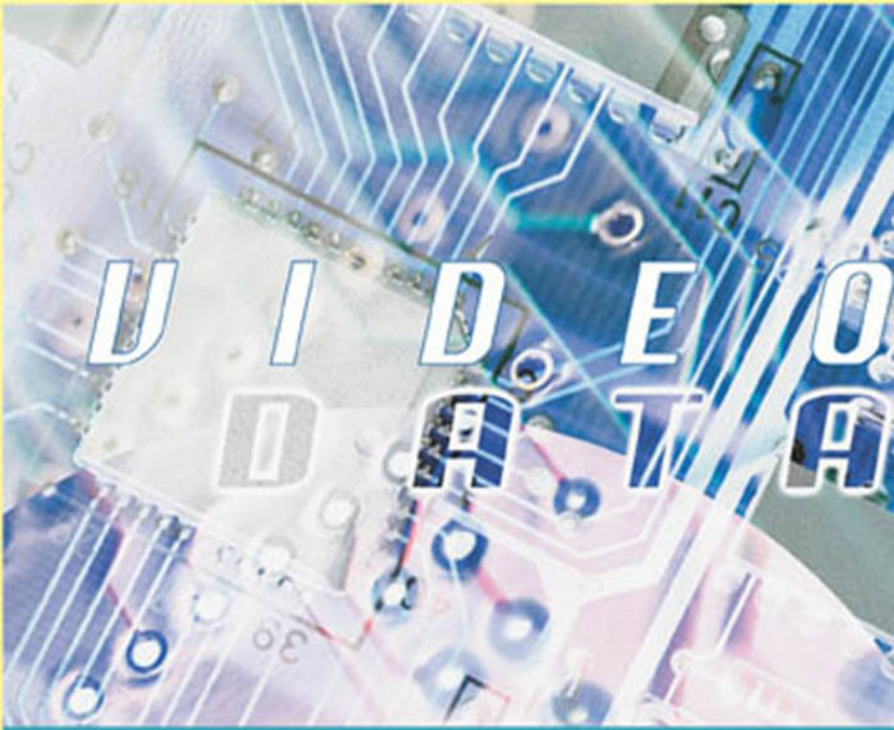
SAGARMAY DEB

VIDEO DATA MANAGEMENT AND INFORMATION RETRIEVAL



Video Data Management and Information Retrieval

Sagarmay Deb
University of Southern Queensland, Australia

IRM Press
Publisher of innovative scholarly and professional information technology titles in the cyberage

Hershey • London • Melbourne • Singapore

Acquisitions Editor:	Mehdi Khosrow-Pour
Senior Managing Editor:	Jan Travers
Managing Editor:	Amanda Appicello
Development Editor:	Michele Rossi
Copy Editor:	Jane Conley
Typesetter:	Jennifer Wetzel
Cover Design:	Lisa Tosheff
Printed at:	Yurchak Printing Inc.

Published in the United States of America by
IRM Press (an imprint of Idea Group Inc.)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033-1240
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@idea-group.com
Web site: http://www.irm-press.com

and in the United Kingdom by
IRM Press (an imprint of Idea Group Inc.)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 3313
Web site: http://www.eurospan.co.uk

Library of Congress Cataloging-in-Publication Data

Video data management and information retrieval / Sagarmay Deb, editor.
p. cm.
Includes bibliographical references and index.
ISBN 1-59140-571-8 (h/c) -- ISBN 1-59140-546-7 (s/c) -- ISBN 1-59140-547-5 (ebook)
1. Digital video. 2. Database management. 3. Information storage and retrieval
systems. I. Deb, Sagarmay, 1953-
TK6680.5.V555 2004
006--dc22
2004022152

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Video Data Management and Information Retrieval

Table of Contents

SECTION I:
AN INTRODUCTION TO VIDEO DATA MANAGEMENT AND INFORMATION RETRIEVAL

SECTION II:
VIDEO DATA STORAGE TECHNIQUES AND NETWORKING

Section III:
Video Data Security and Video Data Synchronization and Timeliness

Section IV:
Video Shot Boundary Detection

Section V: Video Feature Extractions

Section VI: Video Information Retrieval

Preface

INTRODUCTION

Video data management and information retrieval are very important areas of research in computer technology. Plenty of research is being done in these fields at present. These two areas are changing our lifestyles because together they cover creation, maintenance, accessing, and retrieval of video, audio, speech, and text data and information for video display. But still lots of important issues in these areas remain unresolved and further research is needed to be done for better techniques and applications.

