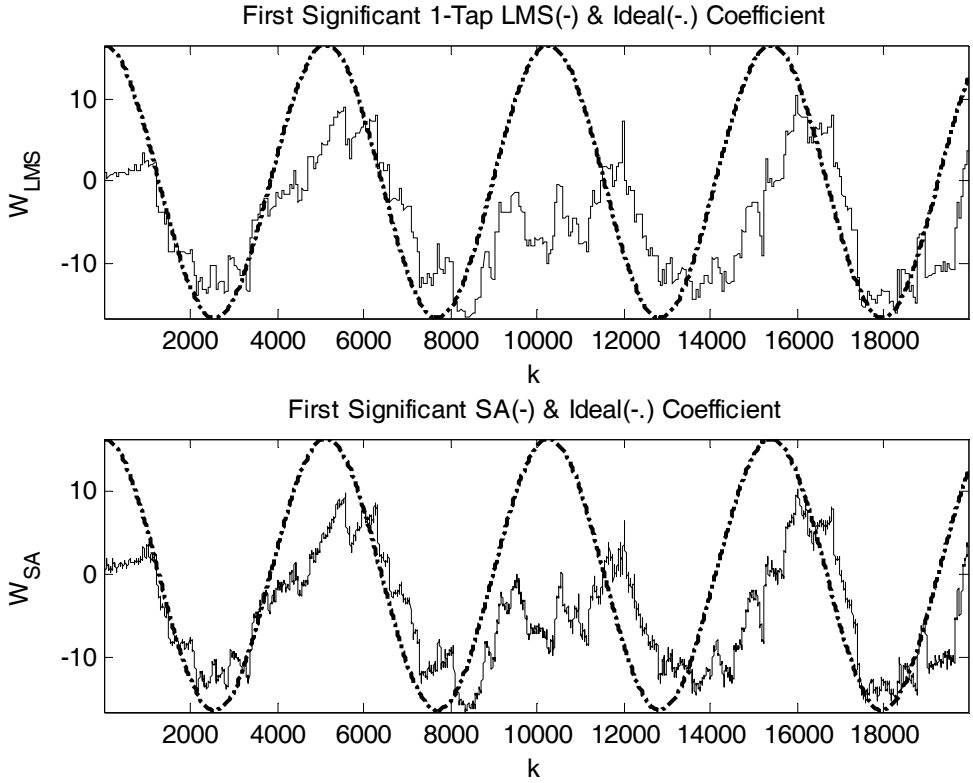


Fig. 6. Rapidly Varying System Identification, 1st Coefficient

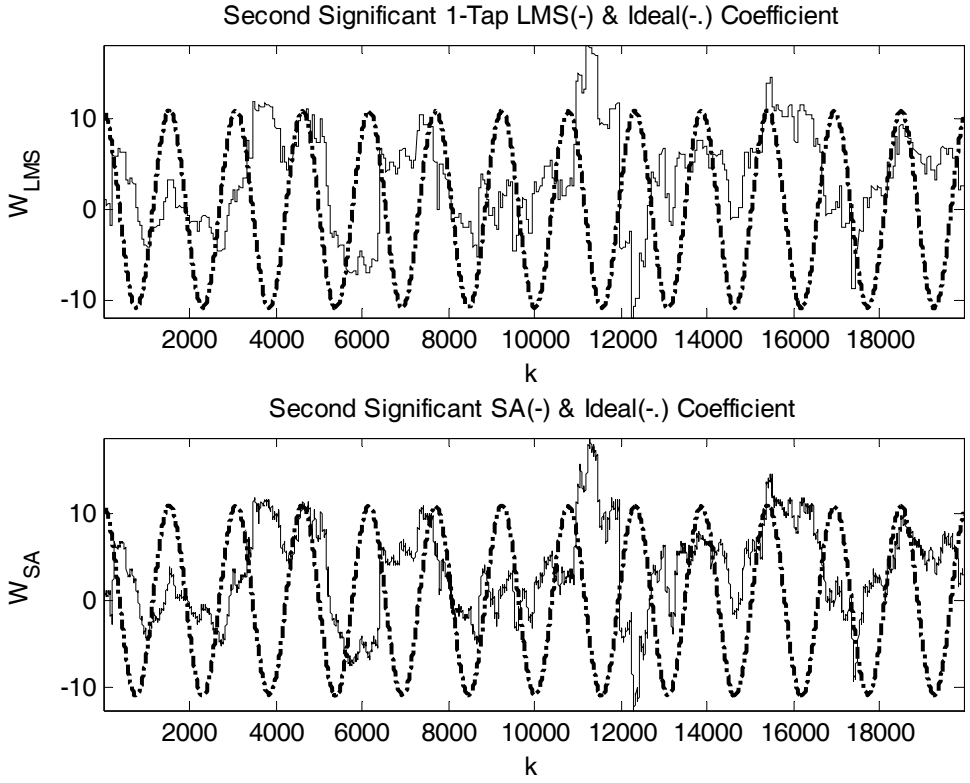


Fig. 7. Rapidly Varying System Identification, 2nd Coefficient

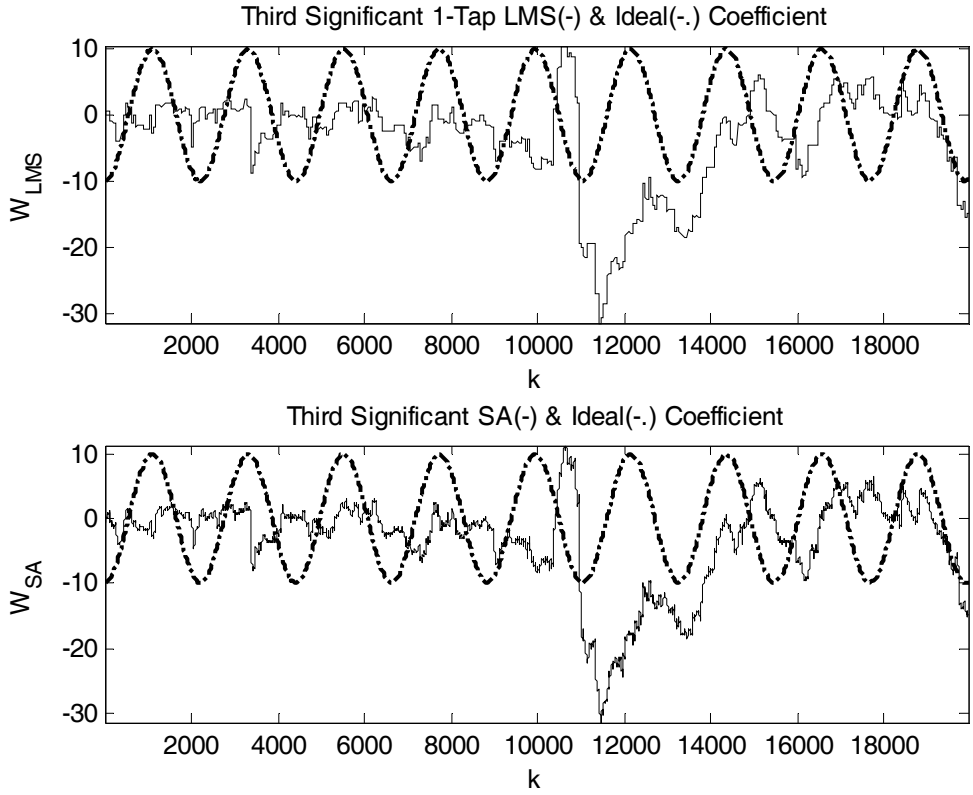


Fig. 8. Rapidly Varying System Identification, 3rd Coefficient

Method Name	Stationary MSE [10^3]	Slowly Varying MSE [10^4]	Rapidly Varying MSE [10^4]
Full LMS	0.003	0.035	1.607
1 Tap LMS	9.915	2.956	36.39
OPNVEA	9.373	2.788	34.40
SA	9.501	2.832	34.83
LKFA	10.217	1.274	2.357

Table 2. Summary of Periodic Update Results

6. Conclusion and future work

We have shown the applicability and feasibility of the Kalman filter modeling for LMS-based algorithms. Particularly, it was observed that such modeling allows for realizations of update methods with MSD bounds lower than those obtained by LMS. Furthermore, it was

shown that the system and observation plant setup provides for a convenient formulation of a length constraint adaptive FIR filters. In particular, it was shown that this setup allows for both post mortem and real-time optimization of system identification process. The future work would focus on application of the current method on update policies, other than the periodic one. Furthermore, the application of smoothing Kalman filter on length-constrained MSE would be investigated.

It is noted that the improvements gained by applying the standard Kalman filter are temporally localized. This is indeed in line with the linear structure of a typical Kalman filter. As such the application of the Extended Kalman filter will also be investigated under non-stationary environments.

7. References

Amit Kumar Kohli , D. K. Mehra (2008), Adaptive Multiuser Channel Estimation using Reduced Kalman/LMS Algorithm, Wireless Personal Communications, Vol. 64, No. 4, 507-521, 0929-6212

Christian Bauckhage and John K. Tsotsos(2005). Separable Linear Classifiers for Online Learning in Appearance Based Object Deection, In: *Computer Analysis of Images and Patterns*, A. Gagalowicz and W. Philips , 347-353, Springer, 978-3-540-28969-2 , Berlin

Frank L. Lewis,(1986)*Optimal Estimation With An Introduction to Stochastic Control Theory*, John Wiely and Sons, 0-47183741-5,New York

John R. Taylor, (1997) *An Introduction to Error Analysis*, University Science Books, 0-935702-42-3, Sausalito

K. Mayyas and T. Aboulnasr (2004), Reduced-Complexity Transform-Domain Adaptive Algorithm with Selective Coefficient Update, IEEE Trans. On Cir. and Sys. II, Vol. 51, No. 3, 136-142

Mal-Rey Lee (2001), 3D Shape Reconstruction of Hybrid Reflectance Using the LMS Algorithm. *IJPRAI*, Vol. 15, No. 4, 723-734, 1793-6381

Masayuki Hoshino, Takeo Ohgane, and Yasutaka Ogawa, Adaptive DEF Control with Differential LMS Algorithm, *Electronics and communication in Japan - Part 1,* Vol. 84, No. 1, Nov. 2001, 99-108, 87566621

Paulo A. C. Lopes, Gonc¸alo Tavares and Jos´e B. Gerald, A New Type of Normalized LMS Algorithm Based on the Kalman Filter, ICSAAP 2007, Vol. 3, 1345-1348, 1-4244-0727-3, USA, April 2007, Honolulu

S.C. Douglas (1997), Adaptive Filters Employing Partial Updates, IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing, Vol. 44, No. 3, 209-216

Simon Haykin, (2002) *Adaptive Filter Theory*, 4th Ed., Prentice Hall, 0-13-048434-2, New Jersey

Sristi, P. Lu, W.-S. and Antonion, A. (2001), **ISCAS 2001,** pp. 721-724, 0-7803-6685-9, Australia, May, 2001, IEEE, Sydney

Yang Xinyu, Zeng Ming, Zhao Rui and Shi Yi(2004). A Novel LMS Method for Real-Time Network Traffic Prediction, Proceedings of ICCSA 2004,12-136, 978-3-540-22060-2,Italy, May 2004, Springer, Assisi

Delay Difference between the Linear Traffic Model and the Polynomial Traffic Model

Woo-Young Ahn, Seon-Ha Lee and Gyeong-Seok Kim
Faculty of Department of Civil & Environmental Engineering
Kongju National University
South Korea

1. Introduction

In traffic-responsive signal control, a technique for interpreting the real-time detector data plays an important role in decision making process. The traditional Linear Traffic Model (LTM) is called a vertical queue traffic model and it is adopted widely in traffic signal control at signalized intersections. This model represents all vehicles have the same travel time before joining a queue. Thus, a queue is supposed to form vertically at the stop-line without occupying any space on the link. Due to the simplicity of this model, all vehicle motions are identical and unaffected by the signal display on their approach to the intersection. In the linear traffic model, the departure time of the leading vehicle is assumed to coincide with the start of the effective green time (Clayton, 1940; Webster & Cobbe; 1966; Allsop, 1970) and then departure times for the successive following vehicles are estimated in accordance with the saturation departure time at the stop-line. The delays and departure time estimates given by the vertical queue traffic models are not always realistic, because this model does not consider any braking motion until the stop-line. Namely, it is more focused on the queue delay rather than the geometric delay. The Robertson (1969) and TRANSYT (2004) uses vertical queue concepts in fixed-time signal optimization. Also, in traffic-responsive signal control, Miller (1963), Gartner (1983), and Heydecker (1990) use constant mean travel time from the detector to the stop-line. Traditional traffic theory focuses on modeling of queue discharge flow rates at signalized intersections with relatively little consideration of the corresponding queue discharge speed characteristics. This model uses a constant saturation flow rate and converts to an effective green time (Webster & Cobbe, 1966; Akcelick, 1981 & TRB, 2000).

In traffic-responsive signal control, the motion of each vehicle from the upstream detector to the downstream stop-line is needed for full interpretation of the detector output and for performance evaluation. It would be tedious and time consuming to describe the motion of each vehicle in detail at each time-step. Akcelick, et al (2002) describes the exponential queue discharge flow and speed models. In this paper, the concepts of kinematics in physics are applied to derive a Polynomial Traffic Model (PTM) at signalized intersections. The model developed in this paper requires one upstream detector, the position of detector is not an issue in this research. According to this model, arrival times and departure times of

all detected vehicles at the stop-line, and also their delays are estimated on the basis of the on-line detector data. The model is developed to represent the individual vehicle motion in relation to the general car-following concept. Thus, it can be applied in the dynamic signal optimization at a microscopic level (Ahn, 2004).
This paper explores a microscopic traffic model based on the one dimensional Kinematic equations in physics. The motion of vehicles from the upstream detector to the downstream stop-line is formulated analytically as a function of the start of green time. The formulae are derived for two different vehicle groups: a motion of the leading vehicle is determined by the signal indication and then successive following vehicles are estimated using the behavior of the vehicles in front of them. A comparative evaluation in delay and sensitivity of delay difference between the LTM and PTM is presented. The results show that the delay estimate in the LTM model is always greater than or equal to the PTM; however, the sensitivity of delay in the PTM is greater than the LTM.

2. Kinematic equations in physics

Kinematics is the study of motion irrespective of the forces; it deals with the mathematical description of motion in terms of position, speed, acceleration (or braking) and time. If any three of those variables are known, then the fourth variable can be calculated by using Kinematic equations. According to these concepts, we can describe the motion of vehicles in the vicinity of signalized intersections. If a vehicle is moving, the speed v is defined as the displacement of the vehicle divided by the time over which the displacement occurs. Furthermore, acceleration rate a refers to the rate of change of speed over time, which is defined as the change of speed divided by the change of time. The acceleration is equal to the second derivative x of with respect to time t:

$$a = \frac{dv}{dt} = \frac{d^2x}{dt^2} \tag{1}$$

The simplest case of translational motion assumes that the acceleration is constant, which means all vehicles are moving with maximum uniform acceleration rate. By using the Equation 1, the speed equation as a function of time, the position equation as a function of speed, and the position equation as a function of time are obtained as follows:

$$v_f = v_i + a(t_f - t_i) \tag{2}$$

$$x_f = x_i + \frac{v_f^2 - v_i^2}{2a} \tag{3}$$

$$x_f = x_i + v_i(t_f - t_i) + \frac{a}{2}(t_f - t_i)^2 \tag{4}$$

where v_i is the initial speed, v_f is the final speed, x_i is the initial position, and x_f is the final position.

3. Notation

The following notation will be used for the trajectory of the leading vehicle n =1 and the following vehicles n ($2 \le n \le N$), where N is the serial number of the most recent detected vehicle at the position of detector. Let

a be the acceleration rate (a >0)
b be the braking rate (b >0)
v_0 be the free-flow speed
t_g be the start of green time (including the reaction time)
v_g be the speed of leading vehicle n =1 at time t_g
X_g be the position of leading vehicle n =1 at time t_g
X_d be the position of detector ($X_d \le 0$)
X_s be the position of stop-line (X_s = 0)
$\overline{X}_v$ be the position at which any delayed vehicles can regain the free-flow speed ($\overline{X}_v = v_0^2 / 2a$)
$\overline{X}_b$ be the maximum boundary of upstream ($\overline{X}_b = X_d - v_0^2 / 2b$)
t_d^n be the time at which the vehicle n is detected at position X_d
v_d^n be the speed of the vehicle n at time which it detected at position X_d
X_b^n be the position at which the vehicle n starts to brake ($X_b^n = X_b^{n-1} - L$, where L is the safety margin)
t_b^n be the time at which the vehicle n starts to brake
v_b^n be the speed of the vehicle n at position X_b^n ($v_b^n = v_0$)
t_a^n be the time at which the vehicle n starts to accelerate ($2 \le n \le N$)
v_a^n be the speed of the vehicle n at time which the acceleration starts($2 \le n \le N$)
X_a^n be the position of vehicle n starts to accelerate($2 \le n \le N$)
X_q^n be the position at which the vehicle n stops completely after a full braking
t_q^n be the time at which the vehicle n stops after a full braking at position X_q^n
X_v^n be the position at which the vehicle n regains the free-flow speed
t_v^n be the time at which the vehicle n regains the free-flow speed at position X_v^n
t_s^n be the time at which the vehicle n crosses the stop-line X_s
v_s^n be the speed of the vehicle n at time which it crosses the stop-line X_s

4. Polynomial Traffic Model (PTM) development

This model assumes a constant acceleration rate and braking rate, no overtaking is allowed and no vehicle exceeds the free-flow speed. Here, the start of green time t_g is defined as the beginning of the green time plus a reaction time τ , Gipps (1981) used τ = 2/3 seconds

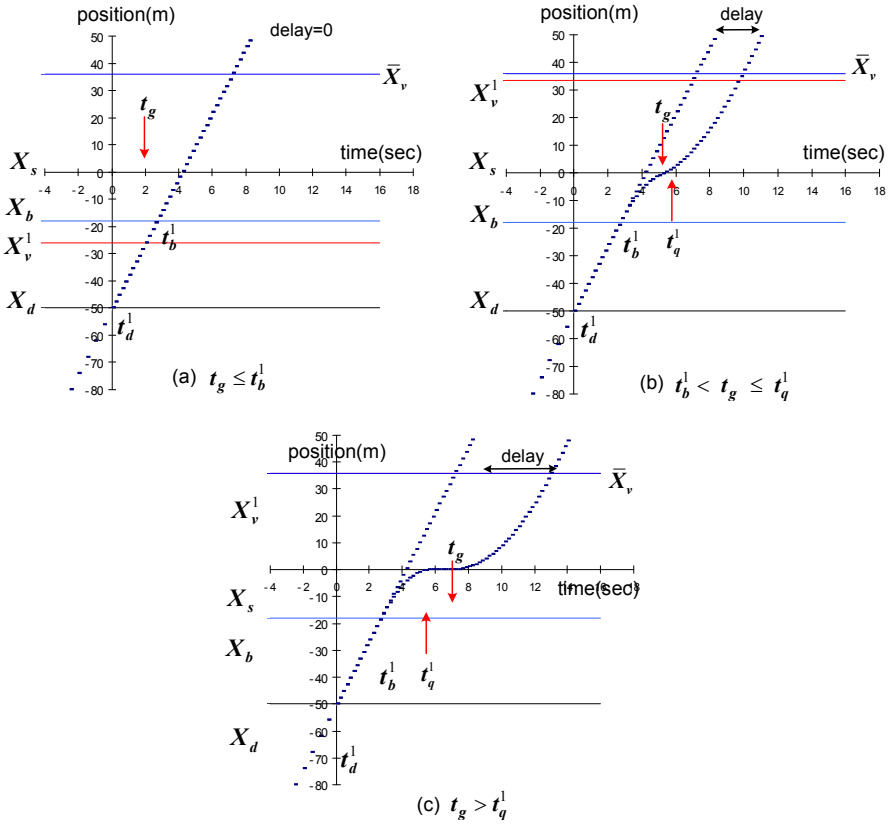


Fig. 1. Trajectory and delay for the leading vehicle in varying start of green time: Polynomial Traffic Model (PTM)

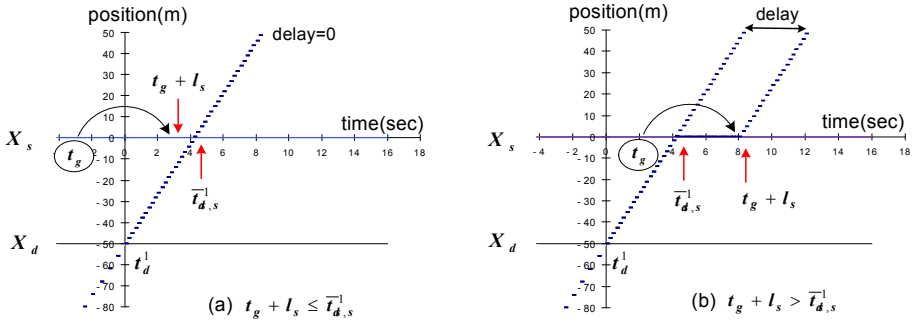


Fig. 2. Trajectory and delay for the leading vehicle in varying start of green time: Linear Traffic Model (LTM)

4.1 Trajectory of the leading vehicle

As can see in Figure 1, the motion of the leading vehicle is affected by the current signal indication. In respect of the start of green time t_g, the motion can be classified broadly into four regions: free-flow, braking (or braking and stopping), acceleration and free-flow. If current signal is green, the leading vehicle will cross the detector with free-flow speed and then reaches stop-line without experiencing any delays. However, if the signal is currently red, the vehicle from the detector can travel up to the braking position with free-flow speed, and on reaching that point the vehicle has to start to brake in order to stop safely at the stop-line. Meanwhile, if the signal changes to green, the vehicle will start to accelerate until it regains the free-flow speed; otherwise, the vehicle has to go through stopping until the next green starts. In this way, we can identify that the vehicle will pass the stop-line either at the free-flow speed or under. For the leading vehicle n =1, three different trajectories are considered as follows:

If $t_g \le t_b^n$: maintain free-flow speed (see Figure 1a)

If $t_b^n < t_g \le t_q^n$: free-flow → braking → acceleration → free-flow (see Figure 1b)

If $t_g > t_q^n$: free-flow → braking → stopping → acceleration → free-flow(see Figure 1c).

The motion of the leading vehicle from the detector position up to the braking position is unaffected by the current signal indication, thus it maintains free-flow speed. For the leading vehicle n =1, the braking position X_b^n , the braking time t_b^n and the stopping time t_q^n are calculated as follows:

$$X_b^n = -\frac{v_0^2}{2b},\ t_b^n = t_d^n - \frac{v_0}{2b} - \frac{X_d}{v_0} \text{ and}$$

$$t_q^n = t_b^n + \Delta t_b^n = t_d^n + \frac{v_0}{2b} - \frac{X_d}{v_0} \ (\text{where } \Delta t_b^n = v_0 / b) \tag{5}$$

When the vehicle has reached its braking position, its further motion is determined by the current signal indication. There are three different trajectories to be considered with respect to the start of green time t_g for the leading vehicle n =1:

If $t_g \le t_b^n$ then

$$v_g = v_0,\ X_g = X_d + v_0(t_g - t_d^n),$$
$$t_v^n = t_g \text{ and } X_v^n = X_g \tag{6}$$

If $t_b^n < t_g \le t_q^n$ then

$$v_g = v_0 - b(t_g - t_b^n),\ X_g = X_b^n + v_0(t_g - t_b^n) - \frac{b}{2}(t_g - t_b^n)^2,$$

$$t_v^n = t_g + \frac{b}{a}(t_g - t_b^n) \text{ and } X_v^n = X_b^n + v_0(\frac{a+b}{a})(t_g - t_b^n) - (\frac{ab+b^2}{2a})(t_g - t_b^n)^2 \tag{7}$$

If t_g starts after the vehicle has stopped ($t_g > t_q^n$) then

$$v_g = 0,\ X_g = 0,\ t_v^n = t_g + \frac{v_0}{a} \text{ and } X_v^n = \frac{v_0^2}{2a} \tag{8}$$

Finally, for the leading vehicle n =1, the crossing time t_s^n and its speed v_s^n at the stop-line can be obtained by comparing the fixed position X_s and the varying position X_v^n :

If $X_v^n \le X_s$, which means that the vehicle is crossing the stop-line with free-flow speed, then

$$t_s^n = t_v^n - \frac{X_v^n}{v_0} \text{ and } v_s^n = v_0 \tag{9}$$

If $X_v^n > X_s$, which means that the vehicle is crossing the stop-line during the acceleration, then

$$v_s^n = \sqrt{v_g^2 - 2aX_g} \text{ and } t_s^n = t_g + \frac{\sqrt{v_g^2 - 2aX_g} - v_g}{a} \tag{10}$$

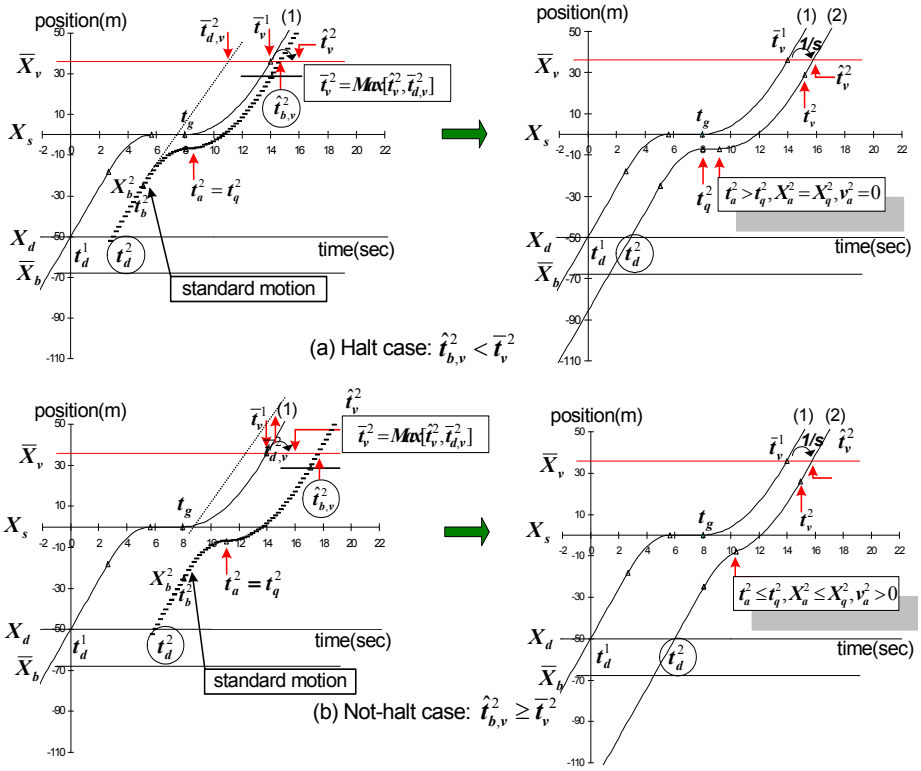


Fig. 3. A standard motion test for the following vehicles

4.2 Trajectory of the following vehicles

The variables estimated for the leading vehicle are used as parameters for the following vehicles trajectory in the following section. The basic concepts we use in the following calculations are the following vehicles cannot depart the downstream free-flow

position $\bar{X}_v$ $\bar{X}_v$ with less than the minimum headways. Once we have characterised the full trajectory of the leading vehicle n =1 as a function of the start of green time t_g , the trajectory of all successive following vehicles n =2, 3, … N can be calculated directly based on the motion in front.