The primary objective of the book is to combine these two related areas of research together and provide an up-to-date account of the work being done. We addressed research issues in those fields where some progress has already been made. Also, we encouraged researchers, academics, and industrial technologists to provide new and brilliant ideas on these fields that could be pursued for further research.

Section I gives an introduction. We have given general introduction of the two areas, namely, video data management and information retrieval, from the very elementary level. We discussed the problems in these areas and some of the work done in these fields since the last decade.

Section II defines video data storage techniques and networking**.** We present a chapter that describes the design for a High-performance Data Recording Architecture (HYDRA) that can record data in real time for large-scale servers. Although digital continuous media (CM) is being used as an integral part of many applications and attempts have been made for efficient retrieval of such media for many concurrent users, not much has been done so far to implement these ideas for large-scale servers. Then a chapter introduces video data management techniques for computational augmentation of human memory, i.e., augmented memory, on wearable and ubiquitous computers used in our everyday life. In another chapter, in order to organize and manipulate vast amount of multimedia data in an efficient way, a method to summarize these digital data has been presented. Also we present a contemporary review of the various different strategies available to facilitate Very Low Bit-Rate (VLBR) coding for video communications over mobile and fixed transmission channels as well as the Internet.

Section III talks about video data security and video data synchronization and timeliness. We describe how to present different multimedia objects on a web-based presentation system. A chapter is devoted to highlighting the biometrics technologies, which are based on video sequences, viz face, eye (iris/retina), and gait.

Section IV will present various video shot boundary detection techniques. A new robust paradigm capable of detecting scene changes on compressed MPEG video data directly has been proposed. Then an innovative shot boundary detection method using an unsupervised segmentation algorithm and the technique of object tracking based on the segmentation mask maps are presented. We also describe a histogram with soft decision using the Hue, Saturation, and Intensity (HSV) color space for effective detection of video shot boundaries.

Section V will throw light on video feature extractions. We address the issues of providing the semantic structure and generating abstraction of content in news broadcast.

Section VI covers video information retrieval techniques and presents an up-to-date overview of various video information retrieval systems. As the rapid technical advances of multimedia communication have made it possible for more and more people to enjoy videoconferences, important issues unique to personal videoconference and a comprehensive framework for indexing personal videoconference have been presented. Then we have dealt with video summarization using human facial information through face detection and recognition and also a discussion on various issues of video abstraction with a new approach to generate it.

The audience for this book would be researchers who are working in these two fields. Also researchers from other areas who could start-up in these fields could find the book useful. It could be a reference guide for researchers from other related areas as well. Reading this book can benefit undergraduate and post-graduate students who are interested in multimedia and video technology.

CHAPTER HIGHLIGHTS

In Chapter I, *Video Data Management and Information Retrieval,* we present a basic introduction of the two very important areas of research in the domain of Information Technology, namely, video data management and video information retrieval. Both of these areas still need research efforts to seek solutions to many unresolved problems for efficient data management and information retrieval. We discuss those issues and relevant work done in these two fields during the last few years.

Chapter II, *HYDRA: High-performance Data Recording Architecture for Streaming Media,* describes the design for a High-performance Data Recording Architecture (HYDRA). Presently, digital continuous media (CM) are well established as an integral part of many applications. In recent years, a considerable amount of research has focused on the efficient retrieval of such media for many concurrent users. The authors argue that scant attention has been paid to large-scale servers that can record such streams in real time. However, more and more devices produce direct digital output streams either over wired or wireless networks, and various applications are emerging to make use of them. For example, in many industrial applications, cameras now provide the means to monitor, visualize, and diagnose events. Hence, the need arises to capture and store these streams with an efficient data stream recorder that can handle both recording and playback of many streams simultaneously and provide a central repository for all data. With this chapter, the authors present the design of the HYDRA system, which uses a unified architecture that integrates multi-stream recording and retrieval in a coherent paradigm, and hence provides support for these emerging applications.

Chapter III, *Wearable and Ubiquitous Video Data Management for Computational Augmentation of Human Memory*, introduces video data management techniques for computational augmentation of human memory, i.e., augmented memory, on wearable and ubiquitous computers used in our everyday life. The ultimate goal of augmented memory is to enable users to conduct themselves using human memories and multimedia data seamlessly anywhere, anytime. In particular, a user's viewpoint video is one of the most important triggers for recalling past events that have been experienced. We believe designing an augmented memory system is a practical issue for real-world video data management. This chapter also describes a framework for an augmented memory album system named Scene Augmented Remembrance Album (SARA). In the SARA framework, we have developed three modules for retrieving, editing, transporting, and exchanging augmented memory. Both the Residual Memory module and the I'm Here! module enable a wearer to retrieve video data that he/she wants to recall in the real world. The Ubiquitous Memories module is proposed for editing, transporting, and exchanging video data via real-world objects. Lastly, we discuss future works for the proposed framework and modules.