As can see in Fig 3, when the following vehicle n =2 crosses the detector at time t_d^n , the first order of task is finding the possible departure time $\bar{t}_v^n = \text{Max}[\hat{t}_v^n, \bar{t}_{d,v}^n]$ at position $\bar{X}_v$, here $\hat{t}_v^n = \bar{t}_v^{n-1} + 1/s$ (where s is a saturation flow) is the earliest departure time at position $\bar{X}_v$ and $\bar{t}_{d,v}^n = t_d^n + (\bar{X}_v - X_d)/v_0$ is the free-flow travel time from the detector to the position $\bar{X}_v$. By comparing variables $\hat{t}_v^n$ and $\bar{t}_{d,v}^n$, we can decide whether or not the following vehicle will be delayed. If $\bar{t}_{d,v}^n \geq \bar{t}_v^n$, the detected vehicle is identified as undelayed so it can reach the position $\bar{X}_v$ $\bar{X}_v$ from the detector with free-flow speed. Thus, it is not necessary to find acceleration variables. However, if $\bar{t}_{d,v}^n < \bar{t}_v^n$, the vehicle is identified as delayed, then we need standard motion test to find acceleration variables, in which we can test whether or not any stopped lost time due to a queue is involved. Here, the standard arrival time is the longest approach time from the braking position X_b^n to the position $\bar{X}_v$ supposing that the vehicle has not stopped. The braking position is calculated by of adding minimum safe spacing L . The final information we are seeking for each following vehicle n ($2 \leq n \leq N$) is its crossing time t_s^n , speed v_s^n at stop-line X_s and departure time $\bar{t}_v^n$ at position $\bar{X}_v$: the time t_s^n will be used to estimate the number of vehicles that can pass the stop-line if the green is extended by a certain control decision time, and $\bar{t}_v^n$ will be used to estimate its delays. In this analysis, the final information we are seeking for each following vehicle n ($2 \leq n \leq N$) is its crossing time t_s^n , speed v_s^n at stop-line X_s and departure time $\bar{t}_v^n$ at position $\bar{X}_v$: the time t_s^n will be used to estimate the number of vehicles that can pass the stop-line if the green is extended by a certain control decision time, and $\bar{t}_v^n$ will be used to estimate its delays. The calculation algorithm for the following vehicles trajectory is as follows:

Step 0: (Leading vehicle trajectory)
Identify the leading vehicle trajectory n =1 with respect to the current signal indication, and all identified variables are used as parameters in the following steps
$\Rightarrow$ if N =1, stop processing (no following vehicles)

Step 1: (Departure time at $\bar{X}_v$)
Find the earliest departure time $\hat{t}_v^n$ and free-flow arrival time $\bar{t}_{d,v}^n$ at position $\bar{X}_v$, then the departure time will be $\bar{t}_v^n = \text{Max}[\hat{t}_v^n, \bar{t}_{d,v}^n]$, where $\hat{t}_v^n = \bar{t}_v^{n-1} + 1/s$ and $\bar{t}_{d,v}^n = t_d^n + (\bar{X}_v - X_d)/v_0$

Step 2: (Motion definition for undelayed case or delayed case)
Find the expected motion of the each following vehicle:
2-1) If $\bar{t}_{d,v}^n \geq \hat{t}_v^n$ which corresponds to undelayed motion, then standard motion test is not required

2-2) If $\overline{t}_{d,v}^{n} < \hat{t}_{v}^{n}$ which corresponds to delayed motion, then standard motion test is required

Step 3: (Braking position and time)

$X_b^n = X_b^{n-1} - L$ (t_b^n is only possible to find when $X_b^n \geq \overline{X}_b$; otherwise, out of range)

From Step 2, if the motion is identified as an undelayed case and:

If ($X_b^n \geq \overline{X}_b$), then $t_b^n = t_d^n + (X_b^n - X_d)/v_0$,

Else, t_b^n is unknown

$\Rightarrow$ **go to Step 4** to find the variable of crossing time to the stop-line.

From Step 2, if the motion is identified as a delayed case:

If $X_d \leq X_b^n < X_s$, then $t_b^n = t_d^n + (X_b^n - X_d)/v_0$

If $\overline{X}_b \leq X_b^n < X_d$, then $t_b^n = t_d^n - (v_0 - \sqrt{v_0^2 + 2b(X_b^n - X_d)})/b$

$\Rightarrow$ **go to Step 5** to find additional variables.

Step 4: (Crossing time to the stop-line for undelayed vehicle)

The crossing time t_s^n at stop-line X_s can be calculated by using a free-flow motion equation, that is $t_s^n = t_d^n + (X_s - X_d)/v_0$

$\Rightarrow$ if $n < N$, **go to Step 1**; otherwise, stop processing

Step 5: (Standard motion test for delayed vehicle: halt case or not-hale case)

The standard motion test is only necessary if the vehicle is identified as delayed and $X_b^n \geq \overline{X}_b$

From the braking position standard motion test is needed to find whether or not it will come to a halt

5-1) The standard motion arrival time, that is $\hat{t}_{b,v}^{n} = t_b^n + v_0(a+b)/2ab + (\overline{X}_v - X_b^n)/v_0$

5-2) Motion definition: halt-case or not-halt case

If $\hat{t}_{b,v}^{n} < \overline{t}_v^n$, the motion is identified as a halt-case, then the stopping is involved

If $\hat{t}_{b,v}^{n} \geq \overline{t}_v^n$, the motion is identified as a not-halt case, then the stopping is not involved

Step 6: (Acceleration position, time and speed variables for delayed vehicle)

6-1) For a halt-case, acceleration position is equal to the stopped position, then

$X_a^n = X_q^n$ (where $X_q^n = X_b^n + v_0^2/2b$), v_a^n =0 and $t_a^n = \overline{t}_v^n - \frac{v_0}{2a} - \frac{(\overline{X}_v - X_a^n)}{v_0}$

6-2) For a not-halt case, acceleration starts meanwhile of braking, then

$t_a^n = t_b^n + \sqrt{\frac{(X_b^n - \overline{X}_v) + v_0(\overline{t}_v^n - t_b^n)}{(ab+b^2)/2a}}$, $X_a^n = X_b^n + v_0(t_a^n - t_b^n) - \frac{b}{2}(t_a^n - t_b^n)^2$ and $v_a^n = v_0 - b(t_a^n - t_b^n)$

Step 7: (Position and time of regaining the free-flow speed for delayed vehicle)

If $X_b^n \geq \overline{X}_b$, the time t_v^n and the position X_v^n can be calculated by using acceleration variables; otherwise, they can be calculated corresponding to the departure time $\overline{t}_v^n$ at $\overline{X}_v$

7-1) If ($X_b^n \geq \overline{X}_b$), $t_v^n = t_a^n + (v_0 - v_a^n)/a$ and $X_v^n = X_a^n + v_a^n(t_v^n - t_a^n) + \frac{a}{2}(t_v^n - t_a^n)^2$

7-2) If ($X_b^n < \overline{X}_b$) which is defined as a out of boundary case, then

$$t_v^n = t_d^n + \sqrt{\frac{2\left[(X_d - \bar{X}_v) + v_0(\bar{t}_v^n - t_d^n)\right]}{a}} \quad \text{and} \quad X_v^n = X_d + v_d^n(t_v^n - t_d^n) + \frac{a}{2}(t_v^n - t_d^n)^2$$

where the estimated speed of $v_d^n = v_0 - a(t_v^n - t_d^n)$

Step 8: (Crossing time and speed to the stop-line for delayed vehicle)

The departure time t_s^n at the stop-line X_s can be calculated by comparing position variables

8-1) If ($X_b^n \geq \bar{X}_b$) and

If $X_v^n \leq X_s$, then $t_s^n = t_v^n - (X_v^n / v_0)$

If $X_v^n > X_s$, then $t_s^n = t_a^n + (v_s^n - v_a^n)/a$,

where $v_s^n = \sqrt{(v_a^n)^2 - 2aX_a^n}$

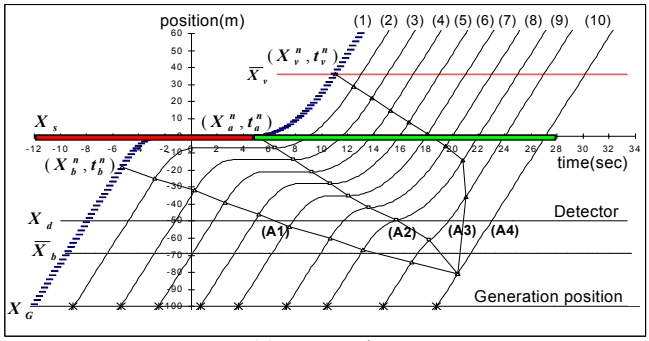
8-2) If ($X_b^n < \bar{X}_b$) and

If $X_v^n \leq X_s$, then $t_s^n = t_v^n - (X_v^n / v_0)$

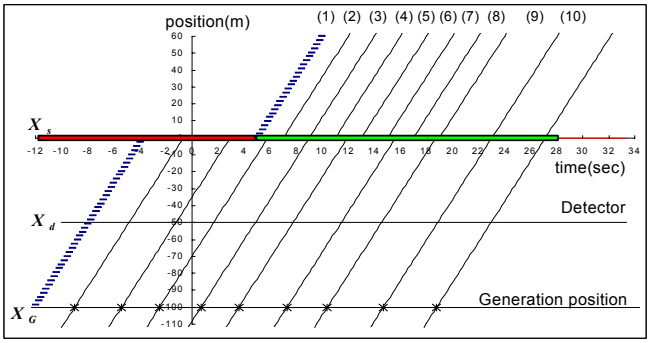
If $X_v^n > X_s$, then $t_s^n = t_v^n + (v_0 - v_s^n)/a$,

where $v_s^n = \sqrt{(v_d^n)^2 - 2aX_d}$

$\Rightarrow$ if $n < N$, **go to Step 1**; otherwise, stop processing.



(a) PTM trajectory



(b) LTM trajectory

Fig. 4. Trajectory of the following vehicles (PTM vs LTM)

As can see in Figure 4a, the PTM can provide plenty of information around the detector and the stop-line. There are four different sets of motions have been identified at the position of the detector during one green period. In respect of the start of green time, any detected vehicles until the time A1 have crossed the detector with free-flow speed, the times between A1 and A2 vehicles have crossed it while braking, the times between A2 and A3 vehicles have crossed it while accelerating, and any following vehicles after the time A4 will cross the detector with free-flow speed. Presumably, the time A4 is the queue (or delay) dissipation time of this stage. More detailed analysis is suggested by Ahn (2004).

5. Delay and sensitivity of delay

In this section, delay and sensitivity of delay difference between the LTM and the PTM is discussed in relations to varying start of green time t_g. The delay estimates in the LTM and the PTM differs around the time of green starts; the LTM delay is always greater than or equal to the PTM, but so long as there is a queue the delay estimates are identical; however, the maximum sensitivity of delay in the PTM traffic model is higher than that in the LTM.

5.1 Delay and sensitivity of delay for the leading vehicle

As can see in Figure 2, the delay in the LTM is identified as the time difference between the start of effective green time $t_g + l_s$ (where start lag $l_s = v_0 / 2a$) and its free-flow arrival time to the stop-line $\overline{t}_{d,s}^n = t_{d,s}^n - X_d / v_0$. The LTM delay for the leading vehicle n =1 with respect to the time t_g is expressed as $D_V^n(t_g)$, which are calculated as follows:

If ($t_g + l_s \le \overline{t}_{d,s}^n$) then

$$D_V^n(t_g) = 0 \tag{11}$$

If ($t_g + l_s > \overline{t}_{d,s}^n$) then

$$D_V^n(t_g) = (t_g + l_s) - \overline{t}_{d,s}^n \tag{12}$$

If t_g starts before the free-flow arrival time minus the start lag, no delay is incurred in LTM, and the resulting sensitivity is 0. However, from that time if t_g is increased by an amount ε, delay is increased by an identical amount ε, so that the resulting sensitivity is 1. By differentiating the delay Equation (11 and 12) with respect to the time t_g, we can get the sensitivity of delay as follows:

$$D_V^{1\prime}(t_g) = \begin{cases} 0 & t_g \le \overline{t}_{d,s}^n - l_s \\ 1 & t_g > \overline{t}_{d,s}^n - l_s \end{cases} \tag{13}$$

As seen in Figure 1, the delay for the PTM is calculated on the basis of the time t_g in accordance with vehicular characteristic variables, such as, acceleration rate, braking rate and free-flow speed. The delay for the leading vehicle n =1 with respect to the time t_g is expressed as $D_K^n(t_g)$, which are calculated as follows:

If $t_g \le t_b^n$ then

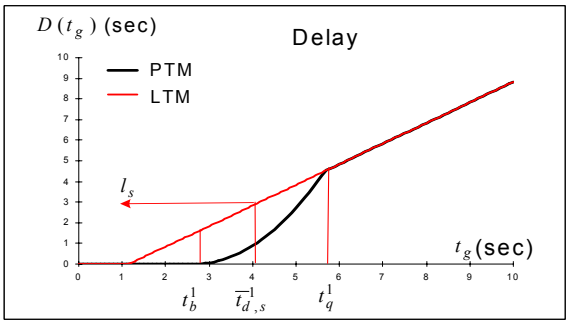
$$D_K^n(t_g) = 0 \tag{14}$$

If $t_b^n < t_g \le t_q^n$ then

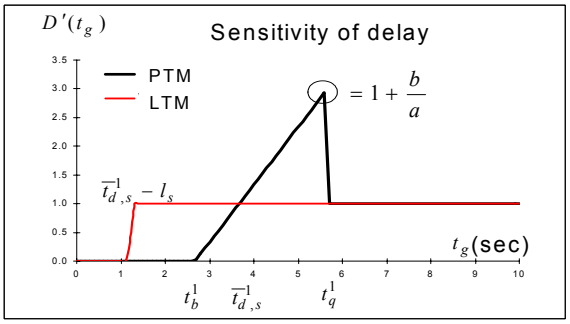
$$D_K^n(t_g) = (\bar{t}_b^n - \bar{t}_d^n) - \frac{(X_b^n - X_d)}{v_0} + \frac{(ab + b^2)}{2av_0}(t_g - \bar{t}_b^n)^2 \tag{15}$$

If $t_g > t_q^n$ then

$$D_K^n(t_g) = (t_g - \bar{t}_d^n) + \frac{v_0}{2a} + \frac{X_d}{v_0} \tag{16}$$



(a) Delay



(b) Sensitivity of delay

Fig. 5. Delay and sensitivity of delay comparison for the leading vehicle

As can see in Figure 5, if t_g starts before braking, no delay is incurred and the resulting sensitivity is 0. If t_g starts between braking and stopping, the delay increases quadratically, and the resulting sensitivity increases linearly. Once the vehicle has stopped, if t_g is increased by an amount ε , delay is increased by an identical amount ε , so that the resulting sensitivity is 1. By differentiating the delay Equations (14, 15 and 16) with respect to the varying time t_g, we can get the sensitivity of delay as follows:

If $t_g \le t_b^n$ then

$$D_K^{n\prime}(t_g) = 0 \tag{17}$$

If $t_b^n < t_g \le t_q^n$ then

$$D_K^{n\prime}(t_g) = \frac{(ab + b^2)}{av_0}(t_g - t_b^n) \tag{18}$$

If $t_g > t_q^n$ then

$$D_K^{n\prime}(t_g) = 1 \tag{19}$$

The maximum sensitivity of delay for the leading vehicle n =1 in the LTM is always 1. In contrast, the maximum sensitivity of delay in the PTM is $1+(b/a)$ that is obtained by differentiating the t_v^n in Equation (7) that obtained when $t_g = t_q^1$.

5.2 Delay and sensitivity of delay for the following vehicles

In this section, the following results are based on the simulation. Two different cases of traffic are considered: a low density case and a high density case. In the present examples, the maximum number of vehicles that can be held in the queue with this given condition is 8-vehicle.

		vhicle n =							
		1	2	3	4	5	6	7	8
Low density α =0.5 h_0 =3.0	u	-	0.613	0.321	0.824	0.569	0.851	0.312	0.67
	H	-	3.97	5.27	3.38	4.12	3.32	5.33	3.79
	t_d^n	0.0	4.0	9.3	12.7	16.8	20.1	25.4	29.2
High density α =0.9 h_0 =2.0	u	-	0.613	0.321	0.824	0.569	0.851	0.311	0.671
	H	-	2.54	3.26	2.21	2.62	2.18	3.29	2.44
	t_d^n	0.0	2.5	5.8	8.0	10.6	12.8	16.1	18.5

Table 1. Vehicle generation by using shifted exponential distribution of headways

In this The detector is located 50m away from the stop-line, and the minimum spacing of each following vehicle is L =7m, saturation departure time is 1.8 sec/vehicle (2,000 veh/h) and two cases of vehicles are generated based on *shifted exponential distribution of headways*, which is given by

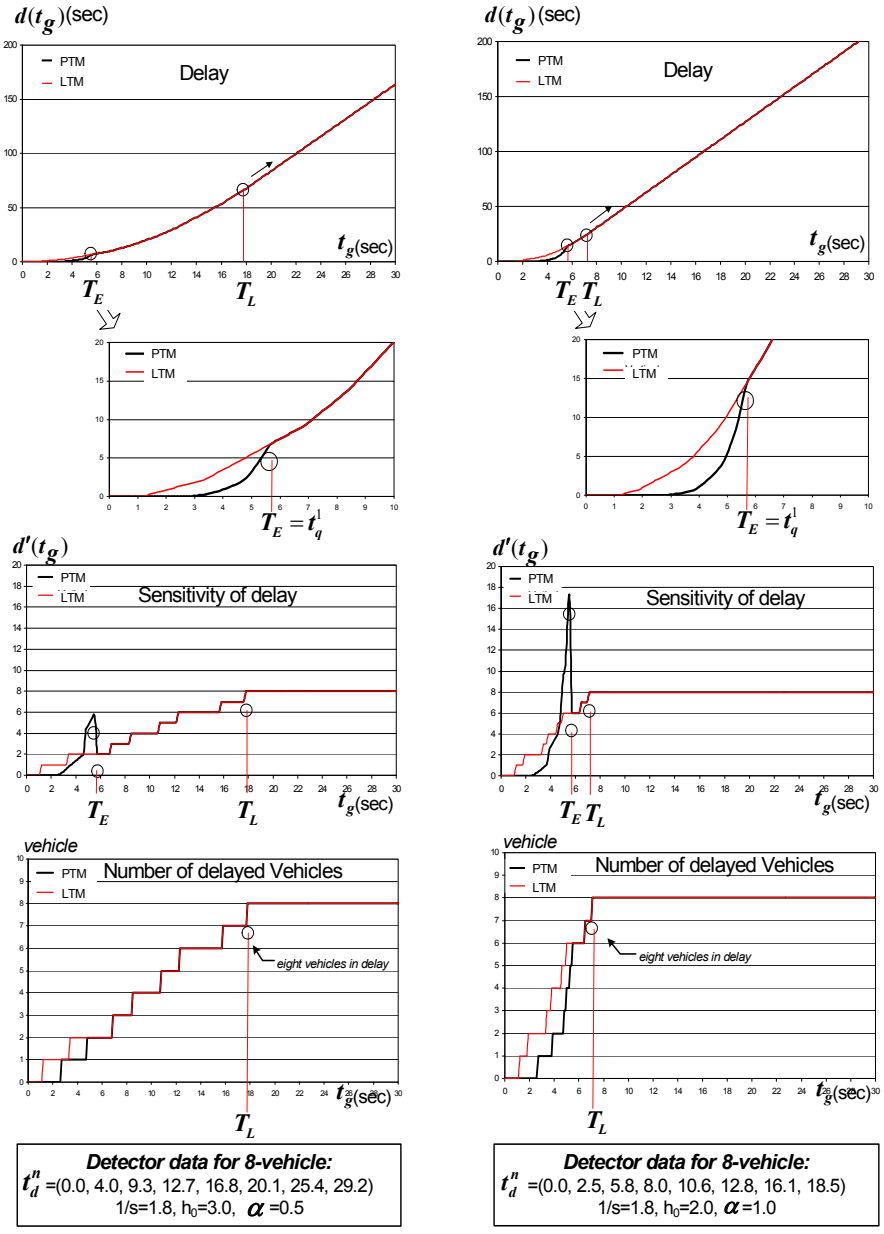
$$H = h_0 - \frac{1}{\alpha}\ln(u) \tag{20}$$

where

u is the random value generation, in which variables are generated with equal probability between [0, 1]

h_0 is the minimum gap of following ($h_0 \ge \tau + L/v_0$, where τ is 0.67sec)

α is the density parameter



(a) Low density case (b) High density case

Fig. 6. Delay and sensitivity of delay comparison for the following vehicles

Using Equation (20), we can generate the vehicles without causing any headway violation. The detection time t_d^n can be given by $t_d^n = t_d^{n-1} + H$ ($n \geq 2$). Based on Table 1 data, the sensitivity of delay for 8-vehicle is tested with respect to the variations in the start of green time: in the range of t_g = 0 ~ 30 sec and t_g is incremented by 0.1 sec. As can see in Figure 6, let T_E be the time at which the total delay of 8-vehicle for the PTM and the LTM become same, namely $T_E = t_q^1$, where T_L is the time at which total delay becomes increasing linearly.

If t_g starts before T_L, which means that all 8-vehicle have not delayed yet; in this range, we can suppose that some vehicles may go through the stop-line without experiencing any delay, thus the sensitivity fluctuates.

If t_g starts after T_L, which means that all 8-vehicle have been delayed, the sensitivity of delay in this range with respect to t_g is equal to the total number of delayed vehicles. The sensitivity of delay for the LTM is equal to the number of delayed vehicles in the time range. But it differs for the PTM, if t_g starts before T_E, in some time range it shows a sensitivity higher than the total number of delayed vehicles. For the high density case simulation (see Figure 6b), the LTM delay is greater than or equal to the PTM delay. T_E =5.7 sec and T_L =7.1 sec. At time t_g = 5.5 sec, the PTM shows a sensitivity of delay $D_K^{'}$ = 17.28, and the LTM shows $D_V^{'}$ = 6, which is equivalent to numbers of delayed vehicles. At this time, the sensitivity of delay for the PTM is about three times greater than for the LTM. At time t_g = 7.1 sec, the sensitivity of delay for both models is equal to 8. From that time, we can see that all vehicles will experience some delay.

6. Conclusions

As explored in this paper, the delay estimated in the Linear Traffic Model (LTM) and the Polynomial Traffic Model (PTM) differs around the time of the green start, but so long as there is a queue the delay estimates from these models are identical. With respect to variations in the start of green time and provided that the braking rate exceeds the acceleration rate, the delay estimated in the LTM is always greater than or equal to the PTM traffic model; however, the maximum sensitivity of delay in the PTM is greater than that in the LTM. The PTM proposed in this paper has some attractions compared to the simpler LTM. The delay estimate in the PTM is on the basis of the vehicular characteristics, such as vehicle length, acceleration rate, braking rate and free-flow speed; however, the LTM takes adjusted time parameter of the start of effective green time. Thus, the delay estimated from the LTM is not realistic and always greater than or equal to the PTM. From these results, it is clear that the leading vehicle trajectory is the most important determinant for estimating the motion of vehicles at signalized intersections. Namely, the delay can be minimized in signal operation, if the start of green time begins before the vehicle starts to brake. The work described so far in this study is only at on early stage, in the sense that more work should be done for estimating the motion of vehicles in various intersection configurations.

7. References

Ahn, W-Y. (2004). Dynamic signal optimisation for isolated road junctions. Ph.D. thesis, University College London.

Akcelick, R. (1981). Traffic signals: Capacity and timing analysis. *Research Report ARR No. 123*. ARRB Transport research Ltd, Vwemont South Australia.