Chapter IV is titled *Adaptive Summarization of Digital Video Data.* As multimedia applications are rapidly spread at an ever-increasing rate, efficient and effective methodologies for organizing and manipulating these data become a necessity. One of the basic problems that such systems encounter is to find efficient ways to summarize the huge amount of data involved. In this chapter, we start by defining the problem of key frames extraction then reviewing a number of proposed techniques to accomplish that task, showing their pros and cons. After that, we describe two adaptive algorithms proposed in order to effectively select key frames from segmented video shots where both apply a two-level adaptation mechanism. These algorithms constitute the second stage of a Video Content-based Retrieval (VCR) system that has been designed at Old Dominion University. The first adaptation level is based on the size of the input video file, while the second level is performed on a shot-by-shot basis in order to account for the fact that different shots have different levels of activity. Experimental results show the efficiency and robustness of the proposed algorithms in selecting the near optimal set of key frames required, to represent each shot.

Chapter V, *Very Low Bit-rate Video Coding*, presents a contemporary review of the various different strategies available to facilitate Very Low Bit Rate (VLBR) coding for video communications over mobile and fixed transmission channels and the Internet. VLBR media is typically classified as having a bit rate between 8 and 64Kbps. Techniques that are analyzed include Vector Quantization, various parametric model-based representations, the Discrete Wavelet and Cosine Transforms, and fixed and arbitrary shaped pattern-based coding. In addition to discussing the underlying theoretical principles and relevant features of each approach, the chapter also examines their benefits and disadvantages together with some of the major challenges that remain to be solved. The chapter concludes by providing some judgments on the likely focus of future research in the VLBR coding field.

Chapter VI is titled *Video Biometrics.* Biometrics is a technology of fast, user-friendly personal identification with a high level of accuracy. This chapter highlights the biometrics technologies that are based on video sequences viz face, eye (iris/retina), and gait. The basics behind the three video-based biometrics technologies are discussed along with a brief survey.

Chapter VII is titled *Video Presentation Model.* Lecture-on-Demand (LOD) multimedia presentation technologies among the network are most often used in many communications services. Examples of those applications include video-on- demand, interactive TV, and the communication tools of a distance learning system, and so on. We describe how to present different multimedia objects on a web-based presentation system. Using characterization of extended media streaming technologies, we developed a comprehensive system for advanced multimedia content production: support for recording the presentation, retrieving the content, summarizing the presentation, and customizing the representation. This approach significantly impacts and supports the multimedia presentation authoring processes in terms of methodology and commercial aspects. Using the browser with the Windows media services allows students to view live video of the teacher giving his speech, along with synchronized images of his presentation slides and all the annotations/comments. In our experience, this very approach is sufficient for use in a distance learning environment.

Chapter VIII is titled *Video Shot Boundary Detection.* The increasing use of multimedia streams nowadays necessitates the development of efficient and effective methodologies for manipulating databases storing this information. Moreover, content-based access to video data requires in its first stage to parse each video stream into its building blocks. The video stream consists of a number of shots; each one of them is a sequence of frames pictured using a single camera. Switching from one camera to another indicates the transition from a shot to the next one. Therefore, the detection of these transitions, known as scene change or shot boundary detection, is the first step in any video analysis system. A number of proposed techniques for solving the problem of shot boundary detection exist, but the major criticisms of them are their inefficiency and lack of reliability. The reliability of the scene change detection stage is a very significant requirement because it is the first stage in any video retrieval system; thus, its performance has a direct impact on the performance of all other stages. On the other hand, efficiency is also crucial due to the voluminous amounts of information found in video streams.

This chapter proposes a new robust and efficient paradigm capable of detecting scene changes on compressed MPEG video data directly. This paradigm constitutes the first part of a Video Content-based Retrieval (VCR) system that has been designed at Old Dominion University. Initially, an abstract representation of the compressed video stream, known as the DC sequence, is extracted, then it is used as input to a Neural Network Module that performs the shot boundary detection task. We have studied experimentally the performance of the proposed paradigm and have achieved higher shot boundary detection and lower false alarms rates compared to other techniques. Moreover, the efficiency of the system outperforms other approaches by several times. In short, the experimental results show the superior efficiency and robustness of the proposed system in detecting shot boundaries and flashlights (sudden lighting variation due to camera flash occurrences) within video shots.

Chapter IX is titled *Innovative Shot Boundary Detection for Video Indexing.* Recently, multimedia information, especially the video data, has been made overwhelmingly accessible with the rapid advances in communication and multimedia computing technologies. Video is popular in many applications, which makes the efficient management and retrieval of the growing amount of video information very important. To meet such a demand, an effective video shot boundary detection method is necessary, which is a fundamental operation required in many multimedia applications. In this chapter, an innovative shot boundary detection method using an unsupervised segmentation algorithm and the technique of object tracking based on the segmentation mask maps is presented. A series of experiments on various types of video are performed and the experimental results show that our method can obtain object-level information of the video frames as well as accurate shot boundary detection, which are both very useful for video content indexing.

In Chapter 10, *A Soft-Decision Histogram from the HSV Color Space for Video Shot Detection*, we describe a histogram with soft decision using the Hue, Saturation, and Intensity (HSV) color space for effective detection of video shot boundaries. In the histogram, we choose relative importance of hue and intensity depending on the saturation of each pixel. In traditional histograms, each pixel contributes to only one component of the histogram. However, we suggest a soft decision approach in which each pixel contributes to two components of the histogram. We have done a detailed study of the various frame-to-frame distance measures using the proposed histogram and a Red, Green, and Blue (RGB) histogram for video shot detection. The results show that the new histogram has a better shot detection performance for each of the distance measures. A web-based application has been developed for video retrieval, which is freely accessible to the interested users.

Chapter 11, *News Video Indexing and Abstraction by Specific Visual Cues: MSC and News Caption*, addresses the tasks of providing the semantic structure and generating the abstraction of content in broadcast news. Based on extraction of two specific visual cues — Main Speaker Close-Up (MSC) and news caption, a hierarchy of news video index is automatically constructed for efficient access to multi-level contents. In addition, a unique MSC-based video abstraction is proposed to help satisfy the need for news preview and key persons highlighting. Experiments on news clips from MPEG-7 video content sets yield encouraging results, which prove the efficiency of our video indexing and abstraction scheme.

Chapter XII is titled *An Overview of Video Information Retrieval Techniques*. Video information retrieval is currently a very important topic of research in the area of multimedia databases. Plenty of research has been undertaken in the past decade to design efficient video information retrieval techniques from the video or multimedia databases. Although a large number of indexing and retrieval techniques has been developed, there are still no universally accepted feature extraction, indexing, and retrieval techniques available. In this chapter, we present an up-to-date overview of various video information retrieval systems. Since the volume of literature available in the field is enormous, only selected works are mentioned.

Chapter XIII is titled *A Framework for Indexing Personal Videoconference*. The rapid technical advance of multimedia communication has enabled more and more people to enjoy videoconferences. Traditionally, the personal videoconference is either not recorded or only recorded as ordinary audio and video files, which only allow the linear access. Moreover, besides video and audio channels, other videoconferencing channels, including text chat, file transfer, and whiteboard, also contain valuable information. Therefore, it is not convenient to search or recall the content of videoconference from the archives. However, there exists little research on the management and automatic indexing of personal videoconferences. The existing methods for video indexing, lecture indexing, and meeting support systems cannot be applied to personal videoconference in a straightforward way. This chapter discusses important issues unique to personal videoconference and proposes a comprehensive framework for indexing personal videoconference. The framework consists of three modules: videoconference archive acquisition module, videoconference archive indexing module, and indexed videoconference accessing module. This chapter will elaborate on the design principles and implementation methodologies of each module, as well as the intra- and inter-module data and control flows. Finally, this chapter presents a subjective evaluation protocol for personal videoconference indexing.

Chapter XIV is titled *Video Abstraction*. The volume of video data is significantly increasing in recent years due to the widespread use of multimedia applications in the areas of education, entertainment, business, and medicine. To handle this huge amount of data efficiently, many techniques have emerged to catalog, index, and retrieve the stored video data, namely, video boundary detection, video database indexing, and video abstraction. The topic of this chapter is Video Abstraction, which deals with short representation of an original video and helps to enable the fast browsing and retrieving of the representative contents. A general view of video abstraction, its related works, and a new approach to generate it are discussed in this chapter.