Akcelick, R. & Besley, M. (2002). Queue Discharge flow and speed models for signalised intersections. *Transport and Traffic Theory in the 21st Century*, theory, pp. 99-118.

Allsop, R.E. (1970). Optimisation techniques for reducing delay to traffic in signalised road junction. Ph.D. thesis, University College London.

Clayton, A.J.H. (1940). Road traffic calculation. *J. Inst. Civil Engineering*. 16 (7), pp. 247-284.

Gartner, N.H. (1983). OPAC: A demand-responsive strategy for traffic signal control. TRB, *Transportation Research Record 906*, pp. 75-84.

Gipps, P.G. (1981). A behavioural ca-following model for computer simulation. *Transportation Research*, 15(2), pp. 105-111.

Heydecker, B.G. (1990). A continuous-time formulation for traffic-responsive signal control. *Proceedings of the 11th International Symposium on Transportation and Traffic Theory*, Yokohama, pp. 599-618.

Miller, A. J. (1963). A computer system for traffic networks. *Proceedings of the Second International Symposium on the Theory of Road traffic Flow* (ed. Almond, J.), OECD, Paris, pp. 200-220.

Robertson, D.I. (1969). TRANSYT: A traffic network study tool. *RRL Report, LR253.*

TRANSYT (2000). A traffic network study, McTrans.

TRB (2000). Highway Capacity Manual. *Transportation Research Board*, National Research Council, Washington, D.C., U.S.A. (*HCM 2000*).

Webster, F.V. and Cobbe, B.M. (1966). *Traffic signals*. HMSO, London.

20

Collaborative Negotiation to Resolve Conflicts among Replicas in Data Grids

Ghalem Belalem and Belabbes Yagoubi
Department of Computer Science, Faculty of Sciences, University of Oran
Algeria

1. Introduction

In Data Grids, data replication is a fundamental mechanism widely used to increase data availability, improve performance, balancing the load among nodes of the grid and support the fault tolerance components. The replication management and its implementation are not simple tasks and produce other problems, like consistency management of replicas. One of the concerns major in the consistency management approaches called optimistic, it is the conflicts resolution among replicas. One of the main problems encountered in the use of replication, the management of consistency among the replicas. Replication techniques are used to provide multiple critical copies and to maintain them. In coherent state, they improve the overall system availability and performance.
The main objective of a replica consistency approach is to avoid or even reduce the inconsistency between replicated data. Many current applications can barely tolerate a certain degree of contradiction between replicas where the strong consistency is not a condition, for examples in the approximate readings from meteorological sensors often suffice when performing predictive modelling of weather conditions, the network security applications or in video conferencing applications.
Our contribution consists of the proposal for a model, for consistency management of replicas in data grids, which combines at the same time the pessimistic approaches, which support the quality of service (QoS), and the optimistic approaches which are focused on the improvement performance. Our effort in this contribution, aims to resolve conflicts between the replicas by using a collaborative negotiation between representatives of the nodes in the data grid. We try to show in this work that trading is influenced by the load balancing of the grid.
Thereafter, we organize our paper as follows: Section 2 gives an overview some work for the resolution of conflicts between replicas, section 3 will be reserved for a brief description of the management techniques used consistency of replicas, the Section 4 presents our approach to consistency management approach that combines optimistic and upbeat and then we propose in Section 5, a process for resolving conflicts encountered between replicas is based on economic models of the real market. Section 6 allows us to study the influence of load balancing on quality replicas. The experimental study of our approach is presented in Section 7, we compare the behaviour of our proposal in comparison with conventional

techniques, in a first phase, we try to show that our approach reduces significantly the divergences and conflicts, in a second part we will present the effect of balancing the quality of replicas. We conclude this paper with a summary and a set of perspectives.

2. Related work

One of the problems hard in the consistency management is the choice of the technique of reconciliation to make converge the divergent replicas. The reconciliation is the activity to detect, control and solve conflicts, in order to converge towards the same state. Several work proposed different techniques to produce a new coherent value of the counterparts of the same object. Coda (Kistler and Satyanarayanan 1992) was among the first hoarding systems, using user profiles to decide what to hoard and requiring user intervention for conflict resolution. In the system Bayou (Petersen et al. 1996) conceived for the collaborative applications in the mobile environments and using the optimistic replication to weak coherence, the coherence of the replicas east guarantees thanks to an epidemic diffusion of the updates between the waiters. If a conflict emerges the waiters in Bayou proceed by fusion of the requests of the writings according to a quite selected scheduling. In the system IceCube (Kermarrec et al. 2001) uses the reconciliation based on semantic specificities of the application and the intention of the user. He amalgamates the newspapers compared to recorded operations and of which the goal to minimize the conflicts. In the work presented in (Molli et al. 2002; Vidot et al. 2000), authors use the technology of the operational transforms by exploiting the semantic properties of the operations such as causality for serializability in order to lead to the coherence of a divided object. Instead, the work presented here is based on less constraining assumptions about the semantic of data and thus their ability to reconcile inconsistent data.

3. Replication and consistency management

Replication techniques are used to provide multiple critical copies and to maintain them. In coherent state, they improve the overall system availability and performance. In splitting of replication advantages, there are many problems that must be resolve like (Belalem & Bouhraoua, 2007; Belalem & Slimani, 2007; Gray et al., 1996; Xu et al., 2002):

How do we select and estimate the metrics for taking replication decisions?
When do we replicate a given object?
Where do we place the replicas of a given object?
How do we ensure consistency of all replicas of the same object?
How do we route client requests to appropriate replicas?

Among these problems, the main critical concerns the consistency problem that needs to maintain the data consistency between a set of replicated data distributed among a set of computer (Saito & Shapiro, 2005; Goel et al., 2005; Cameron et al. 2004). The main objective of a replica consistency approach is to avoid or even reduce the inconsistency between replicated data. Many current applications can barely tolerate a certain degree of contradiction between replicas where the strong consistency is not a condition, for examples in the approximate readings from meteorological sensors often suffice when performing

predictive modelling of weather conditions, the network security applications or in video conferencing applications (Belalem & Slimani, 2007; Dorcey, 1995).

3.1 Pessimistic approach

The Pessimistic approach prohibited any access to a replica unless it is provably up to date. This makes the users believe that they have only one consistent copy. The main advantage of this approach is that all replicas converge at the same time, a fact which permits to guarantee a high consistency. Hence, any problem of divergence could be avoided. This type of approach is well adapted to small and middle scale systems but becomes very complex when adapted to large scale systems. Thus, we can raise three major drawbacks of this approach (Saito & Shapiro, 2005):
it is very badly adapted to uncertain or unsteady environments, such as the mobile systems, or grids with high rate of change;
it cannot bear the updating cost when the degree of replication is very high.

3.2 Optimistic approach

Unlike the pessimistic approach, this approach allows acceding to any replica and at any time. In this way, it is then possible to reach any replica even if it is not necessarily coherent. So, this approach tolerates a certain divergence among replicas. On the other hand, it requires a phase to detect divergence among replicas then a phase to correct this divergence by converging the replicas toward a coherent state. Although it does not guarantee a high consistency as in the pessimistic case, it has, however, a certain number of advantages. They are mainly summarized as follows (Olston & Widom, 2005; Saito & Shapiro, 2005):

- since it admits a divergence among replicas, this approach can be adapted to uncertain or mobile environments;
- it does not block the accesses to replicas;
- it allows the passage to scale by accepting a high number of replicas.

4. Consistency management of replicas in Data Grid

Our main contribution in this present work is to propose a service of consistency management for replicas in a Data Grid environment. The service combines the pessimistic and optimistic approaches in order to increase service quality replicas and improve the performance of the Data Grid. The service proposed is supported by a model hierarchical and distributed to two levels (see Figure 1), which enables him to adapt to the scalability, to be flexible and to allow the tolerance of certain faults. We consider a grid as a collection of distributed collections of Computing Elements (CE's) and Storage Elements (SE's). These elements are linked together through a network to form a Site or a Cluster. Sites are in turn linked together to form a grid. Replicas are stored on Storage Elements and are accessible from Computing Elements.

In our work, the data grid consists of two levels:
- Level 0: In this level, the management of replicas of data within a site is performed using one or more replicas managers (RM) as replication strategies (cite!!!). Two strategies can be

used: the replication strategy type uni-master where the master is the replica manager (RM) and the multi-masters where the site is managed by several managers replicas.

- Level 1: This level consists of a set of representatives (virtual nodes). Each representative of the level 1 is a site of level 0. For each site, a representative is elected from the manager replicas.

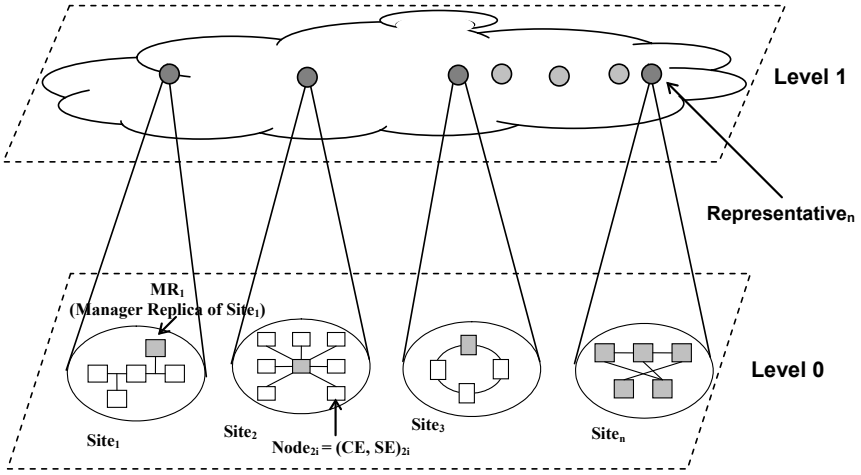


Fig. 1. Proposed Model for Consistency Management

The proposed service for consistency management uses an economic model combined with several models used in the domain of market economy in order to resolve conflicts encountered between replicas. This service of consistency management proposed begins with a preliminary stage which consists in making a pre-processing before the service is started. This preliminary stage is composed of three phases (see Figure 2):

a. The phase of collection of information: to have a life on the status of the site, the manager replicas (MR) collects information on each data (metadata) periodically each node containing replicas of this data. This information can be represented the version of the vector of each replica, the timestamp and the origin of the update,…

b. The phase of information analysis: this step, which is based on the information collected to calculate a set of measures that we used allows, for example, the average standard deviation, distance, the number divergence average per site, the number conflict average per site, the distance a local site,…

c. The phase of decision: Starting metrics generated by the analysis phase, the management service consistency decides or not on the trigger the spread of updates, intra-or inter-site locations.

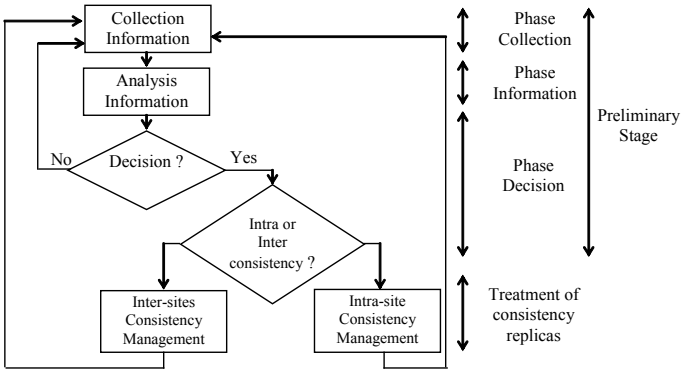


Fig. 2. Preliminary stage of consistency management in Data Grid

4.1 Intra-site consistency

The essential goal of consistency management in the level 0 is to make converging replicas of a data, stored in the same site, towards local reference replica. This service of consistency management, inspired of the optimistic approaches (Belalem & Slimani, 2007; Saito & Shapiro, 2005), supports the response time compared to coherence. On the basis of this objective, it is thus possible that the replicas of a file can diverge. Thus, a user can, at a given moment, observe copies of the same file with different values.

The intra-site process of consistency management is composed mainly of the following tasks:

Treatment of requests: This task has as a role to carry out a request subjected by client;

Propagation: Since we accept the fact that the replicas of a file can diverge, the task of propagation consists in making a transfer of the updates within a site given towards replicas which are not up to date. This propagation ensures, that at the end of a finished time, the replicas will converge towards the local reference replica;

Tolerance with the faults: The purpose of this task is to deal with the failures of nodes of site given and a site to the level of all the grid;

The Model Updating: This task fulfils the functions necessary to the model updating, and in particular the integration of new nodes in a site.

One of two main strategy replications can be chosen for each site to accommodate requests from users:

- Intra-site of consistency management with single-master strategy: The process of intra-site consistency management, with single-master strategy (Goel et al. 2005; Saito & Shapiro, 2005), starts by selecting a replica master among all the replicas stored in the nodes of a given site. The node containing this replica will be called the manager of replicas of a file given inside a site. In the single-master strategy, a request of a client of writing type, as for a given file, should be carried out only on this master, whereas a request of reading can reach any replica of this file.

- Intra-site of consistency management with multi-masters strategy: For intra-site of consistency management with multi-master strategy, a request of a client, type reading or writing, can be carried out on any node containing a replica of the file called upon by the request. When a client subjects a request $Req_i(k)$ from a site S_i using replication multi-masters strategy, then this request will be able to reach any replica p of the object O_k of a node of the site S_i.

We can summarize the various stages of sub-service intra-site consistency by the Algorithm Intra-Site Consistency:

Algorithm 1 Intra-Site Consistency

```
Begin
/ * Is a request arriving at the site controlled by MRl */
Switch (Strategy) Do
Case Single Master: If Master = free Then treatment of the
request of the customer
        Else to deposit the request in the queue of the
        site
Case Multi Master: If ∃Master = free Then treatment of the
request of the customer
        Else to deposit the request in the queue of the
        site
EndSwitch
If Critical_situation(MRi)
        Then Algorithm Inter-sites consistency
            /* Sitei in Crisis   */
                    Else Propagate of updates.
End
```

4.2 Inter-sites consistency

The main objective of inter-sites consistency management, in the level 1, is to converge replicas of the data grid to a reference replica, using economic models of real market (Buyya & Vazhkudai, 2001).
This sub-service management of the consistency between sites using a pessimistic approach, it is composed of the following steps:

1) Selecting a representative: choose a representative of a site among all nodes, or the master or a master among masters of the rest according to a selection (for example, is the reply that suffered more updates, the most popular by a coefficient of reliability of the storage element, etc.).
2) Collection of information: One of the features of the representative is to receive information on the replicas (the meta-data) of the corresponding site through the management of replicas.
3) Collaboration: a representative collaborating with the other representatives of other sites, to converge the replicas of the same data.
4) Detection and resolution of conflicts replicas: If a conflict is detected, then the negotiation process for the resolution of conflict is triggered.

5. Process of conflict resolution

The collaborative relationship is based on the principle of domination replicas (Hasegawa et al. 1998; Ratner et al. 1997). If this dominance is not successful, the process detects conflicts; it is at this moment that the resolution process is started. A second situation that can trigger this process of resolution is that the analysis representing the state of its corresponding site through managers' replicas, if the replicas of the same data are too different from a threshold of tolerance, site is considered as being in crisis.
We say a site is in a critical situation if its versions replicas of the same object are too dispersed. In this case, it is called a site in crisis. The main stages of this consistency are represented in Figure 3:

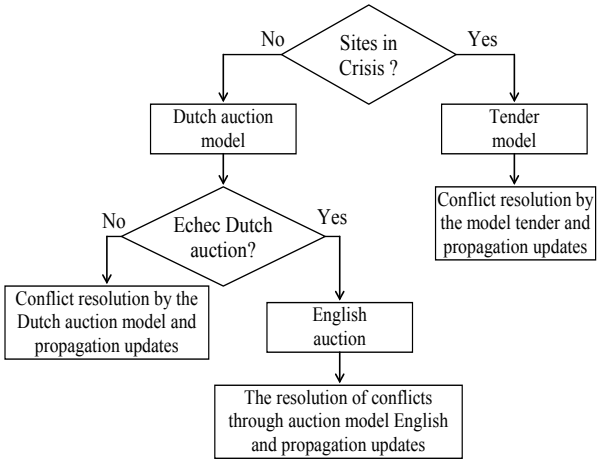


Fig. 3. The main stages of process of conflict resolution

If the trigger event is then checked, one of the two situations is encountered:
Are one or more sites are in a crisis situation, ie, that the manager (MR) of aftershocks following the analysis of information collected said that the information collected is very dispersed. Because of this situation that the decision to trigger the sub-service the overall consistency is taken. In this situation, an economy model type tender is better (see Algorithm Call_Tender).

Algorithm 2 Call_Tender

```
Calculate τi , Di, σi  /* measurements of i^th MR */
candidates ← false, Nbr_candidates ← 0
/* Nbr_candidates represents the sites candidates which can help
the site in crisis */
/* A site is called stable if is not in crisis */
for  all elements of the group of sites in crisis do
Representative_a ← MR of site in crisis
j ← 1
repeat
    Representative_b ← representative of the stable site
    if (|τa - τb| < ε) ∨ (|Da - Db| < ε) ∨ (|σa >σb| < ε) then
          Nbr_candidates ← Nbr_candidates+1
       end if      /* Where  ε << 1 */
       j ← j +1
 until  j ≥ Number of elements of the group of stable sites
 if  (Nbr_candidates > 1) then
    candidates ← true
    Algorithm Negotiation /* Negotiation of candidates */
 else
  Propagation of the updates of the stable site to the sites in crisis
 end if
end for
```

The result of this algorithm can be:
Several MR are candidates to offer their services to sites in crisis, then a negotiation (see Algorithm 3) is underway between them, to choose the site stable (non-crisis) to resolve the crisis sites to propagate its information;
If one MR selected by the algorithm tender, which will spread its information to the sites in crisis.

The negotiation process called by the algorithm of the invitation to tender allows to select the best stable candidate which solves the critical situation of a site in crisis

Algorithm 3 Negotiation

```
τi , Di, σi /* measurements of i^th MR for stable site*/
winner ← first of all sites stable  /* Candidate supposed winner */
τi , Di, σi  /* measurements of i^th MR */
/* Where MR_i ≠ winner
no_candidate ← false
for  all MR – { winner } do
    if (τi  < τwinner) ∧ (Di < Dwinner) ∧ (σi <σwinner) then
        winner ← reprensentativei
    end if
    winner propagates its updates with the sites in crisis ¶
end for
```

Either a chosen period in advance is reached and no site of the data grid is in crisis, then the economy model chosen was that of the Dutch auction (Algorithm 4).

Algorithm 4 Dutch_Auction

representative$_{max}$ ← *MR* having the most recent version
version_reserves /* the average of all vectors of versions */
τ_i , D_i, σ_i /* measurements of ith *MR* */
no_candidate ← false
for all *MR* – {representative$_{max}$} **do**
  if ($\tau_{max} > \tau_i$) ∨ ($D_{max} > D_i$) ∨ ($\sigma_{max} > \sigma_i$) **then**
    if version$_i$ ≤ version_reserves **then**
      no_candidate ← true
    else
      representative$_{max}$ ← MR_i
      no_candidate ←false
    end if
  end if
end for
if (no_candidates = false) **then**
representative$_{max}$ propagates the updates with the whole of representatives
else
  Algorithm English Auction
end if

If no candidate is selected as the most favourable then we will make use of the economy model of English auctions (Algorithm 5).

Algorithm 5 ENGLISH AUCTION

representative$_{min}$ ← *MR* having the oldest version
version_reserves /* the average of all vectors of versions */
τ_i , D_i, σ_i /* measurements of ith *MR* */
for all *MR* – {representative$_{min}$} ***do***
  if ($\tau_{min} > \tau_i$) ∨ ($D_{min} > D_{local\_i}$) ∨ ($\sigma_{min} > \sigma_i$) **then**
    representative$_{min}$ ← MR_i
  end if
end for
representative$_{min}$ propagates the updates with the whole of the representatives

6. Impact of load balancing on the replicas

To improve the quality replicas, we have extended the service of consistency management proposed by a load balancing module (see Figure 4). The main objective of this module is that it allows to study and to measure the influence of the load balancing (Li & Lan, 2004) on quality replicas. We have defines two strategies of balancing that of requests and that of the replicas.

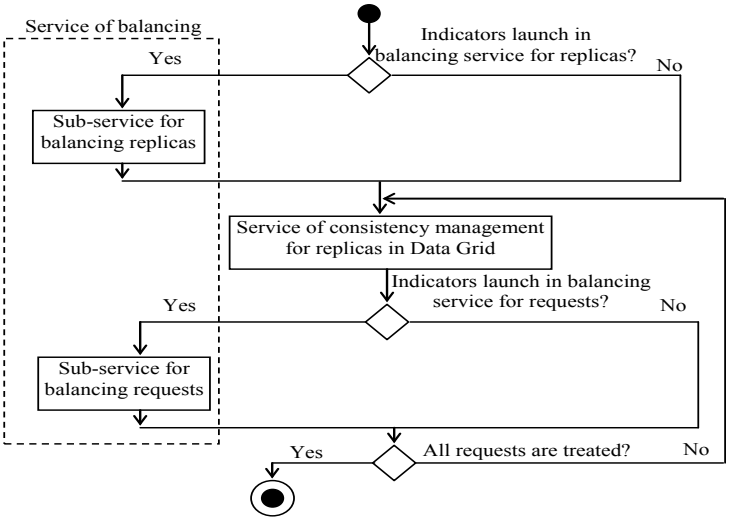


Fig. 4. Service of consistency management extended by the module balancing

6.1 Load balancing of replicas

In accordance the proposed model, we propose one load balancing level for replicas: Inter-sites. The indicators which we chose for sub-service of management of balancing of replicas are developed in Table 1 according to:

Measures	Definition
n_i^d	Number of replicas of the same data d in the $Site_i$
nbT_rep^d	Total of replicas of the same data d in Data Grid
$nb\_req_i^d$	Number of queries that access the same data d in the $Site_i$
nbT_req^d	Total number of queries that access the same data d in the Data Grid
ζ_i^d	The rate of requests accessing the same data d in $Site_i$ $= \frac{nb\_req_i^d}{nbT\_req^d} * 100$
$nb\_necessary\_rep_i^d$	Number of replicates necessary depending on the number of requests of the $Site_i$ $= \frac{\zeta_i^d * nbT\_rep^d}{100}$
ω_i^d	$\frac{n_i^d}{nb\_necessary\_rep_i^d}$
μ	The threshold of imbalance replicas

Table 1. Calculated measures in balancing of replicas

Design of sub-balancing service replicas is composed of three main steps:

1. Phase of collection of information: The manager starts by collecting following information:

- The number of replicas of the data *d* of its site;
- The full number of replicas of data *d* of Data Grid;
- The number of requests of the data *d* of its site;
- The total number of requests of data *d* in Data Grid.

2. Phase of treatment: The manager calculates:

- The rate of requests of the data *d* in its site;
- The necessary number of replicas according to the number of requests of the site;
- The report ω (See Table1).

3. Phase Decision making: During this stage, the manager of the site decides opportunity to start load balancing for replicas.

6.2 Balancing of requests

The management of the balancing requests service intra and inter-sites can be represented by the collaboration diagram as follows (Figure 5):

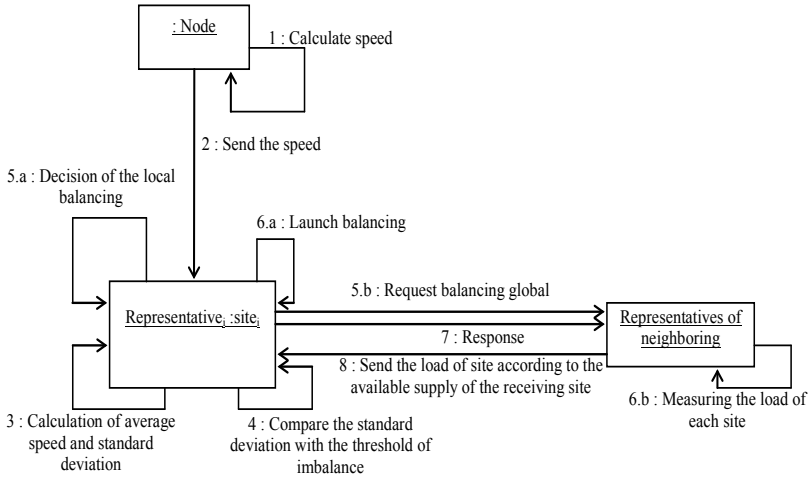


Fig. 5. Collaboration diagram

In accordance with the hierarchical structure of the proposed model, we distinguish two load balancing levels of requests: Intra-site and Inter-sites. The indicators which we chose for sub-service of management of balancing of requests are developed in Table 2.

Measures	Definition
m_i	The number of nodes in the $Site_i$
V_{ij}	The speed of $node_j$ owned $Site_i$ = $Nbr\_req\_processed_{ij} / Total\_nbr\_req_{ij}$
$\overline{V_i}$	The average speed of $Site_i$ $\overline{V_i} = \frac{1}{m_i}\sum_{j=1}^{m_i} V_{ij}$
σ_i	Standard deviation of the $Site_i$ $= \sqrt{[\frac{1}{m_i}\sum_{j=1}^{m_i}(V_{ij} - \overline{V_i})^2]}$
R_{size}	the size of the request
bdp	bandwidth intersites
TTij	transfer time of request of $Site_i$ to $Site_j$
$Demand_{ij}$	Number of requests to transfer from $node_j$ belonging to the $Site_i$
$Offer_{ij}$	Number of requests received by the $node_j$ belonging to $Site_i$
NR	List of recipients nodes
NS	List of sources nodes
$\overline{\omega}$	The threshold of hope
λ	The threshold of the imbalance

Table 2. Calculated measures in request balancing

The request "Demand" is the number of requests to transfer from one node to another node. On first calculates the difference between average speed of the $Site_i$ and speed of $node_j$ appointed (see equation 1),

$$dif_i = \overline{V_i} - V_{ij} \tag{1}$$

this difference is equal to the ratio between the number of queries are not processed by the total number of queries (equation 2)

$$dif_{ij} = \frac{Total\_nbr\_req_{ij} - Nbr\_req\_processed_{ij}}{Total\_nbr\_req_{ij}} = \frac{Nbr\_req\_not\_processed_{ij}}{Total\_nbr\_req_{ij}} \tag{2}$$

from Subtracting this demand, the number of complaints not dealt with, and we write (equation 3)

$$Nbr\_req\_not\_processed_{ij} = dif_{ij} * Total\_nbr\_req_{ij} \tag{3}$$

$$Demand_{ij} = Nbr\_req\_not\_processed_{ij} \tag{4}$$

The offer is the number of queries that the node can receive while remaining balanced. It first calculates the difference between the speed of node j and the average speed of sitei, this difference is called (equation 5),

$$\beta_{ik} = V_{ik} - \overline{V_i} = Nbr\_req\_not\_processed_{ik} / Total\_nbr\_req_{ik} \quad (5)$$

this difference represents the ratio between the number of queries are not addressed by the total number of queries (equation 5), from this formula it is deduced that the offer (equation 6) is equal to the number of queries untreated

$$Offer_{ik} = \beta_{ik} * Total\_nbr\_req_{ik} \quad (6)$$

This sub-service can run a balancing local (balancing intra-site), ie, within a site, if this balance can not be satisfactory, the sub-service triggered a balance between the sites of the grid (balancing inter-sites).

A. Balancing intra-site

In the case of balancing intra-site, three stages can be also used:

Estimate of the load of the site: This stage defines the mechanisms of measurement and of communication of load. Knowing the number of the nodes of the site like their respective capacities, each manager estimates the capacities of the site with which it is associated by carrying out the following actions: Considering the requests are more or less of the same size, we propose like index of load, the speed of treatment requests V. We consider the mean speed of the site starting from the information received periodically by its nodes. This measurement makes it possible to give us a sight on dispersion of speeds of the nodes. We chose to calculate the standard deviation with an aim of measuring the extent of the variations between the mean velocity of the site and speeds of its nodes. Each node sends the information of load to its associated manager.

Decision making: During this stage, the manager of the site decides opportunity to start a local balancing of load. For that, it carries out the following actions:

The state of load of the sites we can say that a site is in a state of balance when this variation is relatively weak. That means that the speed of each node converges towards the mean speed of its site.

State of balance: In practice it is a question of defining a threshold of balance, to leave we can say that the standard deviation tends towards zero and thus the site is in state of balance. Thus we can write: If ($\alpha < \lambda$) then the site is balance if not it site is in a state of imbalance.

Partitioning of the site: When a site is imbalance, we can consider it release of an operation of balancing of requests. To determine if a node of a site is in a suitable state to take part in a transfer of requests like source or like receiver, we divide the site in two groups of Nodes: (i) Overloaded nodes (sources), (ii) undercharged nodes (receivers). This classification depends on the difference between the speed of each node and that of its site.

Transfer of requests: To carry an operation of balancing of requests, we propose the following heuristics:

To calculate request (Demand), i.e., number of requests to be transferred necessary by overloaded node.

To calculate the offer, i.e., the number of requests to be received. We can distinguish three types of undercharged nodes:
The nodes which never received requests, and in this case Offer can have any number of requests
The nodes which treated all their requests Offer can have any many requests
The nodes which have treated requests and untreated requests In this case, if the offer is not able to satisfy the request sufficiently, it is not recommended to start a local balancing. To measure the offer compared to ask, If (Offer>Demand) then to start local balancing If not to start global balancing.

B. Balancing inter-sites
In the case of balancing inter-sites, three stages can be also used:
Estimate of the load of the site;
Decision making;
Transfer of the requests.
If a manager fails to balance his load locally, it estimates its charge compared to its vicinity. In the case of an imbalance, the manager decides to transfer his requests towards sites under charged closest (pertaining to its vicinity). In addition to the collection of information of load of his neighbours, the manager of the site must take account of the costs of communication induced by possible transfer of requests.

7. Experimental results and discussion

To study the management of consistency and its extended version of the module balance, compared to the conventional optimistic approach, we developed a simulator in Java to meet that goal. We used these simulations to a set of parameters. Table 3 describes the main parameters used in our series of experiments.

Parameters	Interval
Number of requests	[10..1000]
Number of data	[1..10]
Number of replicas per data	[10..100]
Number of nodes per site	[10..100]
Number of sites	[5..50]
Size of data (MB)	[5..100]

Table 3. Simulation parameters

In this work, we present three experiments: the first series of experiments used to study the impact of the number of requests on our proposed approach and the optimistic one, for the second series of experiments we measure the impact of the number of sites in the data grid on the results of both approaches.

7.1 Impact of the variation of number requests

In this series, the simulation results have been achieved with the following parameters: 5 sites, 30 nodes per site, 1 data, 30 replicas per site depending on the multi-masters, the

bandwidth is set to 1000 Mb/ms for a star topology in intra-site and 500 Mb/ms inter-sites, we varied the number of requests (such as writing) from 25 to 150 in steps of 25.

A. Impact load balancing of replicas

Figures 6 and 7 show the behaviour of four approaches: ours called hybrid, hybrid with balancing, optimistic approach and the classical optimistic extended by balancing module. The curve is below the curve of the optimistic approach. We conclude that our proposed approach reduces the number of differences and conflicts in relation to the optimistic and that the extension of the approach by balancing module replicas to improve in a positive way the hybrid approach proposed.

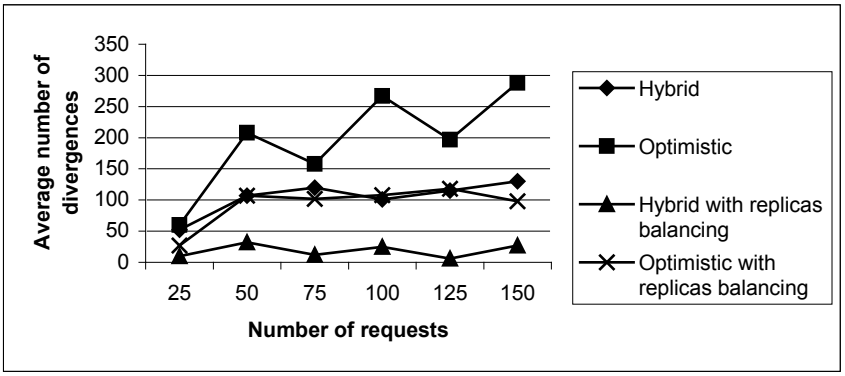


Fig. 6. Average number of divergences / Number of requests

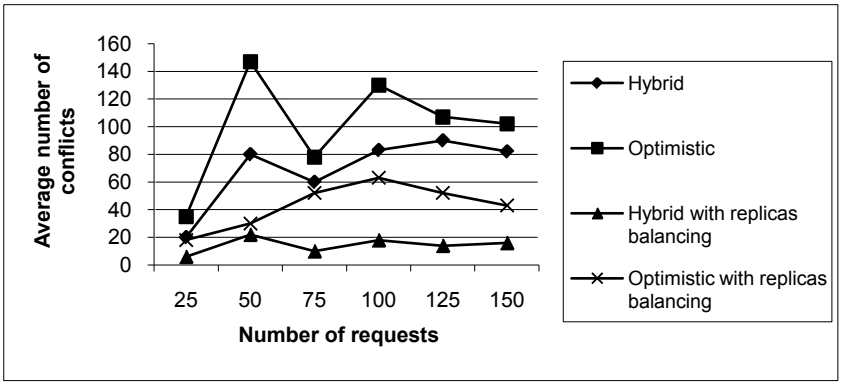


Fig. 7. Average number of conflicts / Number of requests

B. Impact load balancing of requests

Figures 8 and 9 show in a first time for the hybrid approach based on negotiation generates less divergences and conflicts over the traditional approach, and in a second time as the load balancing of requests sent customers by improving the quality of replicas of a very interesting way.

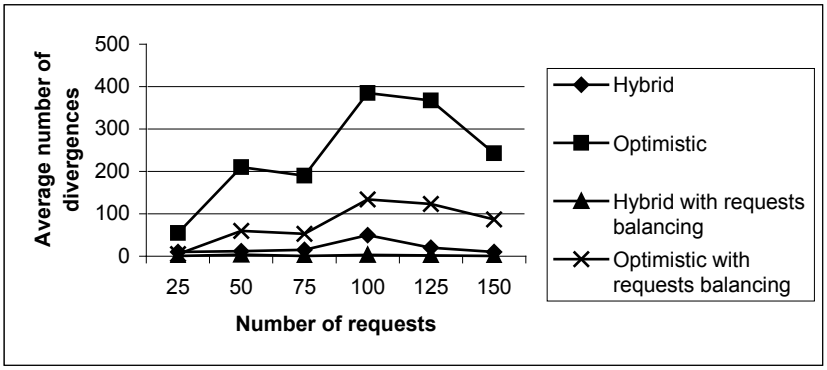


Fig. 8. Average number of divergences / Number of requests

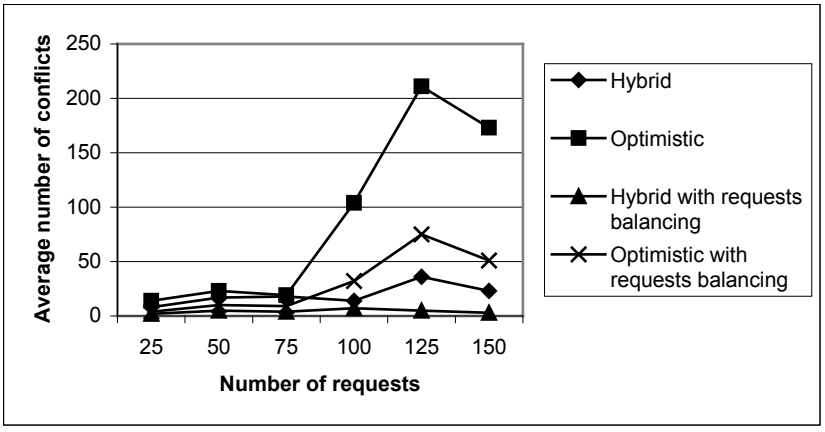


Fig. 9. Average number of conflicts / Number of requests

7.2 Impact of the variation of number sites

This series of simulations is the influence of the number of sites on divergences and conflicts. The parameters we used are: 30 nodes, 100 requests for records, data replicated 30 times in each site according to the multi-master, 1000 Mb/ms bandwidth for a star topology in intra-site and 500 Mb/ms inter-sites and we considered the number of sites ranging from 5 to 20 in steps of 5.

The main objective that we set for the following two simulations is shown how the approaches behave with the scalability of system.

A. Impact load balancing of requests

Figures 10 and 11 show, that the hybrid approach, is still better than the optimistic approach with balancing requests and without. We can also note that our proposal supports the scalability of the system.

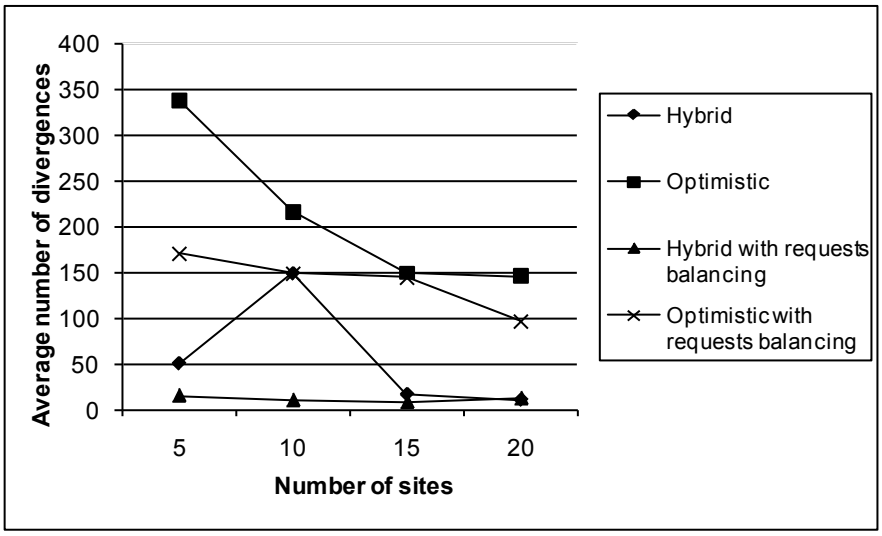


Fig. 10. Average number of divergences / Number of sites

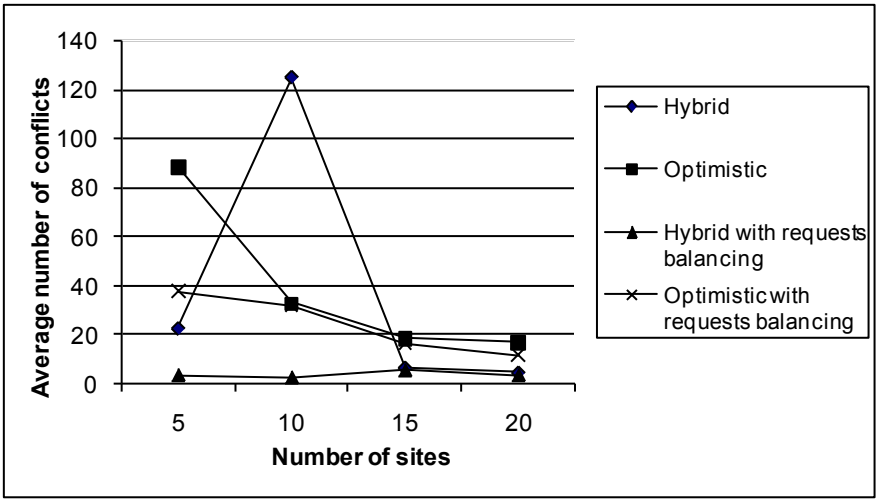


Fig. 11. Average number of conflicts / Number of sites

B. Impact load balancing of replicas

The same result is obtained by the figures 12 and 13. We note, from the number of sites 15, the hybrid approach proposed tends to the hybrid approach with by balancing the module (see Figures 12 and 13).

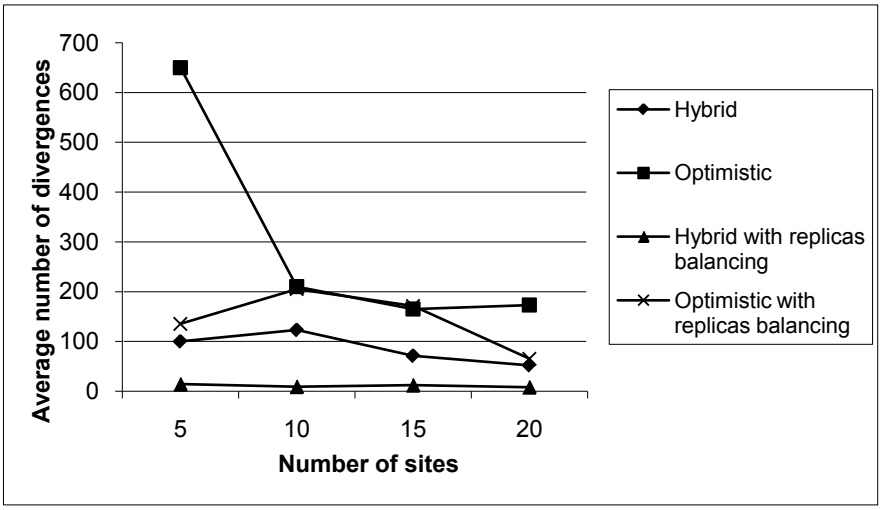


Fig. 12. Average number of divergences / Number of sites

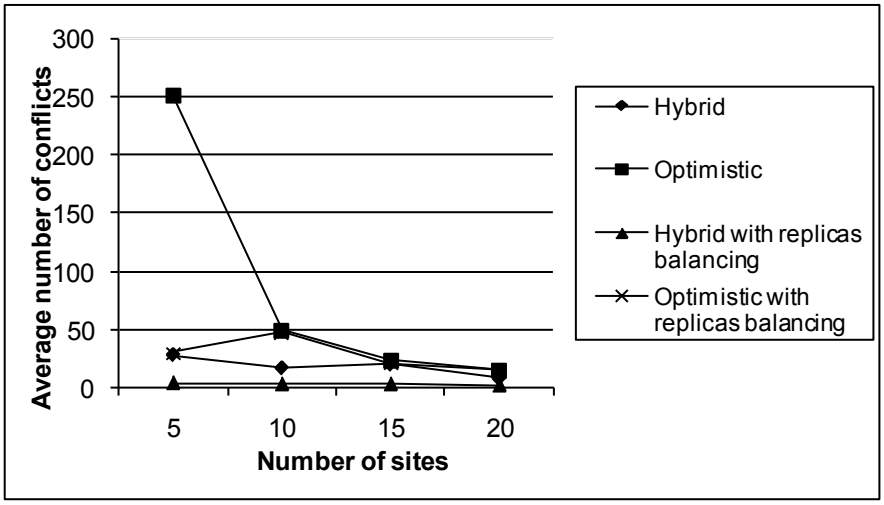


Fig. 13. Average number of conflicts / Number of sites

8. Conclusion

The replication technique is used for data management in distributed systems and grids to ensure data availability and fault tolerance. However, the use of the technique raises the problem of maintaining consistency of replicas. Unfortunately, to ensure the consistency of replicas, you must have a strong consistency, where the degradation of performance.

We have proposed a model for consistency management of replicas in data grids, which combines at the same time the pessimistic approaches, which support the quality of service (QoS), and the optimistic approaches which are focused on the improvement performance. Our effort in this contribution, aims to resolve conflicts between the replicas by using a collaborative negotiation between representatives of the nodes in the data grid. We try to show in this work that trading is influenced by the load balancing of the grid.
The results of the comparison showed that our approach generally ensures a better quality replicas while keeping the same system performance. From these results, we can conclude that the load balancing influence in a positive way on the quality of the replicas.
We want to show by this comparison that the factor balancing requests or replicas has a direct impact on reducing the number of divergences and conflicts between replicas. The work is far from over, we can suggest several perspectives that can be the subject of future work:
Implement our approach in the GLOBUS project;
Improve this approach to tolerate faults in the Data Grid;
Studying the behaviour of the proposed approach in Wireless Grids.

9. References

Belalem, G. & Bouhraoua, F. (2007). Dynamic Strategy of Placement of the replicas in Data Grid, *Malyshkin V. (Ed.) : Parallel Computing Technologies, 9th International Conference, PaCT 2007, Pereslavl-Zalessky, Russia, September 3-7, 2007, Proceedings, Lecture Notes in Computer Sciences (LNCS),* volume 4671/2007, pp. 496-506.

Belalem, G. & Slimani, Y. (2007). A hybrid approach to replica management in data grids, *International Journal Web and Grid Services (IJWGS). Vol.* 3, No. 1, pp. 2-18.

Buyya, R. & Vazhkudai, S. (2001). Compute Power Market: Towards a Market-Oriented Grid, *CCGRID'01, 1st International Symposium on Cluster Computing and the Grid,* May 15-18, 2001, Brisbane, Australia, pp. 574-581.

Cameron, D. G., Casey, J., Guy, L., Kunszt, P.Z., Lemaitre, S., McCance, G., Stockinger, H., Stockinger, K., Andronico, G., Bell, W. H., Ben-Akiva, I., Bosio, D., Chytracek, R., Domenici, A., Donno, F., Hoschek, W., Laure, E., Lucio, L., Millar, P., Salconi, L., Segal, B. & Silander, M. (2004), Replica Management in the European DataGrid Project, *Journal Grid Computing*, Vol. 2, No. 4, pp. 341-351.

Dorcey T. (1995). Cu-SeeMe desktop videoconferencing software. *Connexions,* Vol. 9, No. 3, pp. 42-45.

Foster, I. & Kesselmann, C. (2004). *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier Series in Grid Computing. Morgan Kaufmann Publishers.

Goel, S., Sharda, H., & Taniar, D. (2005). Replica synchronisation in grid databases. *International Journal Web and Grid Services (IJWGS)*, Vol. 1, No. 1, pp. 87-112.

Gray, J., Helland, P., Neil, P. O., & Shasha, D. (1996). The dangers of replication and a solution. *In ACM SIGMOD International Conference on Management of Data*, pp. 173-182, Montreal, Quebec, Canada, 4-5 June 1996. ACM Press.

Hasegawa, K., Higaki, H. & Takizawa, M., (1998). Object Replication Using Version Vector, *In Proceedings International Conference on Parallel and Distributed Systems* (ICPDS'98), 14-16 Dec 1998, Tainan, Taiwan, pp. 147-154.

Kermarrec, A-M., Rowstron, A., Shapiro, M. & Druschel, P. (2001). The IceCube approach to the reconciliation of divergent replicas. *PODC '01: Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pp. 210-218, Newport, Rhode Island, United States.

Kistler, J. J. & Satyanarayanan, M. (1992). Disconnected operation in the coda file. *ACM Trans. on Computer Systems*, Vol. 10, No. 1, pp. 3-25.

Li, Y. & Lan, Z. (2004). A survey of load balancing in grid computing. *High Performance Computing and Algorithms*, Computational and Information Science, First International Symposium, CIS 2004, Shanghai, China, December 16-18, 2004, Lecture Notes in Computer Science (LNCS), volume 3314/2004, pp. 280-285.

Molli, P., Oster, G., Rusinowitch, M. & Imine, A. (2002). Development of Transformation Functions Assisted by Theorem Prover. *The Fourth International Workshop on Collaborative Editing*, ACM CSCW 2002, New Orleans, Louisiana, USA.

Olston, C. & Widom, J. (2005). Efficient monitoring and querying of distributed, dynamic data via approximate replication. *IEEE Data Eng. Bull.*, Vol. 28, No. 1, pp.11-18.

Petersen, K., Spreitzer, M., Terr,y D. & Theimer, M. (1996). Bayou: replicated database services for world-wide applications, *EW 7: Proceedings of the 7th workshop on ACM SIGOPS European workshop*, pp. 275-280, Connemara, Ireland.

Ratner, D., Reiher, P. & Popek, G. (1997). Dynamic version vector maintenance. *Technical Report CSD-970022*, UCLA, June 1997.

Saito, Y. & Shapiro, M. (2005). Optimistic replication. *ACM Computing Surveys*, Vol. 37, No. 1, pp. 42-81.

Vidot, N., Cart, M., Ferrié, J. & Suleiman, M. (2000). Copies convergence in a distributed real-time collaborative environment, *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 171-180, Philadelphia, Pennsylvania, USA.

Xu, J., Li, B. & Li, D. (2002). Placement problems for transparent data replication proxy services. *In IEEE Journal on Selected areas in Communications*, Vol. 20, No. 7, pp. 1383-1398.

21

Performance Improvements of Peer-to-Peer File Sharing

Dongyu Qiu
Concordia University
Canada

1. Introduction

In recent years, *Peer-to-Peer* (P2P) applications, in which peers serve as both clients and servers, have changed the Internet dramatically. Compared to traditional client/sever applications (such as *FTP, HTTP*), P2P systems normally have much better scalability. The performance of client/server applications deteriorates rapidly as the number of clients increases, while in a well-designed P2P system, more peers generally means better performance. Among all the P2P applications, file sharing has been one of the most popular. Traffic from P2P file sharing applications, such as Kazaa, Gnuttella, eDonkey, and BitTorrent (Cohen, 2003), has been dominating the Internet bandwidth in recent years. In this chapter, we will focus on the performance improvements of BitTorrent networks.

The performance of a BitTorrent network is affected by many factors. For example, how many pieces the served file is divided? How many neighbors a given peer has? Are peers cooperative or not? etc. In this chapter, we will try to improve the performance of a BitTorrent network from two aspects. Firstly, we assume that all peers in the network are cooperative, i.e., peers are willing to contribute by uploading. Under this assumption, we propose a stochastic model to study how to design an efficient P2P system. Secondly, we relax the cooperation assumption and study how to prevent selfish peers from free-riding. In BitTorrent, there are two built-in mechanisms to prevent free-riding. However, they are not very efficient. In this chapter, we will focus on one of the mechanisms called "optimistic unchoking" and discuss how its performance can be improved.

In a P2P network, if a peer is cooperative, it contributes to the network through uploading and hence it is important to efficiently utilize the upload bandwidth of each peer. The number of pieces that a given peer has is an important factor that affects the upload bandwidth utilization. For example, when a peer first enters the network, it has no pieces at all and hence can not upload to anyone. The upload bandwidth utilization is 0 in this case. On the other hand, when a peer has most of the pieces, it is very likely that it can upload to others and hence the utilization is close to 1. Motivated by this fact, we will propose a stochastic model to study the peer distribution with regards to the number of pieces that a peer has. More specifically, we are interested in P_i, which is the probability that a random peer has i pieces, where $0 \leq i \leq N$ and N is the total number of pieces of the served file. Note that in BitTorrent, peers that have the whole file are called seeds, while other peers are called downloaders. By numerically solving the proposed model, we will be able to gain interesting insight on how the performance of a P2P file sharing network is affected by different parameters such as the piece numbers of the

file, the number of neighbors of a peer, and the seed departure rate etc. We will then provide some useful guidelines on how to design an efficient P2P system based on our results.
In a real network, however, peers may not always be cooperative. The so-called free-riders are selfish peers that try to download from the network while not contribute (or upload) at all. BitTorrent has a built-in incentive mechanism called "Tit-for-Tat" to encourage peers to upload and another mechanism called "optimistic unchoking" to find upload bandwidth information about neighbors. However, both "Tit-for-Tat" and "optimistic unchoking" have some drawbacks. For example, a peer can adjust the upload rate and the number of uploads to take advantage of Tit-for-Tat. It has also been shown that a free-rider can obtain at least one fifth download rate of a normal peer just from optimistic unchoking. Hence, the two mechanisms can not efficiently prevent free-riding. In this chapter, we will propose a new optimistic unchoking algorithm for BitTorrent. Theoretical analysis and simulation results will be provided to show the effectiveness of our new mechanism.
The chapter is organized as follows. In Section 2, we will study how the network efficiency is affected by different parameters and discuss some guidelines on how to design an efficient P2P file sharing network. In Section 3, we will propose a new optimistic unchoking algorithm for BitTorrent and show how the new mechanism can improve the performance of the network.