In Chapter XV, *Video Summarization Based on Human Face Detection and Recognition*, we have dealt with video summarization using human facial information through the face detection and recognition. Many efforts of face detection and face recognition are introduced, based upon both theoretical and practical aspects. Also, we describe the real implementation of video summarization system based on face detection and recognition.

Acknowledgments

The editor would like to extend his thanks to all the authors who contributed to this project by submitting chapters. The credit for the success of this book goes to them.

Also sincere thanks go to all the staff of Idea Group Publishing for their valuable contributions, particularly to Mehdi Khosrow-Pour, Senior Academic Editor, Michele Rossi, Development Editor, and Carrie Skovrinskie, Office Manager.

Finally, the editor would like to thank his wife Ms. Clera Deb for her support and cooperation during the venture.

Sagarmay Deb
University of Southern Queensland, Australia

Section I

An Introduction to Video Data Management and Information Retrieval

Chapter I

Video Data Management and Information Retrieval

Sagarmay Deb
University of Southern Queensland, Australia

ABSTRACT

In this chapter, we present a basic introduction to two very important areas of research in the domain of Information Technology, namely, video data management and video information retrieval. Both of these areas need additional research efforts to seek solutions to many unresolved problems for efficient data management and information retrieval. We discuss those issues and relevant works done so far in these two fields.

INTRODUCTION

An enormous amount of video data is being generated these days all over the world. This requires efficient and effective mechanisms to store, access, and retrieve these data. But the technology developed to date to handle those issues is far from the level of maturity required. Video data, as we know, would contain image, audio, graphical and textual data.

The first problem is the efficient organization of raw video data available from various sources. There has to be proper consistency in data in the sense that data are to be stored in a standard format for access and retrieval. Then comes the issue of compressing the data to reduce the storage space required, since the data could be really voluminous. Also, various features of video data have to be extracted from low-level features like shape, color, texture, and spatial relations and stored efficiently for access.

The second problem is to find efficient access mechanisms. To achieve the goal of efficient access, suitable indexing techniques have to be in place. Indexing based on text suffers from the problem of reliability as different individual can analyze the same data from different angles. Also, this procedure is expensive and time-consuming. These days, the most efficient way of accessing video data is through content-based retrieval, but this technique has the inherent problem of computer perception, as a computer lacks the basic capability available to a human being of identifying and segmenting a particular image.

The third problem is the issue of retrieval, where the input could come in the form of a sample image or text. The input has to be analyzed, available features have to be extracted and then similarity would have to be established with the images of the video data for selection and retrieval.

The fourth problem is the effective and efficient data transmission through networking, which is addressed through Video-on-Demand (VoD) and Quality of Service (QoS). Also, there is the issue of data security, i.e., data should not be accessible to or downloadable by unauthorized people. This is dealt with by watermarking technology which is very useful in protecting digital data such as audio, video, image, formatted documents, and three-dimensional objects. Then there are the issues of synchronization and timeliness, which are required to synchronize multiple resources like audio and video data. Reusability is another issue where browsing of objects gives the users the facility to reuse multimedia resources.

The following section, *Related Issues and Relevant Works*, addresses these issues briefly and ends with a summary.

RELATED ISSUES AND RELEVANT WORKS

Video Data Management

With the rapid advancement and development of multimedia technology during the last decade, the importance of managing video data efficiently has increased tremendously. To organize and store video data in a standard way, vast amounts of data are being converted to digital form. Because the volume of data is enormous, the management and manipulation of data have become difficult. To overcome these problems and to reduce the storage space, data need to be compressed. Most video clips are compressed into a smaller size using a compression standard such as JPEG or MPEG, which are variable-bit-rate (VBR) encoding algorithms. The amount of data consumed by a VBR video stream varies with time, and when coupled with striping, results in load imbalance across disks, significantly degrading the overall server performance (Chew & Kankanhalli, 2001; Ding Huang, Zeng, & Chu, 2002; ISO/IEC 11172-2; ISO/IEC 13818-2). This is a current research issue.

In video data management, performance of the database systems is very important so as to reduce the query execution time to the minimum (Chan & Li, 1999; Chan & Li, 2000; Si, Leong, Lau, & Li, 2000). Because object query has a major impact on the cost of query processing (Karlapalem & Li, 1995; Karlapalem & Li, 2000), one of the ways to improve the performance of query processing is through vertical class partitioning. A detailed cost model for query execution through vertical class partitioning has been developed (Fung, Lau, Li, Leong, & Si, 2002).