2. The Efficiency of Peer-to-Peer File Sharing

In P2P file sharing, an interested file is divided into many pieces. The size of each piece ranges from several hundred kilobytes to several megabytes. When a new peer joins the network, it begins to download pieces from other peers. As long as it obtains one piece of the file, the new peer can start to serve other peers by uploading pieces. Since peers are downloading and uploading at the same time, when the network becomes large, although the demands increase, the service provided by the network also increases. Hence, the performance of the P2P network scales very well.
BitTorrent has been one of the most popular P2P file sharing applications and has attracted a lot of research attentions. While early work on P2P systems has mainly focused on system design and traffic measurements (Ng et al., 2003; Ripeanu, 2001; Ripeanu et al., 2002), some recent research has emphasized on performance modeling. In (Ge et al., 2003), a closed queueing system is used to model a general P2P file sharing system and basic insights on the stationary performance are provided. In (Clevenot & Nain, 2004; Clevenot et al., 2005), a stochastic fluid model is used to study the performance of P2P web cache (SQUIRREL) and cache clusters. In (de Veciana & Yang, 2003; Yang & de Veciana, 2004), a branching process is used to study the service capacity of BitTorrent-like P2P file sharing in the transient regime and a simple Markovian model is presented to study the steady-state properties. In (Susitaival et al., 2006), a spatio-temporal model is proposed to analyze the resource usage of P2P systems. In (Susitaival & Aalto, 2006), an approximation for the life time of a chunk in BitTorrent is proposed. (Guo et al., 2005) presents an extensive trace analysis and modeling study of BitTorrent-like systems. In (Tian et al., 2006), the authors studied the behavior of peers in BitTorrent and also investigated the file availability and the dying-out process. In (Qiu & Srikant, 2004), a simple fluid models is proposed to study the performance and scalability of BitTorrent-like P2P systems.
In P2P file sharing networks, the upload bandwidth of each peer is a very important resource of the network. The efficient use of it will impact the system performance significantly. However, little research has been done in this area. In this section, we will focus on the efficiency

of P2P file sharing. More specifically, we will use a stochastic model to study how the efficiency is related to different network parameters, such as, the number of pieces, the number of neighbors, and the seed departure rate etc.
This section is organized as follows. In Section 2.1, we will propose a stochastic model for BitTorrent-like P2P file sharing networks and use this model to study the network efficiency. Simulation and numerical results of our model will be shown in Section 2.2 and we will also discuss the impacts of the results. Finally, we conclude in Section 2.3.

2.1 Stochastic Model

In a P2P network, each peer contributes to the network through uploading and hence it is important to efficiently utilize the upload bandwidth of each peer. The number of pieces that a given peer has is an important factor that affects the upload bandwidth utilization. For example, when a peer first enters the network, it has no pieces at all and hence can not upload to anyone. The upload bandwidth utilization is 0 in this case. On the other hand, when a peer has most of the pieces, it is very likely that it can upload to others and hence the utilization is close to 1. Motivated by this fact, we will next propose a stochastic model to study the peer distribution with regards to the number of pieces that a peer has. More specifically, we are interested in P_i, which is the probability that a random peer has i pieces, where $0 \leq i \leq N$ and N is the total number of pieces of the served file. Note that in BitTorrent, peers that have the whole file are called seeds, while other peers are called downloaders.
From real trace measurements, it has been observed that a BitTorrent-like P2P file sharing network has three phases (Yang & de Veciana, 2004), a growing phase, a stabilizing phase, and a decaying phase. The stabilizing phase is normally the one that most of the downloads take place and hence it is the one that determines the performance of the system. In this section, we consider a large P2P network that is in the steady state. Hence, the peer distribution $\{P_i\}$ is not changing with time. When a new peer first enters the network, it randomly picks up L peers as its neighbors. For the simplicity of analysis, we assume that the number of neighbors of each peer is fixed at L. For a given peer with i pieces, we assume that these pieces are chosen randomly from the set of all pieces of the file. This is a reasonable assumption because BitTorrent-like systems take a rarest first piece selection policy when downloading. Hence, it is unlikely that one piece has significantly more copies than other pieces. We also assume that these pieces are chosen independently with other peers. In P2P networks, a peer is normally downloading from many neighbor peers at the same time. Hence, a single neighbor's effect on the given peer's pieces can be neglected and it is reasonable to assume the independence between peers. Since we are interested in the efficient use of upload bandwidth, for simplicity of analysis, we assume all peers have the same upload bandwidth and the download bandwidth is unlimited. We use a discrete time model and without loss of generality, we assume that the upload bandwidth of a peer is one piece per time slot.
Note that although our model is relatively simple, it has all the important features of a typical P2P network and we expect the model can shed some light on further research of P2P file sharing efficiency. Next, we will start the analysis by focusing on only two peers in a neighborhood first. Assume that peer A has i pieces and peer B is a neighbor of peer A with j pieces. We define $F(i,j)$ to be the probability that peer B has no pieces that peer A is interested, i.e., peer A has all pieces of peer B, then

$$F(i,j) = \begin{cases} 0, & i < j \\ \frac{\binom{N-j}{i-j}}{\binom{N}{i}}, & i \geq j \end{cases}$$

In BitTorrent-like systems, when a peer obtains a new piece, it will update this information with its neighbors and hence a peer knows what pieces its neighbors have. At the beginning of a time slot, a peer will send requests for pieces to its neighbors if a neighbor has pieces that the peer is interested in. At the same time, the peer will also receive requests from its neighbors. If the peer receives more than one requests, it will randomly pick up one request to fulfill. Note that in a real BitTorrent network, peers will fulfill requests according to a built-in incentive mechanism. How the incentive mechanism will affect the performance is out of the scope of this chapter. Again, let's consider two peers A and B, where A has i pieces and B has j pieces. Under the condition that B has pieces that A is interested in, A will send a request to B. But the request from A may not be fulfilled since B also receives requests from other peers and B will randomly pick up one to fulfill. Let X be the number of requests that peer B receives besides A's request. Then X is a Binomial random variable with parameters $L-1$ and q_j, where $L-1$ is the maximum number of requests B could receive besides A's request and q_j is the probability that a randomly picked neighbor of B sends request to B. We have

$$q_j = \sum_{k=0}^{N} P_k(1 - F(k, j)).$$

The probability distribution of the random variable X is then

$$\mathbb{P}\{X = k\} = \binom{L-1}{k} q_j^k (1 - q_j)^{L-1-k},$$

for $k = 0, \cdots, L-1$. So, when peer A sends a request to B, the probability that the request is fulfilled is

$$\begin{aligned}
G_j &= \sum_{k=0}^{L-1} \frac{1}{k+1} \mathbb{P}\{X = k\} \\
&= \sum_{k=0}^{L-1} \frac{1}{k+1} \binom{L-1}{k} q_j^k (1 - q_j)^{L-1-k} \\
&= \sum_{k=0}^{L-1} \frac{(L-1)!}{(k+1)!(L-1-k)!} q_j^k (1 - q_j)^{L-1-k} \\
&= \frac{1}{Lq_j} \sum_{k=0}^{L-1} \frac{L!}{(k+1)!(L-1-k)!} q_j^{k+1} (1 - q_j)^{L-1-k} \\
&= \frac{1}{Lq_j} \sum_{k=0}^{L-1} \binom{L}{k+1} q_j^{k+1} (1 - q_j)^{L-1-k} \\
&= \frac{1}{Lq_j} \sum_{m=1}^{L} \binom{L}{m} q_j^m (1 - q_j)^{L-m} \\
&= \frac{1 - (1 - q_j)^L}{Lq_j}
\end{aligned}$$

The probability that peer A can download a piece from a random neighbor is then

$$S_i = \sum_{j=0}^{N} P_j(1 - F(i, j)) G_j.$$

In BitTorrent-like systems, at any given time slot, a peer only sends one request for a given piece. Hence, if a peer has i pieces, the maximum number of requests it can send is then $D = \min(L, N - i)$. The number of pieces that it downloads in the same time slot is then a Binomial random variable with parameters D and S_i. Define $r_{i,k}$ to be the probability that a peer with i pieces downloads k pieces in the current time slot, then

$$r_{i,k} = \binom{D}{k} S_i^k (1 - S_i)^{D-k}, \tag{1}$$

where $i = 0, \ldots, N-1$ and $k = 0, \ldots, D$. The average number of pieces that a peer with i pieces can download in a given time slot is then

$$d_i = \sum_{k=1}^{\min(L,N-i)} k r_{i,k}.$$

We also call d_i the average download rate of the peer. When the system is in the steady state, the peer distribution should not change with time. So, we have,

$$P_i = \sum_{k=0}^{\min(i,L)} P_{i-k} r_{i-k,k} \tag{2}$$

for $i = 1, \ldots, N$. The case $i = 0$ is a special one that is related to the peer arrival rate and we deal with it separately as following. When a peer has downloaded all the pieces, it becomes a seed. After a peer becomes a seed, it will stay in the system for an exponentially distributed time. Since we use a discrete time model, we define γ to be the probability that a seed will leave the system in the current time slot. Then $P_N \gamma$ is the total seed departure rate. When the system is in steady state, it should equal to the peer arrival rate. So, we have

$$P_0 = P_0 r_{0,0} + P_N \gamma. \tag{3}$$

Solving Eqs. (1)(2)(3), we can then obtain the peer distribution $\{P_i\}$. Note that it is hard to get a closed form peer distribution. However, the equations can easily be numerically solved and the results will be discussed in Section 2.2.

Once we know the peer distribution, we can use it to study important performance such as the average download time of a P2P network. Let the peer arrival rate be λ and the average number of peers in the system be M. Define T to be the average time a peer stays in the system and T_d to be the average download time of a peer (i.e., the time from a peer enters the system to it becomes a seed). Applying the Little's Law to the whole system, the seeds, and the downloaders respectively, we have

$$M = \lambda T$$
$$M P_N = \lambda \frac{1}{\gamma}$$
$$M(1 - P_N) = \lambda T_d$$

It is easy to see that

$$T = \frac{1}{P_N \gamma} \tag{4}$$

$$T_d = \frac{1 - P_N}{P_N \gamma} \tag{5}$$

Next, we will study how the system performance can be affected by different parameters such as the number of pieces N, the number of neighbors L, and the seed departure rate γ etc.

2.2 Simulation and Numerical Results

To validate the proposed stochastic model, we first simulate a BitTorrent-Like system and compare the simulation results with the peer distribution derived from the stochastic model. Our simulation follows the same assumptions as the stochastic model. Time is slotted and the upload bandwidth of a peer is one piece per time slot. In the simulation, the served file is divided into $N = 200$ pieces. Each peer is connected to $L = 20$ neighbor peers. Peers arrive to the network according to a Poisson process with the average arrival rate λ and the seed departure rate is $\gamma = 0.01$. This is a quite typical BitTorrent system setting since when a peer first enters the network, the default number of peers it gets from the tracker (Cohen, 2003) is normally 20. In Fig. 1 and 2, we show the simulation results of piece distribution for $\lambda = 20$ and $\lambda = 50$ respectively. In both cases, we see that the simulation results match well with the theoretical results derived from the stochastic model. Furthermore, when the peer arrival rate is larger ($\lambda = 50$), the simulation result is closer to the theoretical one. This is because our model assumes a very large P2P network. When λ is large, there will be more peers in the network and hence the stochastic model is more accurate.

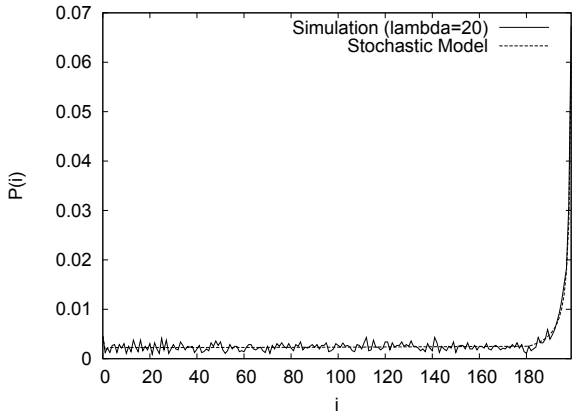


Fig. 1. Piece Distribution ($N = 200, \lambda = 20$)

In Figs. 1 and 2, the peer distribution is with regards to the number of pieces i that a peer has. We see when $i \leq 180$, peers are almost uniformly distributed. And when $i > 180$, the probability that a peer has i pieces increases when i increases. That is because when a piece has most pieces ($i > 180$), it can't fully utilize all of its neighbors anymore and hence its download rate decreases. We call this the end-game effect. In BitTorrent, there is even an end-game mode to deal with this. The details about the end-game mode is out of scope of this chapter. In Fig 3, we also show the average download rate of a peer with i pieces, we clearly see that when $i \leq 180$, the download rate is almost a constant. This explains the reason why in BitTorrent, the download rate of a peer can normally maintain at a high value for most part of the download process. When $i > 180$, however, the download rate keep decreasing and that also explains the result of Figs 1 and 2. Note that starting from Fig 3, we will only show the numerical results derived from the stochastic model to illustrate the figures more clearly. In Fig. 4 and Fig. 5, we change the piece number from 200 to 100 and keep other parameters the same. We observe similar results as in the case of $N = 200$. However, since the piece number is 100 now, the end-game effect starts from $i = 80$.

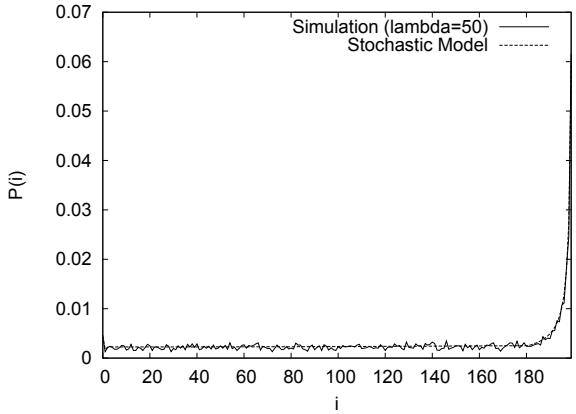


Fig. 2. Piece Distribution ($N = 200, \lambda = 50$)

In Fig 6, we keep $N = 200, L = 20$ and increase γ from 0.01 to 0.9. It shows the normalized download time $\frac{Td}{N}$ as a function of the seed departure rate γ. When γ increases, the download time also increases because there are fewer seeds in the system. However, if $\gamma > 0.3$, when γ increases, the download time doesn't change much. This tells us that the seed departure rate affects the system performance but it is not significant once γ is greater than some threshold. Note that when γ is too large, it may happen that no single seed in the system and hence cause the survivability problem of the network.
In Fig 7, we keep $L = 20, \gamma N = 2$ and increase the piece number N from 25 to 300. We see that when N increases, the average download time decreases as expected since larger N means a peer is more likely to upload to its neighbors. Note that we are keeping γN a constant instead of just γ because when N increases, the length of each time slot decreases (assuming the file size is a constant), hence we need to adjust γ accordingly.
In Fig 8, we keep $N = 200, \gamma = 0.01$ and increase the neighbors from $L = 2$ to $L = 20$. We see that when $L = 6$, the download time is the smallest. The explanation is that when L is too small, the probability that a peer can upload to its neighbor is small and hence not very efficient. However, when L is too large, the end game effect will be significant and will increase the download time. In Fig 9, we show the effect of the neighbor number when the parameters are changed to $N = 100, \gamma = 0.02$. Again, we see that the download time increases when L is too small or too large. In this case, $L = 4$ gives the smallest download time.

2.3 Conclusion

In this section, we propose a stochastic model to analyze the efficiency of P2P file sharing. We also verify the model through simulations. By solving the model numerically, we are able to gain some important insights on how the performance of P2P file sharing is affected by different parameters and based on these results, we are able to obtain some guidelines on how to design an efficient P2P file sharing system. Our contributions are: 1. The end game stage affects the performance significantly. To improve the performance, it is important to alleviate the end game effect. 2. If not considering the survivability, the seed departure rate doesn't affect the performance significantly when it is large enough. Hence, as long as we

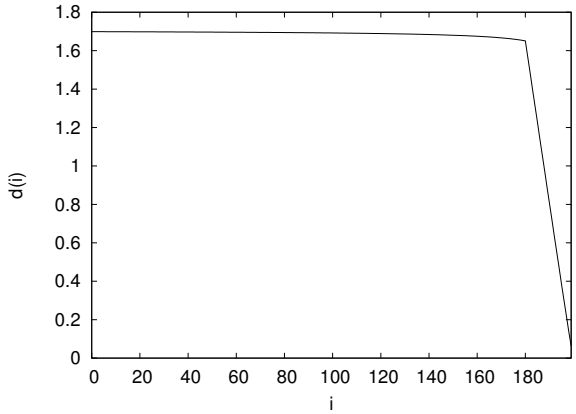


Fig. 3. Download Rate Distribution ($N = 200$)

have at least one seed in the system, it is not necessary to ask seeds to stay in the system for a long period of time. 3. Too many neighbors may affect the performance adversely since it cause more severe end game effect. Hence, it is important to choose a reasonable number of neighbors.

3. The Optimistic Unchoking Algorithm of BitTorrent

In a BitTorrent network, a shared file is divided into many pieces (the default size of a piece is 256KB). Peers that have the whole file are called seeds, while other peers with partial or none of the file are called downloaders. When a peer first joins the network, it connects to a central server called tracker to get a list of peers (Note that the latest version of BitTorrent supports trackerless torrents, where the centralized tracker is replaced by distributed tracking). The new peer then connects to those peers to request for pieces and those peers become the neighbors of the new peer. Once the new peer obtains at least one pieces, it can start to contribute to the network by uploading pieces. The peer then exchanges pieces with its neighbors until it obtains all the pieces and becomes a seed. Once a peer becomes a seed, it may decide to stay in the network to serve other peers or just leave the network. From the above description, we see that in BitTorrent, a peer is normally a client and a server at the same time. When the network becomes large, there will be more peers to request service, but at the same time, there will also be more peers to contribute to the network. Hence, the performance may not degrade and that is why BitTorrent has good scalability.

Note that the good scalability of BitTorrent is based on the assumption that peers are cooperative and are willing to contribute to the network. However, in reality, a large amount of peers are so-called free-riders. Free-riders are selfish peers that try to download from the network while not contribute (or upload) at all. BitTorrent has a built-in incentive mechanism called "Tit-for-Tat" to encourage peers to upload. A peer in BitTorrent normally chooses a fixed number of other peers (default is four) (Cohen, 2003) to upload (also called unchoke in BitTorrent) and those chosen peers are the ones that give the current peer the highest download rates. So basically, if you want to download from others, you should also upload to them

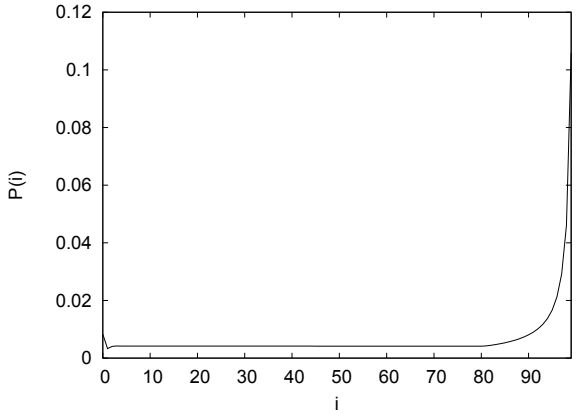


Fig. 4. Piece Distribution ($N = 100$)

as exchange. However, the incentive mechanism also has it own drawbacks. For example, if a new peer with high upload bandwidth joins the network, since it has nothing to upload yet, no other peers will upload to it and hence the high upload bandwidth of the new peer will be wasted. To solve this problem, BitTorrent uses another mechanism called "optimistic unchoking". Besides the normal unchokes we mentioned above, a peer also randomly choose another peer to upload and this is called optimistic unchoking. When a new peer joins the network, it can get its first piece through optimistic unchoking and after that, it can participate in the normal exchange process. Optimistic unchoking is also used to discover peers with high upload bandwidth in the network. By random choosing a peer to upload, it is possible to find a peer with higher upload bandwidth than currently unchoked four peers and hence increase the total download rate.

However, optimistic unchoking also introduces new problems. One of them is the free-riding. In (Qiu & Srikant, 2004), it was shown that a free-rider can obtain at least one fifth download rate of a normal peer just from optimistic unchoking. Since all P2P applications are based on the contributions of individual peers, if free-riding becomes serious, the network may not be able to survive. Hence, it is very important to provide a good mechanism to prevent free-riding. In this section, we will propose a new optimistic unchoking algorithm which can prevent free-riding much more effectively and at the same time, can improve the performance of normal peers.

This section is organized as following. In Section 3.1, we will discuss related work. In Section 3.2, a new optimistic unchoking algorithm for BitTorrent is proposed. In Section 4, a stochastic model is proposed to study the new algorithm. In Section 4.1, we present the simulation results. Finally, we draw the conclusion in Section 4.2.

3.1 Related Work

As one of the most popular applications in the current Internet, BitTorrent has attracted a lot of research interest. Some recent research has emphasized on performance modeling of BitTorrent. In (Yang & de Veciana, 2004), a branching process is used to study the service capacity of BitTorrent-like P2P file sharing in the transient regime and a simple Markovian

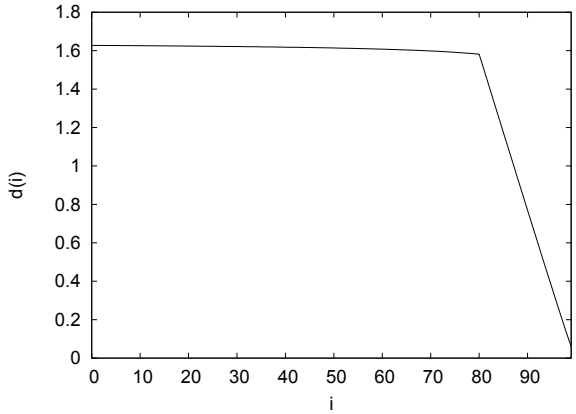


Fig. 5. Download Rate Distribution ($N = 100$)

model is presented to study the steady-state properties. In (Qiu & Srikant, 2004), a simple fluid models is proposed to study the performance and scalability of BitTorrent-like P2P systems. In (Tian et al., 2006), the authors studied the behavior of peers in BitTorrent and also investigated the file availability and the dying-out process.

The incentive mechanism is another important research topic. In (Qiu & Srikant, 2004), the effect of "Tit-for-Tat" is studied when selfish peers are able to adjust their uploading bandwidth. In (Zhang et al., 2007), an overlay formation game model is used to study the existence of Nash equilibrium and the loss of efficiency when peers can change the number of connections. In (Piatek et al., 2007), it is shown that the "Tit-for-Tat" incentive mechanism is generally not robust to strategic clients. A strategic client can take advantage of "Tit-for-Tat" and hurt the performance of other peers.

Our work in this chapter differs from the above-mentioned work in the following respect: we focus on the optimistic unchoking instead of the built-in incentive mechanism of BitTorrent. From our discussion above, we see that the optimistic unchoking plays an important role in BitTorrent. However, the current optimistic unchoking algorithm used in BitTorrent is not effective. Our theoretical and simulation results show that a well-designed optimistic unchoking algorithm can improve the performance significantly. In addition, unlike the incentive mechanism which requires all peers to implement the same mechanism for it to work well, our proposed optimistic unchoking algorithm can be deployed progressively. A peer can improve its performance by using our algorithm no matter other peers use the same algorithm or not. Our proposed algorithm can also co-exist with any incentive mechanisms, no matter it is the built-in "Tit-for-Tat" or other mechanisms proposed in the literature. In this sense, our proposed optimistic unchoking algorithm can be seen as a good complement to the incentive mechanism.

3.2 A New Optimistic Unchoking Algorithm

Before we propose the new optimistic unchoking algorithm, let us do a quick review of the purposes of optimistic unchoking and to see why the current algorithm in BitTorrent is not effective. The main purpose of optimistic unchoking is to discover peers with high upload

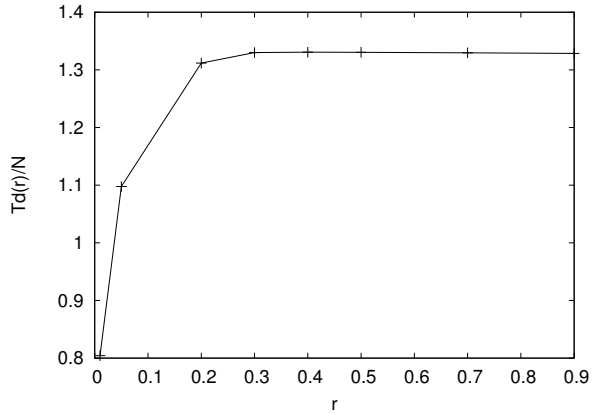


Fig. 6. The effect of seed departure rate

bandwidth and hence to improve the total download rate. The built-in incentive mechanism is not helpful here because the "Tit-for-Tat" only choose from the neighbors that are currently upload to you. If a peer is not uploading to you, the "Tit-for-Tat" will not help you to find what its upload bandwidth is. So, in BitTorrent, besides the four normal unchokes, a peer will also randomly choose a neighbor to unchoke. In its simplest form, a peer may use a round-robin fashion to choose the optimistic unchoking and after several rounds, the peer may be able to discover the upload bandwidth information about its neighbors. The second purpose of optimistic unchoking is to help new joined peers to obtain the first piece quickly. The current optimistic unchoking algorithm in BitTorrent, however, is not very effective. First, it does not take advantage of the history information. A peer will randomly choose a neighbor as optimistic unchoking even if it knows the neighbor may have a very low upload bandwidth. Secondly, it encourages free-riding. In (Qiu & Srikant, 2004), it was shown that a free-rider can obtain at least one fifth download rate of a normal peer just from optimistic unchoking. Next, we will present a new optimistic unchoking algorithm to solve the mentioned problems.
In our new algorithm, we try to utilize the history information we obtained through past optimistic unchokings. To do so, a peer i need to maintain some information about its neighbors. Let L to be the number of its neighbors and $j = 1, \cdots, L$ be the index of a neighbor j. We define the following terms.

N_j The number of times that peer j has been optimistically unchoked.

n_j Among the N_j unchokes, the number of times that peer j responded, i.e., peer j also unchoked (or uploaded to) peer i.

u_j The average upload rate of peer j. Note that if peer j never uploaded to peer i, the information of u_j may not be available.

U_{max} The maximum average upload rate of peer i's neighbors, i.e., $U_{max} = \max u_j$.

Note that in BitTorrent, the default value of L is 20. So the resource required to maintain the history information we proposed here is reasonable. Next, for each neighbor j, we define a gain-value G_j. Basically, G_j is the expected gain or benefit we can obtain if we unchoke peer j

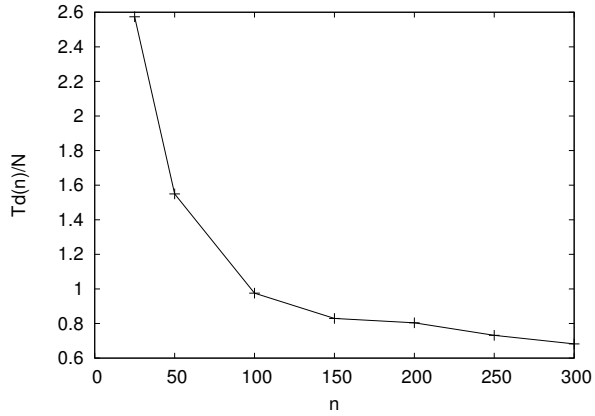


Fig. 7. The effect of piece number

in the current time slot (note that here we use a slotted time model for its simplicity). If $n_j > 0$, i.e., peer j has uploaded to us before and we have the upload rate information of j, then

$$G_j = \frac{u_j n_j}{N_j}.$$

If $n_j = 0$, peer j has never uploaded to us before and we don't have any information about its upload rate. In this case, we define

$$G_j = \frac{U_{max}}{N_j + 1},$$

i.e., we use U_{max} as the estimation of the upload rate of peer j. There are two reasons for doing that. First, recall that the main purpose of optimistic unchoking is to discover the upload bandwidth of unknown peers. So, using U_{max} as the estimation will give the unknown peer j a high gain-value and hence a high chance to be unchoked. Secondly, when a new peer join the network, its $N_j = 0$ and hence $G_j = U_{max}$. That means the new peer will be very likely unchoked and this will help the new peer to obtain its first piece as soon as possible.
Once we obtain the gain-value G_j of all neighbors, we choose the peer that give us the highest gain-value to unchoke. In the case of a tie happens, we randomly choose a peer among those with the highest gain-value.
From the description of the new algorithm, we see that the new algorithm takes advantage of history information and always unchokes the peer with the highest expected gain. While in the original algorithm, a peer choose the optimistic unchoke randomly and overall it may only obtain the average gain. Hence the new algorithm, on one side, can improve the performance of normal peers. On the other side, the new algorithm can also prevent free-riding. A free-rider will never upload to other peers. So its gain-value will be $G_j = \frac{U_{max}}{N_j+1}$. To understand this more clearly, let's assume the system is in a steady state and the highest gain-value converges to a constant $C > 0$. Obviously, when $N_j > \frac{U_{max}}{C} - 1$, the gain-value of the free-rider will always less than C. So the free-rider will only be unchoked a finite number of times. After that, the free-rider will never get any downloading through optimistic unchoking. In other

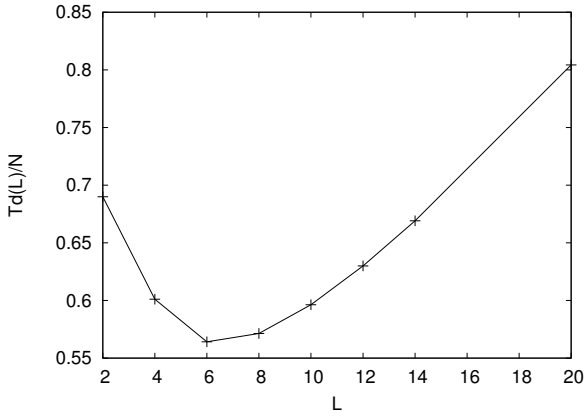


Fig. 8. The effect of neighbor number ($N = 200$)

words, a peer can identify a free-rider through a finite number of unchokes and hence effectively prevent free-riding. Note that in the original algorithm, a free-rider can continuously download through optimistic unchoking since peers don't use history information to identify free-riders.

4. Stochastic Model

In Section 3.2, we have briefly discussed that our algorithm can improve the downloading performance and prevent free-riding. However, in a real network, peers may dynamically join and leave the system, which will make the history information less accurate and hence may affect the performance of our algorithm. In this section, we propose a stochastic model to analyze this issue.

For the simplicity of analysis, we will first make some assumptions. We assume all normal peers have the same upload rate u. Each peer has a fixed number of neighbors. When a neighbor leaves the system, a new peer will be added to make the number of neighbors a constant L. Note that this is a reasonable assumption because BitTorrent has a mechanism to request for new peers from the tracker if some of its neighbors leave the system. We also assume that a free-rider can be identified through one optimistic unchoke. This is a simplification of the case that a free-rider can be identified by a finite number of unchokes as we have discussed in Section 3.2.

For a given peer, we define the following terms.

X The number of known normal peers in the neighborhood.

Y The number of known free-riders in the neighborhood.

γ The leaving probability of a peer in a given time slot.

p The probability that a newly-arrived peer is a free-rider.

Note that since the total number of peers in the neighborhood is L, there will be $L - X - Y$ peers that we don't know if they are normal peers or free-riders. These peers are either newly-arrived or peers that we have never unchoked before.

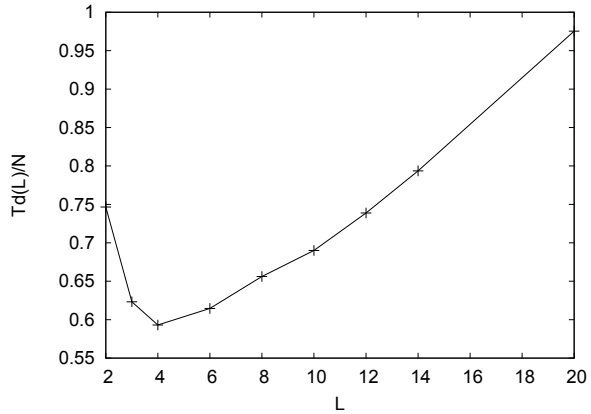


Fig. 9. The effect of neighbor number ($N = 100$)

Now under the assumptions, the random process of (X, Y) becomes a two-dimensional Markov chain. At each time slot, three type of events may occur. First, some known normal peers may leave the system. Let N be the the number of those peers. Then N follow a Binomial distribution with parameters X and γ. Secondly, some known free-riders may also leave the system. Let M be the number of those peers. Then M is also a Binomial random variable with parameters Y and γ. Finally, if $X + Y < L$, that means there are unknown peers in the neighborhood, following our optimistic unchoking algorithm, one of those unknown peers will be chosen to be unchoked and hence will be identified as either a normal peer or a free-rider. Let V be the random variable that indicate that the unknown peer is identified as a free-rider. Then $P\{V = 1\} = p$ and $P\{V = 0\} = 1 - p$.
So in a summary, if the Markov chain is in the state (X, Y) and $X + Y < L$, the probability that the state jumps to $(X - n + (1 - v), Y - m + v)$ in the next time slot is

$$\begin{aligned} q_{n,m,v} &= \mathrm{P}\{N = n, M = m, V = v\} \\ &= \binom{X}{n}\gamma^n(1-\gamma)^{X-n}\binom{Y}{m}\gamma^m(1-\gamma)^{Y-m}p^v(1-p)^{1-v}. \end{aligned}$$

If $X + Y = L$, the probability that the state jumps to $(X - n, Y - m)$ is

$$\begin{aligned} q_{n,m} &= \mathrm{P}\{N = n, M = m\} \\ &= \binom{X}{n}\gamma^n(1-\gamma)^{X-n}\binom{Y}{m}\gamma^m(1-\gamma)^{Y-m}. \end{aligned}$$

The Markov chain is a complicated two-dimensional chain and it is hard to get a closed-form solution for the steady-state distribution. However, given the jumping probability, we can easily find the numerical solution for the steady-state distribution. Let $\pi(x, y)$ be the steady-state probability that the system is in state (x, y). If we assume that unchoking a normal peer will always get response (i.e., the normal peer will also upload to us), the average download

rate a normal peer can obtain will then be

$$D = u \sum_{x=1}^{L} \pi(x, L-x) + u(1-p)(1 - \sum_{x=0}^{L} \pi(x, L-x))$$

The first term corresponds to the case that all peers are known. So the peer can unchoke a normal peer and hence obtain download rate u. The second term corresponds to the case that there are unknown peers. So a unknown peer will be unchoked and our expected download rate from the unknown peer is $u(1-p)$.

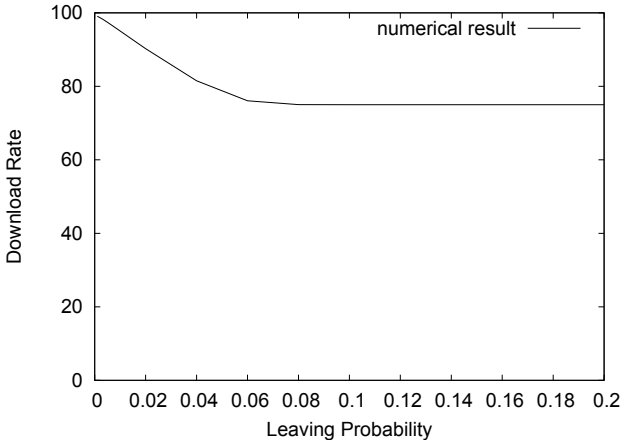


Fig. 10. The average download rate of a normal peer

In Fig. 10, we show the numerical result of the average download rate as a function of the peer leaving probability. Here we set $L = 20$, $u = 100$Kbps, and $p = 0.25$. We see that when γ is small, the neighborhood doesn't change frequently and the download rate is close to the up limit 100Kbps. However, when the leaving probability increases, the performance of our algorithm decrease and eventually the download rate becomes 75Kbps, which we can see as the worst case of our algorithm. Note that even the worst case of our algorithm has the same performance as the original optimistic unchoking algorithm, in which, peers are randomly unchoked and hence 75% of unchoked peers are normal ones.

4.1 Simulation Results

We have also done extensive simulations to evaluate the proposed algorithm. In the first simulation, we simulate a BitTorrent network to verify our stochastic model. We set the same parameters as in the numerical result with $L = 20$, $u = 100$Kbps, and $p = 0.25$. The file of interest is 10M bytes. Each peer in the network will randomly choose $L = 20$ as its neighbors when it joins the network. Peers arrive following a *Poisson process* with parameter $\lambda = 5.0$ and peers in the network leave randomly according to *Exponential departure* with parameter γ. In our simulation, when a peer finishes its download, it becomes a seed and provide uploading to others.

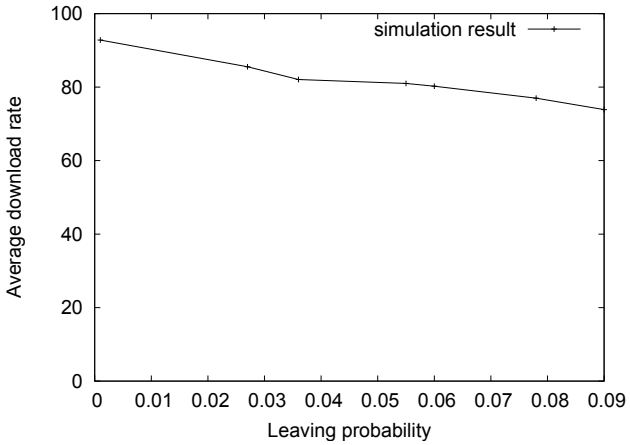


Fig. 11. Simulation: the average download rate of a normal peer

In Fig. 11, we can see that as the leaving probability γ grows, the average download via optimistic unchoking decreases similar to the numerical result. The high leaving probability means that a peer will use optimistic unchoking to test its new neighbor more often. Due to the possibility that newly-arrived peers may be free-riders, a normal peer who is already in the network may lose its chance to get the download via the optimistic unchoking. That is the reason why the average download via the optimistic unchoking process goes down as the leaving probability γ goes up. We can also see that in the simulation, the performance decreases more quickly than in the numerical result. This is because in the stochastic model, we assume that a free-rider can be detected with just one unchoke. While in real BitTorrent networks, it may need several unchokes to identify a free-rider.

In the second simulation, there are $L = 30$ peers in a neighborhood. All peers have the same upload bandwidth 100KB per time slot and among these peers, 50% are free-riders. In Fig. 12, we show the total download of a free-rider as a function of time. We can see that with the original algorithm, a free-rider can continuously obtain data from the network through optimistic unchoking. While with the proposed algorithm, the data a free-rider can download from the network is significantly reduced and hence our algorithm can effectively prevent free-riding. In Fig. 13, the total download of a normal peer through optimistic unchoking is shown as a function of time. We see that with the new algorithm, the download a normal peer can get from optimistic unchoking is almost doubled. So our algorithm can also improve the performance of normal peers significantly.

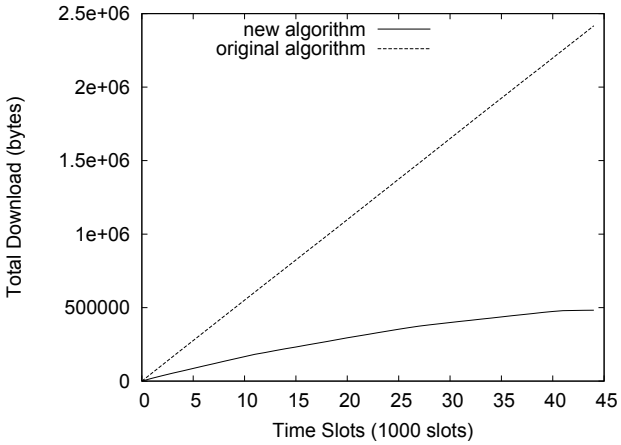


Fig. 12. The total download of a free-rider

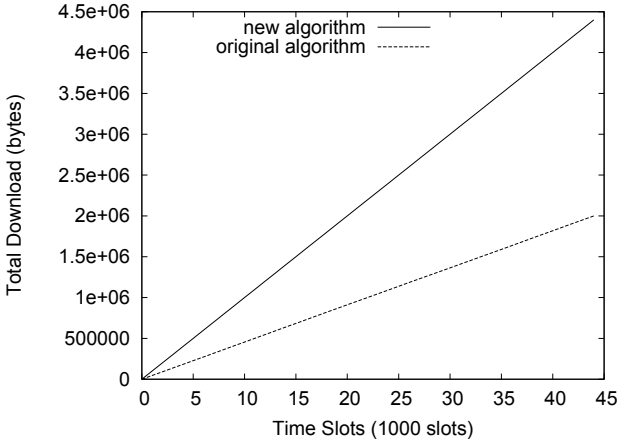


Fig. 13. The total download of a normal peer

4.2 Conclusion

In this section, we proposed a novel optimistic unchoking algorithm for BitTorrent. The basic idea is to take advantage of peer information obtained from past experience and try to unchoke the peer with the highest expected gain. Our theoretical analysis and simulation results show that the new algorithm can significantly improve the performance of normal peers and at the same time, effectively prevent free-riding.

One of the advantages of our algorithm is that it can be deployed progressively. A peer can use our algorithm to improve its performance no matter other peers use the same optimistic unchoking algorithm or not. In addition, our algorithm can work with any incentive mechanisms proposed for BitTorrent and hence it can be easily integrated into BitTorrent networks.

5. References

Clevenot, F. & Nain, P. (2004). A Simple Fluid Model for the Analysis of the Squirrel Peer-to-Peer Caching System, *Proceedings of IEEE INFOCOM*.

Clevenot, F., Nain, P. & Ross, K. (2005). Stochastic Fluid Models for Cache Clusters, *Performance Evaluation* **59**(1): 1–18.

Cohen, B. (2003). Incentives Build Robustness in BitTorrent, *Proceedings of the First Workshop on the Economics of Peer-to-Peer Systemas*, Berkeley, CA, USA.

de Veciana, G. & Yang, X. (2003). Fairness, incentives and performance in peer-to-peer networks, *the Forty-first Annual Allerton Conference on Communication, Control and Computing, Monticello, IL*.

Ge, Z., Figueiredo, D. R., Jaiswal, S., Kurose, J. & Towsley, D. (2003). Modeling peer-peer file sharing systems, *Proceedings of IEEE INFOCOM*.

Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X. & Zhang, X. (2005). Measurements, analysis, and modeling of bittorrent-like systems, *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement (IMC)*, pp. 35–48.

Ma, Z. & Qiu, D. (2009). A Novel Optimistic Unchoking Algorithm for BitTorrent, *Proceedings of Consumer Communications and Networking Conference*, Las Vegas, NV, USA.

Ng, T. S. E., Chu, Y.-H., Rao, S. G., Sripanidkulchai, K. & Zhang, H. (2003). Measurement-Based Optimization Techniques for Bandwidth-Demanding Peer-To-Peer Systems, *Proceedings of IEEE INFOCOM*.

Piatek, M., Isdal, T., Anderson, T., Krishnamurthy, A. & Venkataramani, A. (2007). Do incentives build robustness in bittorrent?, *Proc. of USENIX Symposium on Networked Systems Design & Implementation*.

Qiu, D. & Srikant, R. (2004). Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks, *Proceedings of ACM SIGCOMM*.

Ripeanu, M. (2001). Peer-to-peer architecture case study: Gnutella network, *Technical report*, University of Chicago.

Ripeanu, M., Foster, I. & Iamnitchi, A. (2002). Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design, *IEEE Internet Computing Journal* **6**(1).

Susitaival, R. & Aalto, S. (2006). Modelling the population dynamics and the file availability in a bittorrent-like p2p system with decreasing peer arrival rate, *International Workshop on Self-Organizing Systems (IWSOS)*, pp. 34–48.

Susitaival, R., Aalto, S. & Virtamo, J. T. (2006). Analyzing the dynamics and resource usage of p2p file sharing by a spatio-temporal model., *International Conference on Computational Science (4)*, pp. 420–427.

Tian, Y., Wu, D. & Ng, K. W. (2006). Modeling, Analysis and Improvement for BitTorrent-Like File Sharing Networks, *Proceedings of IEEE INFOCOM*.

Yang, X. & de Veciana, G. (2004). Service Capacity of Peer to Peer Networks, *Proceedings of IEEE INFOCOM*.

Zhang, H., Neglia, G., Towsley, D. & Presti, G. L. (2007). On unstructured file sharing networks, *Proceedings of IEEE INFOCOM*.

A study on Garbage Collection Algorithm for Ubiquitous Real-Time System

Chang-Duk Jung, PhD; You-Keun Park, PhD
Korea University

Most parallel garbage collection algorithms are based on the *mark-and-collect* technique. a mark-and-collect technique an effective asynchronous marking algorithm. There are two basic marking techniques: coloring and stacking. The coloring technique is asynchronous but its time complexity is O(MN) where M and N are the total number of nodes in the list memory and the total number of active nodes, respectively. The stacking technique offers effective marking process having only O(N) time complexity but requires extra stack space which can be as large as the size of entire active nodes (N). A new parallel garbage collection algorithm in ubiquitous environment has been devised which takes advantage of the asynchronous processing of coloring algorithms and the time efficiency of stacking algorithms. The algorithm requires no synchronization between the collectors and the mutators. and its tome complexity is close to O(N) with a small fixed-size stack in ubiquitous real-time system.

Key Words: parallel garbage collection, real-tome, autonomous memory

1. Introduction

Recent interest in using artificial intelligence for time-critical ubiquitous real-time systems controlling physical devices such as dynamically unstable airplanes, nuclear reactors, and robots demands a large search space operating in the manner such that it *never runs out of space* [1] because these applications are required to create and to release large amounts of data continuously without interruption.

Lists are the fundamental data structure in artificial intelligence programs. List processing systems free the programmer from managing the memory storage. Instead, the list processing facility maintains a set of *free* memory cells (called nodes) and dynamically collects garbage memory nodes which are no longer accessible by the program into the set of free memory nodes. Thus, in time-critical real-time list systems, a *real-time garbage collection system* is a must.

With the emergence of multiple processor systems, parallel garbage collection algorithms have been proposed where more than one processor, called the collectors, exclusively collect garbage concurrently with the activity of other processors, called the mutators, which are dedicated only to the application processes [3, 2, 4]. The mutators proceed with the list processing activity while the collectors reclaim the garbage nodes concurrently. Thus, the mutator and the collector should operate independently of each other.

Recently, Shin [7] developed a parallel garbage collection algorithm using associative tags. His algorithm requires no synchronization between the collectors and the mutators. The algorithm is based on the *mark-and-collect* technique which consists of three phases.
In the *initializing phase*, every node except free nodes is initialized as a garbage node. Every reachable node is marked during the marking phase as in active node, starting form a set of root nodes. Nodes which remain as garbage nodes are reclaimed in the *reclaim phase.* The time complexity of Shin's marking process is O(N) and is does not require extra stack memory.
Although Shin's algorithm can be used for implementing a massively parallel garbage collection system, his algorithm requires associative tags which are expensive to construct.
we have developed a parallel garbage collection algorithm in ubiquitous real-time system which is linearly scalable and operates asynchronously; thus it is possible to construct a massively parallel garbage collection system for a large time-critical real-time system.
In section 2, we introduce general parallel garbage collection algorithms. We present a new parallel garbage collection algorithm suitable for implementing the autonomous memory in Section 3. In Section 4, The proof of correctness of the algorithm is given and the time complexity of the algorithm is presented in Section 5. Conclusions are drawn in section6.

2. Garbage Collection in Ubiquitous real-time system

There are two basic techniques for parallel garbage collection: copying [6] and mark-and-collect techniques[7, 5]
The copying technique employs the principle of *freezing* the memory at the instant in time at which the copying begins and preserving the list structure for the collectors.
There are two basic methods in the copying technique based on how the copying process is performed: duplication methods [8] and evacuating methods [6].
The duplication method is required to make a copy of a part or of the entire list memory before beginning the garbage collection precess. The mutator proceeds with the list processing activity while the collector reclaims the garbage nodes from the unchanging second copy of the list structure.
The evacuation method divides the available memory into two logical space: *newspace* and *oldspace*. A garbage collection cycle starts with a *flip*, in which newspace is converted to oldspace, and vice-versa. The mutator proceeds with the list processing activity in newspace while the collector copies (*evacuates*) all accessible nodes in oldspace into newspace by tracing them from a set of roof nodes. After completion of the evacuation precess, oldspace contains only garbage nodes and may be transferred to free nodes.
The copying garbage collection technique provides several advantages including the ability to reclaim circular structures and to compact the storage which reduces the storage fragmentation and improves locality of reference. The copying garbage collectors have been widely used in a virtual memory environment. However, this technique not only requires extra memory space which can be as large as the requires that list processing activity by the mutator be suspended while the mutator makes a copy of the list structures (duplication method) or the mutator be suspended while the collector must be tightly synchronized for a flip operation (evacuation method). Thus, the copying technique is not suitable for implementing a massively parallel garbage collection system.
Most parallel garbage collection algorithms are based on the mark-and-collect technique. The mark-and-collect technique requires tag bits. In general, a mark-and-collect algorithm

consists of there phases: initialization, marking, and collection, In the initialization phase, non-free nodes are initialized ad garbage nodes for the subsequent marking process. In the marking phase, all active nodes are marked b tracing form a set of root nodes, All the nodes that have been neither marked nor already declared as free nodes are transformed into free nodes in the collection phase. There are two basic methods for the mark-and-collect technique: coloring [7] methods and stacking [5] methods.

The coloring method starts by initializing all non-free nodes as garbage nodes in the initialization phase. Marking process begins by marking all root nodes. Then, the collectior finds a marked node and marks all its immediate descendants. The procedure continues until there is no marked node with unmarked immediate descendants. Only nodes remaining in garbage status after the marking process are reclaimed in the collection phase. Since the mutator never changes the status of nodes to garbage status, this method does not require the suspension of the mutator. Thus, the coloring method can be used to implement a parallel garbage collection system. However, the time complexity off the marking process is O(MN) where M and N are the total number of nodes in the list memory and the total number of active nodes, respectively[1].

The stacking method offers efficient marking requiring only O(N) time at the expense of extra memory space for the stack which can be ad large as the size of the set of entire active nodes(N). Also, the access to the stack must be done through the critical section since the mutator and the collector both need to update the stack.