Video-on-Demand systems (VoD), which provide services to users according to their conveniences, have scalability and Quality of Service (QoS) problems because of the necessity to serve numerous requests for many different videos with the limited bandwidth of the communication links, resulting in end-to-end delay. To solve these problems, two procedures have been in operation, scheduled multicast and periodic broadcast. In the first one, a set of viewers arriving in close proximity of time will be collected and grouped together, whereas in the second one, the server uses multiple channels to cooperatively broadcast one video and each channel is responsible for broadcasting some portions of the video (Chakraborty**,** Chakraborty, & Shiratori, 2002; Yang & Tseng, 2002). A scheduled multicast scheme based on a time-dependent bandwidth allocation approach, Trace-Adaptive Fragmentation (TAF) scheme for periodic broadcast of Variable-Bit-Rate (VBR) encoded video, and a Loss-Less and Bandwidth-Efficient (LLBE) protocol for periodic broadcast of VBR video have been presented (Li, 2002). Bit-Plane Method (BPM) is a straightforward method to implement progressive image transmission, but its reconstructed image quality at each beginning stage is not good. A simple prediction method to improve the quality of the reconstructed image for BPM at each beginning stage is proposed (Chang, Xiao, & Chen, 2002).

The abstraction of a long video is quite often of great use to the users in finding out whether it is suitable for viewing or not. It can provide users of digital libraries with fast, safe, and reliable access of video data. There are two ways available for video abstraction, namely, summary sequences, which give an overview of the contents and are useful for documentaries, and highlights, which contain most interesting segments and are useful for movie trailers. The video abstraction can be achieved in three steps, namely, analyzing video to detect salient features, structures, patterns of visual information, audio and textual information; selecting meaningful clips from detected features; and synthesizing selected video clips into the final form of the abstract (Kang, 2002).

With the enormous volume of digital information being generated in multimedia streams, results of queries are becoming very voluminous. As a result, the manual classification/annotation in topic hierarchies through text creates an information bottleneck, and it is becoming unsuitable for addressing users' information needs. Creating and organizing a semantic description of the unstructured data is very important to achieve efficient discovery and access of video data. But automatic extraction of semantic meaning out of video data is proving difficult because of the gap existing between low-level features like color, texture, and shape, and high-level semantic descriptions like table, chair, car, house, and so on (Zhou & Dao, 2001). There is another work that addresses the issue of the gap existing between low-level visual features addressing the more detailed perceptual aspects and high-level semantic features underlying the more general aspects of visual data. Although plenty of research works have been devoted to this problem to date, the gap still remains (Zhao et al., 2002). Luo, Hwang, and Wu (2003) have presented a scheme for object-based video analysis and interpretation based on automatic video object extraction, video object abstraction, and semantic event modeling .

For data security against unauthorized access and downloading, digital watermarking techniques have been proposed to protect digital data such as image, audio, video, and text (Lu, Liao, Chen, & Fan, 2002; Tsai, Chang, Chen, & Chen, 2002). Since digital watermarking techniques provide only a certain level of protection for music scores and suffer several drawbacks when directly applied to image representations of sheet music, new solutions have been developed for the contents of music scores (Monsignori, Nesi, & Spinu, 2003).

Synchronization is a very important aspect of the design and implementation of distributed video systems. To guarantee Quality of service (QoS), both temporal and spatial synchronization related to the processing, transport, storage, retrieval, and presentation of sound, still images, and video data are needed (Courtiat, de Oliveira, & da Carmo, 1994; Lin, 2002).

Reusability of database resources is another very important area of research and plays a significant part in improving the efficiency of the video data management systems (Shih, 2002). An example of how reusability works is the browsing of objects where the user specifies certain requirements to retrieve objects and few candidate objects are retrieved based on those requirements. The user then can reuse suitable objects to refine the query and in that process reuse the underlying database resources that initially retrieved those images.

Video Information Retrieval

For efficient video information retrieval, video data has to be manipulated properly. Four retrieval techniques are: (1) shot boundary detection, where a video stream is partitioned into various meaningful segments for efficient managing and accessing of video data; (2) key frames selection, where summarization of information in each shot is achieved through selection of a representative frame that depicts the various features contained within a particular shot; (3) low-level feature extraction from key frames, where color, texture, shape, and motion of objects are extracted for the purpose of defining indices for the key frames and then shots; and (4) information retrieval, where a query in the form of input is provided by the user and then, based on this input, a search is carried out through the database to establish symmetry with the information in the database (Farag & Abdel-Wahab, 2003).