These algorithms are based on the following explicit or implicit assumptions.

1. The marking process is terminated before the mutator exhausts free nodes. Wadler presented sufficient conditions for this assumption in terms of the maximum rate at which free nodes are used, the maximum number of nodes in use at one time, and the total number of nodes in the system, which may be hard to determine. [1] However, in general, the average time needed for the creation of a list element is required to be greater than the average time interval between two consecutive instances of marking a node to guarantee proper termination of the marking process.

2.1 A set to root nodes for all active lists is provided for the collectors

In general, foot nodes are maintained by the mutator in special memory blocks such as internal registers or stack buffers. All accessible nodes, only those active nodes, are referencable via some paths from the nodes in these special memory blocks.

The requirement that the average time needed to create a list element be greater than the average time interval between two consentive instances of marking a node is particularly important because it limits the speed of the mutator relative to the speed of the collector. The most time-consuming process of the coloring method is the marking process, which is O(MN), because the worst time interval between two consecutive instances of marking a node is the tine required to search the whole memory. Shin introduced an algorithm using associative tags which provides a constant one-unit time for searching a marked node. [7] thus, the marking algorithm has time complexity O(N), which is optimal. However, it requires associative tags which are expensive.

3. Algorithm Model in Ubiquitous Real-time System

We present a marking algorithm whose time complexity is close to O(N) by combining the coloring and stacking methods. The algorithm is very similar to the Shin's algorithm[7].
Before proceeding with the algorithm statement, let's present the system model. In our system model, there is a collector for each memory block. The system maintains a separate set of rot nodes for each memory block which is an array of special pointers, ROOT(1), ROOT(2),....,ROOT(R). They contain the pointer to the root nodes of lists residing in the same memory block.
The system also maintains a separate free list for each memory block. There are two pointers to the free list: one to the head of the free list (F-HEAD) from which the mutator accesses the free list and the other to the tail of the free list (F-TALK) to which the collector appends free nodes. The advantage of this organization is that it gives better locality of reference; thus it minimizes the communication between collectors.
Identification of garbage nodes is achieved by marking all active nodes. However, when a mutator redirects an existing pointer which the collector has already marked to another active node which has not yet been marked, problems may occur. The conventional solution of these problems in to let the mutator mark the redirected node which has not yet been marked. This is one of the reasons that the conventional stacking method is required to maintain the backtracking pointers of more than one list which makes the stack size as big as the size of the set of entire active nodes. These problems are alleviated in the proposed system by maintaining a special list, called the *replaced node list* which contains the pointers of redirected nodes. Unlike a conventional list, the replaced node list is only accessible from the tail of the list. However, the first node of the list is always created at a fixed location. The system maintains a special record called R-POINT for the replaced node list. There are two fields in R-POINT and R-TAIL. The R-FLAG (value 1) indicates the recreation of the replaced node list; thus the first node will be created at the fixed location (R-NODE). A node of the replaced node list contains two fields: R-ROOT and R-PREV. R-ROOT contains the pointer of the redirected node. R-PREV contains the pointer of the previous replaced node except in the first node where the value is NIL.
There is a foxed-size small stack. The stack is organized as a last-in first-out circular queue. Thus, the stack maintains the most recent backtracking pointers up to the stack size(S).
The algorithm in ubiquitous real-time system is based on tagged memory. There are two tag bits. Thus, there are four possible states for a node, depending on the values of the tag bits () as followings:

1. (0,0) - F-state: The nodes in the F-state are free nodes (in the free list) which are available for the mutator to create a new list element.
2. (1,0) - G-state: The nodes in the G-state are garbage nodes. Nodes which are not free nodes (not in the F-state) are initialized as garbage nodes (the G-state). The nodes remaining in the G-state after the marking process are garbage nodes which can be transferred to the free list.
3. (1,0) - A-state: The nodes in the A-state are active nodes.
4. (1,1) - N-state: The nodes in the N-state are new nodes which are created during the current garbage collection cycle. A new node may or may not be a garbage node.

If a new node is create and released during the same garbage collection cycle, it is called a *floating* node. The algorithm collects floating nodes in the next garbage collection cycle. Thus, garbage nodes are guaranteed to be collected in two cycles.

The algorithm starts by setting the value of R-FLAG to 1 and then initializing all non-free nodes to garbage nodes (G-state) including the root nodes. After completion of the initialization, the marking process begins.
The nodes are traced starting from ROOT(i) for i = 1 to R. If the status of ROOT(i) is the G-state, the collector marks the nodes from ROOT(i) in a depth-first order. If any of its descendants nodes are in the G-state, is converts its state to the A-state. The algorithm uses the stack for storing backtracking pointers. Although the stack is small and fixed, the collector operates as if there is unlimited stack memory because the stack is organized as a last-in first-out circular queue. However, the stack maintains only the most recent backtracking pointers up to the size of the stack (S). After the stack is empty, the collector reexamines the states of the immediate descendants of ROOT(i). If any of them are still in the G-state, the collector retraces the list starting from ROOT(i).
This procedure is repeated until none of the immediate descendants of ROOT(i) is in the G-state, at which time the collector converts the state of ROOT(i) to the A-state and proceeds to the next ROOT(i). after the marking of the last ROOT(R), the collector traces the replaced node list backward from the tail (R-TAIL) using R-PRVE until the value of R-PREV is NIL. The collector marks the redirected nodes by tracing them from R-ROOT using the same procedures as above. The marking process terminates when the collector has examined all replaced nodes.

A: Initialization Phase in ubiquitous real-time system
Set R-FLAG to 1.
For i = 1 to M, of NODE(i) is not in the F-stage,
initialize it to the G-state.

B: Marking Phase in ubiquitous real-time system

M1: For all i = 1 to R, if ROOT(i) is in G-state,
let ROOT(i) be all parent node,

1. If LP = NIL of the state of NODE(LP) is cot G-state, go to step 3.
2. Push the parent node onto the stack and let NODE(LP) be the new parent node. Repeat step 1.
3. If RP = NIL or the state of NODE(LP) is not G-state, go to step 5.
4. Push the parent node onto the stank and let NODE(NODE(RP)) be the new parent node. Repeat step 1.
5. Mark (convert the modes from G-state to A-state) the parent node. Pop the stack. If the stack is empty, go to step 6. otherwise let the popped node be the new parent node. Repeat step 1.
6. Let ROOT(i) be a parent node. If (LP = NIL or NODE(LP) is not in G-state) and (RP = NIL or NODE(LP) is not in G-state), mark the ROOT(i); otherwise, repeat step 1.

M2: If R-FLAG = 1, skip the M2 procedure.
Let NODE(R-TALK) be a working node (R-N ODE) and NODE(R-ROOT) be a parent node.

1. If LP = NIL or the state of NODE(LP) is not G-state. go to step 3.
2. Push the parent node onto the stack and let NODE(LP) be a new parent node. Repeat step 1.
3. If RP = NIL of the state of NODE(RP) is not G-state, go to step 5.
4. Push the parent node onto the stack and let NODE(NODE(RP)) be a new parent node. Repeat step 1.
5. Mark (convert the nodes from G-state to A-state) the parent node. pop the stack. If the stack is empty, go to step 6, otherwise let the popped node be a new parent node. Repeat step 1.
6. Let NODE(R-ROOT) be a parent node. If (LP = NIL of NODE(LP) is not in G-state) and (RP = NIL or NODE(RP) is not in G-state), mark the NODE(R-ROOT), otherwise repeat step 1.
7. If R-PREV is not NIL, let NODE(R-PREV) be a working node (R-NODE). Let NODE (R-ROOT) be a parent node. Repeat step 1.

C: Collection Phase in ubiquitous real-time system

C1: *For all i = 1 to M, if the stare of NODE(i) is* G-state, append it to the free list after converting its state to F-state

C2: Set R-FLAG to 1.
The only function required in the mutator for garbage collection is to create the replaced node list whenever the mutator redirected a node which is still in the G-state to a node which is in the A-state.

1. If R-FLAG = 1, then reset it to 0 and create the first R-NODE at the special location, otherwise create an R-NODE at any location. Update the R-TAIL with the pointer of the newly created R-NODE.
2. If R-NODE is a frist node, move the NIL value to R-PREV, otherwise move R-TALK to R-PREV

3. Move the pointer to the replaced node to
   R-ROOT

Fig. 2. Mutator Algorithm in ubiquitous real-time system

After completion of the marking process, nodes still remaining in G-state are garbage nodes. The collector converts these nodes to free nodes (F-state) and appends them to The tail of the free lost.
The only function required for the mutator related to garbage collection is the creation of the replaced node list. Whenever the mutator redirects a node which is still in G-state to a node which is in A-state, it creates an R-NODE and places the pointer of the redirected node into R-ROOT and updates R-POINT. However, if the value of R-FLAG is 1, the mutator resets R-FLAG to 0 and creates the first node at the fixed location (R-NODE).
The algorithms of the mutator and the collector are summarized in Figure 1 and Figure 2. The algorithm assumes that each node contains two pointers (LP and RP) to other nodes in the list structure. However, the algorithm can be modified easily to handle list structure having more than two pointers. NODE(LP) and NODE(RP) represent the child nodes pointed to by the pointers LP and RP of the parent node, respectively.

4. Correctness of the algorithm in ubiquitous real-time system

For brevity, we present a brief summary of the proof of correctness of the algorithm. To prove the correctness of the garbage collection algorithm, we need to show:

C1. Invariance: The active list structures should not be modified by the garbage colector

C2. Termination: The garbage collection processes terminate properly.

C3. Validity: No active nodes should be mistaken for garbage nodes.
The following theorems prove the correctness of the algorithm.

Theorem 1: The algorithm satisfies the Invariance condition.

Proof: The collector does not alter any pointer of lists except in the collection phase in which it only alters the pointer of inactive nodes (garbage nodes).

Lemma 2: The Initialization Phase terminates properly.
Lemma 3: The Collection Phase terminates properly.
Proof: The initialization and collection phases involve only the sequential scanning of the memory form top th bottom; they will terminate properly.

Lemma 4: The procedure M1 terminates properly.
Proof: after the initialization phase, all nodes are either in F-state, G-state, or N-state. The nodes in N-state are newly created nodes after the initialization process began. However, after the initialization process, neither the collector nor the mutator changes the state of

nodes to G-state. Thus, there are a fixed number of nodes in G-state. The procedure M1 converts the state of nodes from G-state to A-state; thus, it must terminate properly.

Lemma 5: The procedure M2 terminates properly.
Proof: The problem of redirecting nodes occurs only when the mutator redirects nodes which have not been marked to a node which already has been marked, Thus, after completion of the procedure M1, the redirecting of nodes will not cause a problem. Therefore, redirected nodes pointed to by R-NODE which is created after R-TAIL need not be examined. Since the collector traces the replaced node list backward from R-TAIL to the first R-NODE, it will terminate properly if the redirected lists are not expanded. After the completion of the procedure M1, all active nodes are either marked to A-state or in G-state. If the active node still is in G-state, it must be a member of a redirected list.
However, since there is only a finite number of nodes in G-state, the procedure M2 will terminate properly
Theorem 6: The marking phase terminates properly.
Proof: The Theorem is true by Lemma 4 and Lemma 5.

Theorem 7: The garbage collection processes terminate properly.
Proof: It follow from Lemma 2, Lemma 3, and Theorem6

Theorem 8: NO active nodes should be mistaken for garbage nodes.
Proof: The mutator only converts the state of nodes from F-state to N-state. The collector only initializes non-free nodes in the initialization phase. The active nodes in G-state must be marked by either the procedure M1 or M2. The collector only collects the nodes still remaining in G-state. Thus, no active nodes are mistaken for garbage nodes.

From the above theorems, We conclude that the algorithm is correct. However, the algorithm does not collect all garbage nodes in one cycle. Nodes created and released during the cycle(floating nodes) are collected in the next cycle. Thus, the algorithm guarantees that garbage nodes are collected within two cycles.

5. Time Complexity in the Ubiquitous Real-time System

The time complexity of the algorithm cam be estimated as follows. Assume that the distribution of the depth of lists is a normal distribution function with the mean μ and the variance. Since the time complexity of the initialization phase and the collection phase is O and the variance <IMG src=".\PIC9B.gif" width=13px height=16px >. Since the time complexity of the initialization phase and the collection phase is O an s□n<SPAN STYLE='font-family:"바탕";font-size:13px;color:"#000000";line-height:20px;text-align:justify;'>The time complexity of the algorithm cam be estimated as follows. Assume that the distribution of the depth of lists is a normal distribution function with the mean μ and the variance <IMG src=".\PIC9B.gif" width=13px height=16px >. Since the time complexity of the initialization phase and the collection phase is O and the variance <IMG src=".\PIC9B.gif" width=13px height=16px >. Since the time complexity of the initialization phase and the collection phase is O an s＞n&lt

where S is the size of the stack.

If we assume that the density function is normal distribution, then or by letting we have (1) where
Thus, the time complexity of the algorithm is worse than that of the stack algorithms by a factor of (2)
This is the typical time and space tradeoff. However, the time penalty (Equation 2) is considerably smaller than the spatial gain. If the stack size is, the time complexity of the algorithm is just 16% more than the optimum. If the stack size is, the time complexity of the algorithm is almost optimum (1.001 times of the optimum). The algorithm is not dependent on M or N; thus the time complexity of the algorithm is O(1).

Algorithm	Stacking	Coloring	shin[7]	New Algorithm
Time	O(N)	O(MN)	O(N)	O(N)
Tag Bits	1	2	2	2
Assoc. Memory	no	no	no	no
Critical Section	stack	none	none	none
Extra Space	O(N)	none	none	O(1)

Table 1. A performance comparison of marking algorithms for garbage collection.

This is a substantial improvement over stacking algorithms which require the stack space as big as the entire available memory (M) or coloring algorithms whose time complexity is O(MN).

6. Conclusions

We have presented a parallel garbage collection algorithm in ubiquitous real-time system which can be used for artificial intelligent systems for time-critical real-time applications. The algorithm takes advantage of the time efficiency of stacking algorithms and the space efficiency of coloring algorithms.
The algorithm requires no critical section and the time complexity of its marking process is close to O(N) with a small fixed-size stack. Thus, a massively parallel garbage collection system can be effectively constructed.
In this paper, we didn't discuss the case when a list is expanded form one memory bank to other memory bank for brevity. In this case, an address fault would occur and the collector needs to send the address of the children to other collectors. A receiving collector is required t create a temporary list similar to the replaced node list. At the end of the tracing of the replaced node list, the collector marks the active nodes whose roots are located in other memory blocks by tracing the temporary list.
To construct a massively parallel garbage collection system. we need to design a communication network to interconnect collectors. A communication system called a

has been developed to interconnect multiple processors to from a massively parallel

computer at the Aerospace Technology Center of the allied-Signal Aerospace company [10]. The communication network exhibits a high degree of connectivity, modularity, extensibility, and scalability. The same module would be used for constructing a massively parallel garbage collection in ubiquitous real-time system.

7. Reference

[1] Richard Jones, "Garbage Collection", wiley and Sons, 2006
[2] Bill Venneers, " Java's Garbage collection Heap", Javaworld, August, 2005
[3] N.I.A Woodward, "Alternative approaches to multiprocessor garbage collection", Proc. Int. Conf. Parallel processing, 2004, pp. 205-210
[4] Amsaleg, P. Freerira,M. Fracklin and M. Shapiro, "Evaluating Garbage Collection for Large Persistent Stores", Proceedings of the 1995 OOPSLA Workshop on Object Database Behavior, Benchmarks, and Performance, Austin, Texas, October 2002.
[5] Y. Hibino, "A practical parallel garbage collection algorithm and its implementation", Proc. Proc. Annual Symp. on Computer Architecture, 2001, pp. 113-120
[6] S. G. Abraham and J. H. Patel, "Parallel Garbage Collection on A Virtual Memory System", Proc. Int'l Conf. Parallel Processing, 1987, pp. 243-246
[7] H. Shin, "A Boolean content addressable memory and its applications", Univ. of Texas, Austin, May 1986
[8] R. J. P. Ingria and M. Cohen, "LAMBDA release 3.0 Notes", LISP Machine Inc., Oct., 1986
[9] D. W. Clark and C. C. Green, "An empirical study of list structure in Lisp", comm. ACM 20, 2 (Feb. 1977), pp. 78-86
[10] You-Keun Park, et all, "Distributed Associative Processor" Proc. to ACM Computer Science Conference, Louisville, Kentucky, February, 1989

23

Spam Mail Blocking in Mailing Lists

Kenichi Takahashi[1], Akihiro Sakai[12] and Kouichi Sakurai[12]
[1]*Institute of Systems, Information Technologies and Nanotechnologies*
[2]*Faculty of Information Science and Electrical Engineering, Kyushu University*
Japan

1. Introduction

The increasing popularity of the Internet has led to e-mail becoming one of the widely used essential services in our personal and business life. However, in the recent times, the number of spam mails received has increased rapidly. Symantec has reported that spam mails account for up to 75% of all e-mails (Symantec, 2008). Such a large amount of spam mails waste the valuable time of e-mail users who have to filter out these spam mails. Moreover, the large number of spam mails may prevent users from seeing important mails. Some spam mails may contain viruses, worms, or links to a phishing site and cause information leakage, loss of information, invasion of privacy, and damage to computers.

Therefore, many researchers have proposed and elaborated on several techniques for dealing with spam mail. Spam filtering is one of the widely used techniques against spam mails (Tabata, 2006). Spam filters infer whether a mail is a spam mail or not on the basis of certain preserved keywords in the mail. However, spam filtering techniques also give rise to false positive and false negative results. For example, a large number of spam mails include the word "Viagra." Therefore, a mail including Viagra will usually be assumed to be spam. However, users belonging to a drug company may usually use Viagra in their mails. In such cases, spam filters may produce false positive results. Moreover, AT&T's anti-anti-spam patent (Pfleeger and Bloom, 2005) reveals that the current spam filtering techniques are actually a cat-and-mouse game with spammers.

Whitelisting/blacklisting is one of the techniques used for blocking spam mails. An administrator notes down accepted/rejected mail senders and/or domains in a whitelist/blacklist. Then, mails from non-accepted/rejected mail senders and domains are blocked. However, we cannot expect to know everyone who would e-mail. Hence, it is difficult to specify mails from which mail senders and/or domains can be accepted/rejected. In this situation, we forgo blocking spam mails for our personal mails, but we focus on blocking spam mails in a mailing list. A large number of spam mails also come through in mailing lists. Mailing lists are used to disseminate information within specific groups such as laboratory staff, a project group, or users sharing the same interests. The members of a mailing list share a common mailing list address. When a member sends a mail to the mailing list address, the mail is automatically forwarded to all the mailing list members. Since the members of a mailing list share a common mailing list address, the probability of address leakage increases with an increase in the number of members. Meanwhile, the

mailing list address is shared only among mailing list members and should be used only for mailing to the mailing list. This means that the mailing list address should not be used for online shopping, user registration, etc. Thus, we can assume that the mailing list address is used only for posting mails to mailing list. Hence, changing of a mailing list address is easier than changing personal mail addresses because the influence of the change is limited to the mailing list members. However, frequent changes to the mailing list address are not acceptable since such changes would affect all of the mailing list members.

In this chapter, we introduce a system to block spam mails in a mailing list. In this system, a mailing list system assigns different posting mail addresses to different mailing list members. A mailing list member sends a mail to the posting mail address assigned to him/her in order to send a mail to the mailing list. When a spam mail is received, the posting mail address leading to the receiving of the spam mail is identified and invalidated, and a new posting mail address is assigned to the member. Thus, we can block the spam mails coming from the invalidated address, but the member can post a mail to the mailing list from the new posting mail address. Furthermore, our system is highly compatible with the typical mail systems because our system does not require any particular software to be installed on the client machines.

The remainder of this chapter is organized as follows. The next section analyzes the causes that lead to the receiving of spam mails. Section 3 introduces some techniques against spam mails such as spam filtering, blacklisting/whitelisting and so on. Section 4 introduces our mailing list system for blocking spam mails, and section 5 presents an evaluation of this system. Finally, in section 6, the chapter is concluded.

2. Causes of spam mails

We analyze the causes of spam mails in mailing lists. Generally, we have to know recipient addresses to send an e-mail; thus, spammers also have to obtain recipient addresses for sending spam mails. However, a mailing list addresses is necessary to be shared only by the mailing list members; hence, it should remain unknown to users (e.g. spammers) other than the mailing list members. Therefore, ideally, spammers should not be able to send spam mails to mailing list addresses, as these addresses are unknown to them. However, in fact, a large number of spam mails come through mailing lists. This is because of the leakage of the mailing list addresses. We can classify the causes of address leakage into the following cases (Figure 1).

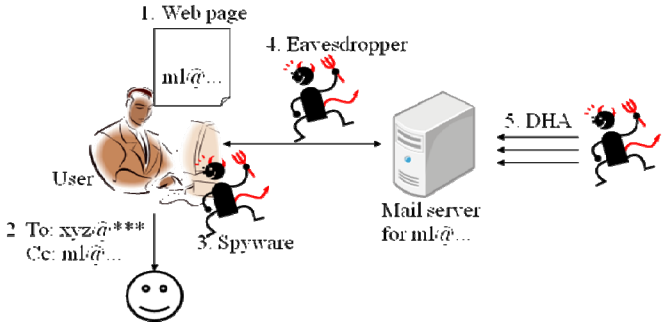


Fig. 1. Causes of address leakage

1. A member puts the mailing list address on his Web pages. Then, a Web crawler collects the address. Alternatively, a member uses the mailing list address for online shopping user registration, etc.
2. A member sends an e-mail to both the mailing list and a non-member of the mailing list. Then, the non-member leaks the mailing list address.
3. A mailing list member uses a machine that has spyware installed. The spyware collects the mailing list address and leaks it to spammers.
4. There is an eavesdropper in a channel between a member and the mailing list server. Alternatively, when a member posts an e-mail to a mailing list from an un-trusted mail server, the un-trusted mail server leaks the mailing list address.
5. An attacker sends a mail to random addresses, and a mailing list address is unfortunately included in the addresses. Then, the attacker remembers the address as a valid address (DHA: Directory Harvest Attack).

In all of the abovementioned cases except for case 5, mailing list address leakage is caused by a mailing list member. A spammer collects a mailing list address leaked by these causes and sends spam mails. Once the address is leaked to a spammer, the address is exchanged between spammers. This leads to an increase in the number of spam mails. Thus, it has become impossible to stop spam mails.

3. Techniques against spam mails

3.1 Spam filtering

Spam filtering techniques are based on the differences in the characteristics of spam mails (Tabata, 2006) and legitimate mails. Some of these techniques use statistical classification based on machine learning. Most of them are Bayesian-based filters. These techniques learn and maintain words that are frequently used in spam mails and legitimate mails. Then, if a mail contains many words used in spam mails, the mail is judged as a spam mail. Further, there is a research on classifying spam mails according to the routing information (Fujii, 2004). However, these techniques produce false positive and false negative results.

3.2 Whitelisting and blacklisting

Whitelists and blacklists are an effective tool for blocking spam mails. They work well in the case of a mailing lists with a limitted posting methods, such as the mailing list of a magazine. Otherwise, it is difficult to list all the acceptable senders and/or domains because the administrator may not know which mail servers are used by the members. For example, it is difficult for a system administrator to list all the temporary methods that a user may use to post a mail, such as Gmail, Yahoo! Mail, cell phones or alternate SMTP servers. Otherwise, the usability will decrease because the administrator has to refuse these methods.

DNSBL (DNS-based Blackhole List) (dnsbl, 2008) provides the list of IP addresses which spam mails are sent frequently. For example, the Spamhaus Project (http://www.spamhaus.org/) provides such a list. Hence, we can filter spam mails according to the list. However, DNSBL is not precise; even legitimate mail servers such as those belonging to universities and government agencies are sometimes registered in this list. Then, the system filters out even legitimate mails. Moreover, DNSBL is not compatible with dynamic IP address systems.

3.3 Sender ID framework

Sender ID Framework (SIDF) (SIDF, 2008; Wong & Schlitt, 2006) is one of whitelisting approach. The overview of SIDF is shown in figure 2. In SIDF, the sender's domain appends *SPF records* in its DNS server. An SPF record contains the list of IP addresses and hostnames that are permitted to send mails from this domain. Then, receivers can check whether a mail is permitted or not by the sender's domain by the following steps:

1. Extracting sender's domain information from mail headers
2. Obtaining SPF records from the sender's domain
3. Checking whether the sender's IP address is listed in the SPF records

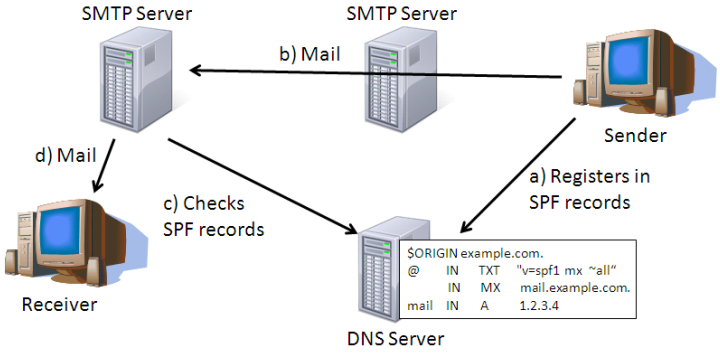


Fig. 2. Overview of Sender ID Framework

Thus, if the IP address is not listed in SPF records, the receiver judges the mail is spam because the SMTP server does not allow a mail from the IP address. However, this technique requires SPF records registered in the DNS server, installation of specific libraries to check the SPF records, and the cooperation of the Internet service provider. Moreover, mail forwarding used in mailing lists may result in false positives because the forwarding server is not listed in the sender's SPF records.