Content-based image retrieval, which is essential for efficient video information retrieval, is emerging as an important research area with application to digital libraries and multimedia databases using low-level features like shape, color, texture, and spatial locations. In one project, Manjunath and Ma (1996) focused on the image processing aspects and, in particular, using texture information for browsing and retrieval of large image data. They propose the use of Gabor wavelet features for texture analysis and provides a comprehensive experimental evaluation. Comparisons with other multi-resolution texture features using the Brodatz texture database indicate that the Gabor features provide the best pattern retrieval accuracy. An application for browsing large air photos is also illustrated by Manjunath and Ma.

Focusing has been given to the use of motion analysis to create visual representations of videos that may be useful for efficient browsing and indexing in contrast with traditional frame-oriented representations. Two major approaches for motion-based representations have been presented. The first approach demonstrated that dominant 2D and 3D motion techniques are useful in their own right for computing video mosaics through the computation of dominant scene motion and/or structure. However, this may not be adequate if object-level indexing and manipulation are to be accomplished efficiently. The second approach presented addresses this issue through simultaneous estimation of an adequate number of simple 2D motion models. A unified view of the two approaches naturally follows from the multiple model approach: the dominant motion method becomes a particular case of the multiple motion method if the number of models is fixed to be one and only the robust EM algorithm without the MDL stage employed (Sawhney & Ayer, 1996).

The problem of retrieving images from a large database is also addressed using an image as a query. The method is specifically aimed at databases that store images in JPEG format and works in the compressed domain to create index keys. A key is generated for each image in the database and is matched with the key generated for the query image. The keys are independent of the size of the image. Images that have similar keys are assumed to be similar, but there is no semantic meaning to the similarity (Shneier & Abdel-Mottaleb, 1996). Another paper provides a state-of-the-art account of Visual Information Retrieval (VIR) systems and Content-Based Visual Information Retrieval (CBVIR) systems (Marques & Furht, 2002). It provides directions for future research by discussing major concepts, system design issues, research prototypes, and currently available commercial solutions. Then a video-based face recognition system by support vector machines is presented. Marques and Furht used Stereovision to coarsely segment the face area from its background and then used a multiple-related template matching method to locate and track the face area in the video to generate face samples of that particular person. Face recognition algorithms based on Support Vector Machines of which both “1 vs. many” and “1 vs. 1” strategies are discussed (Zhuang, Ai, & Xu, 2002).

SUMMARY

A general introduction to the subject area of the book has been given in this chapter. An account of state-of-the-art video data management and information retrieval has been presented. Also, focus was given to specific current problems in both of these fields and the research efforts being made to solve them. Some of the research works done in both of these areas have been presented as examples of the research being conducted. Together, these should provide a broad picture of the issues covered in this book.

REFERENCES

Chakraborty, D., Chakraborty, G., & Shiratori, N. (2002). Multicast: Concept, problems, routing protocols, algorithms and QoS extensions. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 225-245). Hershey, PA: Idea Group Publishing.

Chan, S., & Li, Q. (1999). Developing an object-oriented video database system with spatio-temporal reasoning capabilities. *Proceedings of International Conference on Conceptual Modeling (ER’99)*, LNCS 1728: 47-61.

Chan, S., & Li, Q. (2000). Architecture and mechanisms of a web-based data management system. *Proceedings of IEEE International Conference on Multimedia and Expo* (ICME 2000).

Chang, C., Xiao, G., & Chen, T. (2002). A simple prediction method for progressive image transmission. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 262-272). Hershey, PA: Idea Group Publishing.

Chew, C.M., & Kankanhalli, M.S. (2001). Compressed domain summarization of digital video. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia – Advances in Multimedia Information Processing – PCM 2001* (pp. 490-497). October, Beijing, China.

Courtiat, J.P., de Oliveira, R.C., & da Carmo, L.F.R. (1994). Towards a new multimedia synchronization mechanism and its formal specification. *Proceedings of the ACM International Conference on Multimedia* (pp. 133-140). San Francisco, CA.

Ding, J., Huang, Y., Zeng, S., & Chu, C. (2002). Video database techniques and video-on-demand. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications*, (pp. 133-146). Hershey, PA: Idea Group Publishing.

Farag, W.E., & Abdel-Wahab, H. (2004). Video content-based retrieval techniques. In S. Deb (Ed.), *Multimedia systems and content-based image retrieval* (pp. 114-154). Hershey, PA: Idea Group Publishing.

Fung, C., Lau, R., Li, Q., Leong, H.V., & Si, A. (2002). Distributed temporal video DBMS using vertical class partitioning technique. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 90-110). Hershey, PA: Idea Group Publishing.