3.4 DomainKeys Identified Mail

DomainKeys Identified Mail (DKIM) (DKIM, 2008; Allman et al., 2007) is an approach to verify a sender's address on the basis of a digital signature. In DKIM, a public key to verify the sender's digital signature is prepared in the DNS server of the sender's domain. When a sender sends a mail, the sender's SMTP server automatically appends a digital signature to the mail header. Then, a receiver verifies the digital signature by using the public key on the sender's DNS server (figure 3). The following steps are carried out by the receiver to know whether the mail is spam or not.

1. Extracting the sender's domain information and digital signature
2. Obtaining a public key of the sender
3. Verifying the digital signature by the public key

Thus, if the verification is passed, the receiver is convinced that the mail is sent from a legitimate sender. However, this technique has the same problems as SIDF; it requires a list of public keys registered in the DNS server, installation of specific libraries to verify digital signatures, and has a mail fowarding problem.

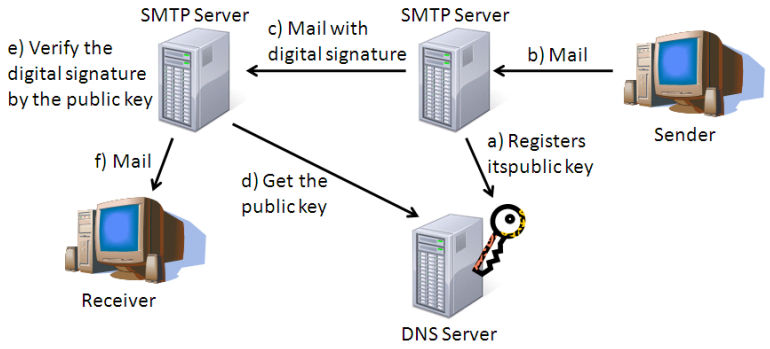


Fig. 3. Overview of DomainKeys Identified Mail

3.5 Other techniques

We can use disposal e-mail address (DEA) (Seigneur & Jensen, 2003) against spam mails. The representative DEA service providers are Spamex (http://www.spamex.com/) and myTrashMail.com (http://mytrashmail.com/). It is, however, not compatible with mailing list systems.

Spammers usually send a large number of spam mails. Therefore, some researchers have attempted a approach where a task is imposed on the mail sender (Dwork & Naor, 1993; Roman et al., 2005). In such an approach, when a receiver receives an e-mail from a sender, the receiver sends an some easy question back to the sender. If the sender does not give the correct answer, the receiver judges the mail as spam. The approach prevents spammers from sending a large number of spam mails because the computation power of the spammers to make the answer is limitted. (Kraut et al., 2005) and (Kuipers et al., 2005) have proposed charging mail senders some money instead of having a computation task. However, these approaches are not applicable to mailing list systems. In addition, the load of spam senders is also imposed on the legitimate e-mail senders. Moreover, we would then have to replace the current mail systems with their new versions, which seems to be impossible.

Privango (Takahashi et al., 2005) uses an encrypted condition as a mail address. When a user receives a mail that does not match the condition in the mail address, the system automatically rejects it. However, it is difficult for us to remember the mail address because the mail address is like a random string sequence.

4. Personal mailing list address to block spam mails

We propose a system to block spam mails in a mailing list; in this system, we assign different addresses to different mailing list members.

4.1 Requirements

Various techniques have been employed to block spam mails, such as SIDF and DKIM. However, these techniques require the cooperation of Internet service providers and the installation of particular software on the client machines. Therefore, these techniques cannot be easily applied to the current mail systems. We need to develop a technique that is

compatible with the typical mail systems. Thus, a system should satisfy the following requirements for compatibility with the typical mail systems:

- Users need not install particular software. Thus, they should be able to use their customized e-mail client systems.
- The technique should be compatible with the current standard mail protocols such as POP and SMTP.
- It should be easy to use with the typical mailing list systems. Thus, the following requirements should be satisfied:
  - Users can make use of an address, which is easy to remember, for mailing to a mailing list;
  - Users can mail to a mailing list from any mail address and any server, such as Gmail, Yahoo! Mail, and their own home mail servers;
  - Users can reply to a mailing list by using the same methods as those used in the typical mailing list systems.

Our system tries to block spam mails by invalidating the address which leads to receiving spam mails. Therefore, the system needs to identify such addresses. However, the identification of the member causing spam mails may be disadvantaged. Moreover, address leakage caused by DHA happens without any relation to members' carelessness. Therefore, we require the following:

- When a spam mail is received, the mailing list members should not be able to identify the member whose address led to the receiving the spam mails.
- The system should be able to distinguish an address leakage caused by DHA from other cases.

We have to develop a system that can block spam mails and also satisfy the abovementioned requirements.

4.2 Proposed system

In the typical mailing list systems, all mailing list members use the same mailing list address. This makes it difficult to change the mailing list address. Therefore, we assign a different address, named *individual address*, to each mailing list member. Then, each member sends mails to his/her assigned individual address. Thus, the individual addresses can be easily changed because each member uses a different individual address. The overview of our proposed system is shown in figure 4.

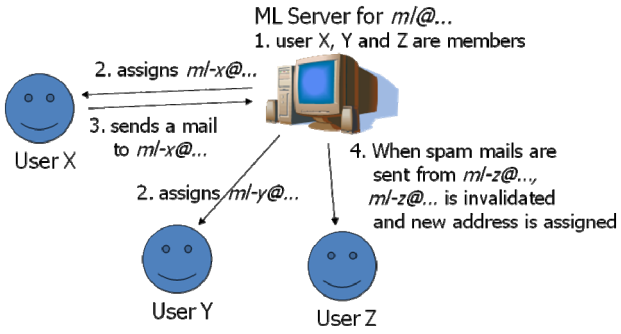


Fig. 4. Overview of proposed system

1. The administrator decides the mailing list members.
2. Then, the random individual addresses are created and assigned to each member. Here, each member can change his/her individual address to his/her customized address by using the additional steps.
3. A member sends a mail to his/her individual address.
4. When spam mails increase, the individual address that leads to the receiving of the spam mails is invalidated and a new individual address is assigned.

In this approach, each member uses a different individual address. Therefore, the invalidation of an individual address does not affect the other members. In other words, the system can block spam mails by changing the address of only the member causing the spam mails.

4.2.1 Creation of mailing list

A mailing list is created by an administrator. First, the administrator decides the address of the mailing list, named *ML address*, the members and the configuration of the mailing list. Note that the ML address is not used for the sending mails to the mailing list. The ML address may be used only in the To field of mails sent from the mailing list. The system ignores direct mails to the ML address. The configuration defines which address should be used in To, Reply-To and From fields, and a penalty to a member who leads to the receiving of spam mails. Then, the system creates an *Address-ML* table, an *Address-Sender* table, a *Penalty* table, and a *History DB*. The Address-ML table records the individual addresses along with the ML address. The Address-Sender table records the individual address and the mail address of a member to whom the individual address is assigned. The Penalty table manages the penalty values of each member, which increases when a spam mail is received. The History DB is a database that manages all the mails sent to the mailing list.

Next, random individual addresses are created, and these are recorded in the Address-ML and the Address-Sender table. Then, an invitation mail is sent out to each member.

When a member receives the invitation mail, he/she can send a reply to the mail including his/her customized address. If the customized address is already used by other member, an acceptance fail mail is sent to the member. Then, the member can try his next customized address. If the attempt is successful, the Address-ML and Address-Sender table are overwritten by the customized address (this will be his/her individual address) and an acceptance mail is sent to the member. As a result, the user can use his customized address, which the user can easily remember, for sending a mail to the mailing list.

Similarly, we can add other members after the mailing list is created.

4.2.2 Sending a mail to a mailing list

A member can send a mail to a mailing list by sending a mail to his/her individual address. The From, To and Reply-To fields of a mail forwarded to the mailing list members are defined by the configuration of each mailing list. Here, we assume that a ML address, a sender's address, and an address for replying to the mailing list are specified in To, From, and Reply-To fields, respectively (Figure 5).

When a mail is received by the mailing list, a pair of the mail and its Message-Id is recorded in the History DB. Next, the ML address corresponding to the To field (sender's individual address) of the mail is determined by the Address-ML table. Then, the To field is changed to

the ML address. After that, if and only if the mail has a Reply-To field, the From field is changed to the address of the Reply-To field. Finally, mails that are forwarded to each mailing list members are created by changing the Reply-To fields to the individual addresses corresponding to each member by using the Address-Sender table. Then, the mails are forwarded to the mailing list members.

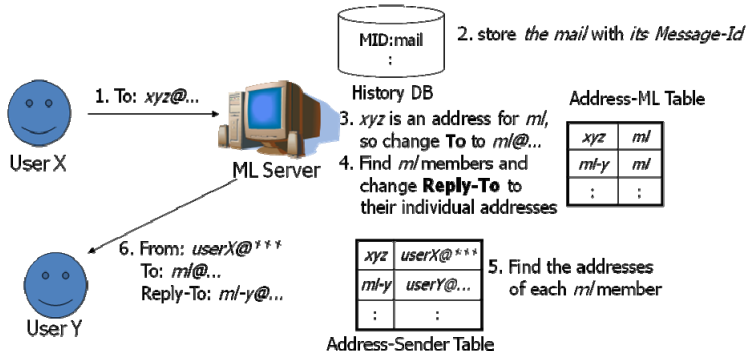


Fig. 5. Flow of sending mail to mailing list

4.2.3 Reply to mailing list

The Reply-To field in a mail sent to a member is the individual address of the member. Therefore, the members can reply to the mailing list in the same way as that used in the typical mailing lists. The flow when the system receives a reply mail is the same as that of a mail sent to a mailing list

4.2.4 When spam mail is received

Mailing list members judge whether the received mail is spam mail or not. Figure 6 shows the flow when a spam mail is received.

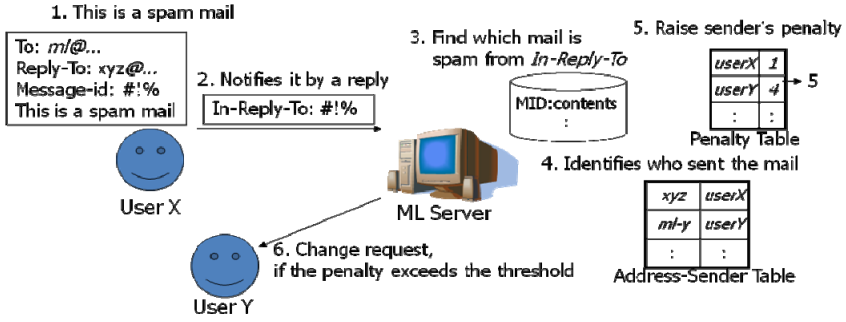


Fig. 6. Flow when spam mail is received

When a member receives a mail, he/she judges whether it is a spam mail or not. If it is a spam mail, he/she replies with a *spam report* mail. Then, by referring to the History DB, the system identifies the mail judged as spam mail from the In-Reply-To field. Next, the

individual address that has been used to send the mail is determined. Then, the penalty value of the member using the individual address is increased. Note that it is better that the penalty value is increased only when several notices are received because of a mistaken spam report.
When the penalty value of a member exceeds the threshold, a *change request* mail to change the individual address is sent to the member. The change request includes one or more candidates of the cause of the individual address leakage described in section 2.
We also can consider the use of a spam filtering tool. The filtering tool will decide whether a mail is a spam mail or not. Thus, spam mails will be blocked without sending them to the mailing list members; however, we have to pay attention to false positive and negative results.

4.2.5 Identify an address leakage caused by DHA

An address leakage caused by DHA happens without relation to members' carelessness. Therefore, we want to distinguish an address leakage caused by DHA from other cases. In DHA, an attacker will usually send spam mails to not only individual addresses but also unavailable addresses. Therefore, the system can identify DHA by monitoring the spam mails sent to unavailable addresses. Thus, dummy addresses that are similar to individual addresses are set as monitoring addresses. For example, if an individual address is xxx1@..., xxx0@... and xxx2@... are set as monitoring addresses. When a spam mail is sent not only to xxx1@... but also to xxx0@... and xxx2@..., the spam mail can be assumed to be caused by DHA. Then, even if a member receives a change request mail, he/she can know the address leakage is not caused by his/her failure. This will avoid unnecessary trouble from the change request mail.

4.2.6 Other functions

Request of a Posting History: A member can obtain his/her sent mail recorded in History DB by sending a *summary* mail to his individual address.
Withdrawal from Mailing List: When a member wants to withdraw from a mailing list, he/she sends a *bye* mail to his individual address. Then, his/her information from the Address-ML, Address-Sender, and Penalty tables is deleted, and an acceptance mail is sent to him/her.
Change of Individual Address: When a member wants to change his/her individual address, he/she sends a *change address* mail along with his/her customized address. The flow is the same as a part of the flow given in section 4.2.1.
These are optional functions, and the administrator of the mailing list can disable them.

5. Evaluations

5.1 Simulation results

We simulate the number of spam mails blocked by utilizing our system. In this simulation, we use the following configurations.

- One year is 360 days; thus, one month is 30 days.
- The individual addresses of each member have leaked out in the probability of 1/360 per day.

- Once an individual address has leaked, a spammer sends an average of one spam mail per day.
- The number of the mailing list members is **M**.
- A typical mailing list (not our mailing list system) changes its mailing list address at the end of each year.
- In our system, when each member receives a spam mail, he/she sends a spam report in the probability of **P**.
- Once our system receives **N** spam reports for a certain individual address, the individual address is changed.

Figure 7 shows the number of spam mails when the number of mailing list members is different, where **P** = 5% and **N** = 3.

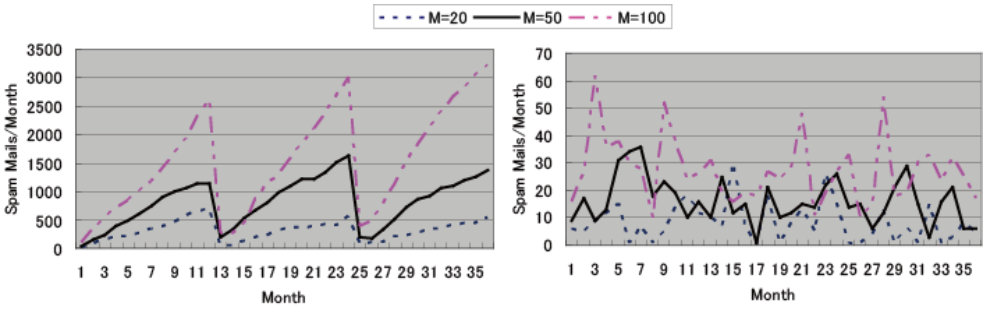


Fig. 7. Number of spam mails for different numbers of mailing list members. The left graph is the result for typical mailing list; the right graph is for our system.

In a typical mailing list, the number of spam mails increase throughout the year, and spam mails become 0 at the end of the year because of the change of mailing list address. When **M** = 20, 10.5 spam mails per day are received on an average; for **M** = 50, 27 spam mails; and for **M** = 100, 51 spam mails. However, in the case of our system does not increase the number of spam mails throughout the year because our system changes the individual addresses leaked to spammers once some spam mails (reports) are received. When **M** = 20, 0.28 spam mails per day are received on an average; for **M** = 50, 0.54 spam mails; and for **M** = 100, 0.91 spam mails. As just described, our system dramatically decreases the number of spam mails. From the viewpoint of address changes, the typical mailing list requires the change of the mailing list address every year so that the mailing list imposes the task of M person-time address changes. Our systems also requires about M person-time address changes (a little smaller than M in theory) because each member's individual address is leaked once a year on an average. Thus, the burden of each member to address changes is almost the same in both the typical and our mailing list systems. The simulation result is also almost the same as shown in table 1.

	M = 20	**M** = 50	**M** = 100
Typical mailing list	20	50	100
Proposed system	18	52.4	98.6

Table 1. Number of address changes in a year (Average of three years)

Moreover, typical mailing list systems do not identify members who leak mailing list addresses. Therefore, a careless member who leaks an address probably does not pay attention to the address leakage. On the other hand, in our system, the careless members receive a change request mail, and thus, they can pay attention to the management of their individual addresses. Thus, our system will work more effectively to reduce spam mails than the typical mailing list systems because each member will be more careful about protecting his/her individual address from leakage.
Next, we show the number of spam mails when the probability **P** is different, where **M** = 50 and **N** = 3 (figure 8).

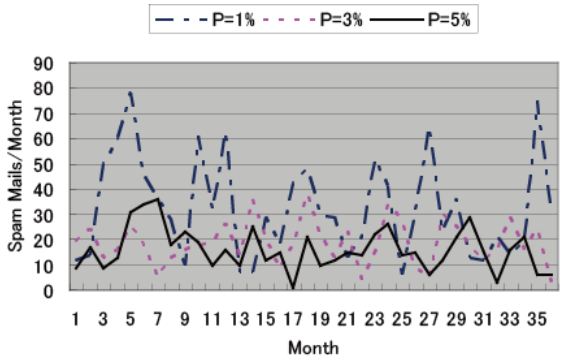


Fig. 8. Number of spam mails of our system for different probabilities of spam report

As the probability increases, the system receives spam reports with high frequency when a spam mail received. Thus, the sooner an individual address is changed, the sooner the spam mails stop. In the simulation result, even if **P** = 1% (only one person sends a spam report when hundred people receive a spam mail), 1.1 spam mails per day are received on an average. As shown here, our system can decrease the number of spam mails even when only a few members are cooperative.
The number of spam mails received when the value of **N** is different is shown in figure 9, where **M** = 50 and **P** = 5%.

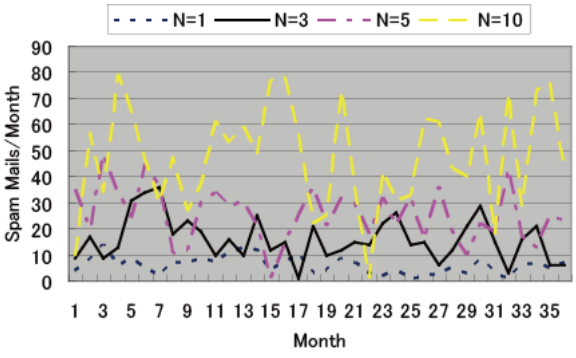


Fig. 9. Number of spam mails when **N** spam reports cause an individual address change

The lower the value of **N**, the sooner will be the individual address change. However, **N** = 1 may cause a mistaken address change, for example, even when a member mistakes a legitimate mail as a spam mail, the individual address is changed. Therefore, **N** should have an appropriate value to prevent this problem. In the simulation result, when **N** = 1; 0.20 spam mails are received per day; for **N** = 3, 0.54 spam mails; for **N** = 5, 0.85 spam mails; and for **N** = 10, 1.59 spam mails. We believe that **N** = 3 and counting spam reports for only one month would be the practical configuration.

5.2 Effectiveness in the difference in cause of individual address leakage

Figure 10 shows the causes of individual address leakage.

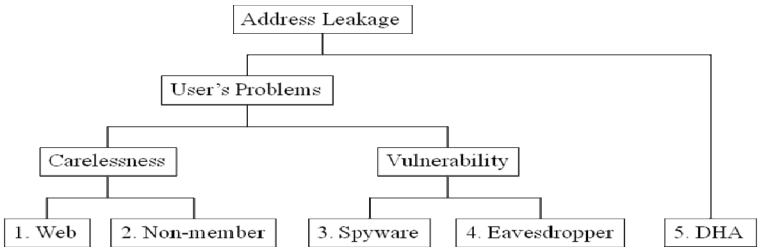


Fig. 10. Causes of individual address leakage

Cases 1 to 4 are caused by the carelessness of the mailing list members or the vulnerability of their machines/environments. Therefore, it may be difficult to identify the cause from these 4 cases.

Cases 1 and 2 are caused by the carelessness of the mailing list members. Therefore, the spam mails will be blocked once the individual address is invalidated since the carelessness is usually not happened frequently. In case 1, the individual address leaks from a Web page. Therefore, this cause cannot be identified by the system. In case 2, the cause can be identified if the non-member's address is written in the To and Cc fields, but not if the non-member's address is written in the Bcc field. In these two cases, the member can identify the cause by investigating his/her Web page and mail log. Then, the member can avoid such carelessness in future.

Case 3 is caused by spyware installed on the member's machine. If the spyware learns the steps involved in our system, it may execute the process of new address assignment instead of the member. Then, spam mails cannot be blocked by invalidating the individual address. Furthermore, if the spyware automatically deletes the change request mails, the member will not even notice the deletions. Therefore, if the spam mails do not stop despite changing the member's individual address several times, the administrator must inform the member that his/her address has been leaked through other means such as a telephone call. Then, the member should obtain his posting history using another machine and check it. If the member finds mails that he/she did not send/receive, it can be assumed that his/her machine is infected with a spyware.

Case 4 also cannot be solved by the invalidation of an individual address if an attacker is continuously tapping mails. In this case, the encryption of mails can solve the problem. However, members have to install software for mail encryption. This will decrease the usability of the system since members cannot easily send mails to a mailing list from Web

mails or other machines that do not have the software installed. Otherwise, we can take countermeasures similar to those in case 3.
Cases 3 and 4 are caused by the vulnerability of the member's machine or the network environment. These will cause not only spam mails but also other losses such as information leakage, deletion of important information, and stepping stone attacks. Therefore, it is advisable that the members check the vulnerability of their machines and network environments when spam mails do not stop.
On the other hand, case 5 is caused without members' carelessness. This attack is identified by using the method discussed in section 4.2.5 and does not happen frequently to same member's individual address. Therefore, spam mails will be blocked once the individual address is invalidated. Moreover, same spam mails are very like to be simultaneously posted to some members' individual addresses. Thus, the system may be able to notice the occurrence of DHA.

5.3 Usability

Our system does not require particular software to be installed on the client machines. Therefore, our system is compatible with the typical mail systems, including the protocols and mail client machines. Further, since the system does not restrict access by access controls, mailing list members can send mails to a mailing list from any mail address and any server, such as Gmail, Yahoo! Mail, and their own home mail servers. However, each member has to register his/her customized individual address. This does not seem to be much of inconvenience to the almost members because the system requires the registration only once. This would rather be useful for the members because they can use their customized individual addresses that the user can easily remember. On the other hand, this may be burdensome to a member who causes spam mails since he has to repeat this process several times. However, since he/she is the cause of spam mails, his/her discomfort seems not to be a problem. Thus, the system can stop spam mails to a mailing list but does not impose a burden on good mailing list members. Furthermore, the system is easy to use because it requires almost the same operation as the typical mailing list systems.

6. Conclusion

In this chapter, we have introduced a system to block spam mails in mailing lists, in which we assign different individual addresses to each mailing list member. When a spam mail is received, the individual address that is the cause for receiving the spam mail is identified and invalidated. Then, the spam mails from the individual address can be blocked. Furthermore, our system does not require the installation of any particular software on the client machines. Therefore, the system is highly compatible with typical mail systems. Further, mailing list members can send/reply to a mailing list by the same operation as that used in the typical mailing list systems. Evaluation results show our system is effective in reducing the number of spam mails in mailing lists.

7. Acknowledgment

This work was partially supported by the Telecommunications Advancement Foundation.

8. References

Allman, E.; Callas, J.; Delany, M.; Libbey, M.; Fenton, J. & Thomas, M. Domainkeys identified mail (DKIM) signatures, *RFC 4871*, http://www.ietf.org/rfc/rfc4871.txt, 2007.

DomainKeys Identified Mail (DKIM), http://www.dkim.org/, 2008.

DNS Blacklist (DNSBL), http://en.wikipedia.org/wiki/DNSBL, 2008.

Dwork, C. & Naor, M. Pricing via Processing or Combatting Junk Mail, *CRYPTO'92*, LNCS 740, pp. 137-147, 1993.

Fujii, M. A New Method of Spam Message Discrimination, *MS Thesis*, Waseda Univ., 2004.

Kraut, R.E.; Sunder, S.; Telang, R. & Morris, J. Pricing Electronic Mail to Solve the Problem of Spam, *Human-Computer Interaction*, Vol. 20, No. 1 & 2, pp. 195-223, 2005.

Kuipers, B.J.; Liu, A.X.; Gautam, A. & Gouda, M.G. Zmail: Zero-sum Free Market Control of Spam, *Proc. of the 25th IEEE International Conference on Distributed Computing Systems Workshop*, pp. 20-26, 2005.

Pfleeger, S.L. & Bloom, G. Canning Spam: Proposed Solutions to Unwanted Email, *IEEE Security & Privacy*, Vol. 3, No. 2, pp. 40-47, 2005.

Roman, R.; Zhou, J. & Lopez, J. Protection against Spam Using Pre-Challenges, *Proc. of 2005 IFIP International Information Security Conference*, pp. 281-293, 2005.

Seigneur, J. & Jensen, C.D. Privacy Recovery with Disposable Email Addresses, *IEEE Security & Privacy*, Vol. 1, No. 6, pp. 35-39, 2003.

Sender ID Framework (SIDF), http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx, 2008.

Symantec. The State of Spam Report, http://www.symantec.com/business/theme.jsp?themeid=state_of_spam, 2008.

Tabata, T. SPAM Mail Filtering: Commentary of Bayesian Filter, *Journal of Information Science and Technology Association*, Vol. 56, No. 10, pp. 464-468, 2006.

Takahashi, K.; Abe, T. & Kawashima, M. Stopping Junk Email by Using Conditional ID Technology: privango, *NTT Technical Review*, Vol. 3, No. 3, pp. 52-56, 2005.

Wong, M. & Schlitt, W. Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1, *RFC 4408*, http://www.ietf.org/rfc/rfc4408.txt, 2006.