ISO/IEC 11172-2, Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbit/s, Part 2; Video.

ISO/IEC 13818-2, Generic coding of moving pictures and associated information, Part 2; Video.

Kang, H. (2002). Video abstraction techniques for a digital library. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 120-132). Hershey, PA: Idea Group Publishing.

Karlapalem, K., & Li, Q. (1995). Partitioning schemes for object oriented databases. *Proceedings of International Workshop on Research Issues in Data Engineering – Distributed Object Management* (RIDE-DOM'95) (pp. 42-49).

Karlapalem, K., & Li, Q. (2000). A framework for class partitioning in object-oriented databases. *Journal of Distributed and Parallel Databases*, 8, 317-50.

Li, F. (2002). Video-on-demand: Scalability and QoS control. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 111-119). Hershey, PA: Idea Group Publishing.

Lin, F. (2002). Multimedia and multi-stream synchronization. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 246-261). Hershey, PA: Idea Group Publishing.

Lu, C., Liao, H.M., Chen, J., & Fan, K. (2002). Watermarking on compressed/uncompressed video using communications with side information mechanism. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications*, (pp. 173-189). Hershey, PA: Idea Group Publishing.

Luo, Y., Hwang, J., & Wu, T. (2004). Object-based Video Analysis and Interpretation. In S. Deb (Ed.), *Multimedia systems and content-based image retrieval* (pp. 182-199). Hershey, PA: Idea Group Publishing.

Manjunath, B.S., & Ma, W.Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8).

Marques, O., & Furht, B. (2002). Content-based visual information retrieval. In T.K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 37-57). Hershey, PA: Idea Group Publishing.

Monsignori, M., Nesi, P., & Spinu, M. (2004). Technology of music score watermarking. In S. Deb (Ed.), *Multimedia systems and content-based image retrieval* (pp. 24-61). Hershey, PA: Idea Group Publishing.

Sawhney, H., & Ayer, S. (1996). Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8).

Shih, T. (2002). Distributed multimedia databases. In T. K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 2-12). Hershey, PA: Idea Group Publishing.

Shneier, M., & Abdel-Mottaleb, M. (1996). Exploiting the JPEG compression scheme for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8).

Si, A., Leong, H.V., Lau, R.W.H., & Li, Q. (2000). A temporal framework for developing real time video database systems. *Proceedings of Joint Conference on Information Sciences: Workshop on Intelligent Multimedia Computing and Networking* (pp. 492-495).

Tsai, C., Chang, C., Chen, T., & Chen, M. (2002). Embedding robust gray-level watermark in an image using discrete cosine transformation. In T. K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 206-223). Hershey, PA: Idea Group Publishing.

Yang, M., & Tseng, Y. (2002). Broadcasting approaches for VOD services. In T. K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 147-171). Hershey, PA: Idea Group Publishing.

Zhao, R., & Grosky, W.I. (2001). Bridging the semantic gap in image retrieval. In T. K. Shih (Ed.), *Distributed multimedia databases: Techniques and applications* (pp. 14-36). Hershey, PA: Idea Group Publishing.

Zhou, W., & Dao, S.K. (2001). Combining hierarchical classifiers with video semantic indexing systems. In *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia – Advances in Multimedia Information Processing – PCM 2001* (pp. 78-85). October, Beijing, China.

Zhuang, L., Ai, H., & Xu, G. (2002). Video based face recognition by support vector machines. *Proceedings of 6th Joint Conference on Information Sciences*, March 8-13 (pp. 700-703). Research Triangle Park, NC

Section II

Video Data Storage Techniques and Networking

Chapter II

HYDRA: High-performance Data Recording Architecture for Streaming Media[1]

Roger Zimmermann, University of Southern California, USA

Kun Fu, University of Southern California, USA

Dwipal A. Desai, University of Southern California, USA

ABSTRACT

This chapter describes the design for High-performance Data Recording Architecture (HYDRA). Presently, digital continuous media (CM) are well established as an integral part of many applications. In recent years, a considerable amount of research has focused on the efficient retrieval of such media for many concurrent users. The authors argue that scant attention has been paid to large-scale servers that can record such streams in real time. However, more and more devices produce direct digital output streams, either over wired or wireless networks, and various applications are emerging to make use of them. For example, cameras now provide the means in many industrial applications to monitor, visualize, and diagnose events. Hence, the need arises to capture and store these streams with an efficient data stream recorder that can handle both recording and playback of many streams simultaneously and provide a central repository for all data. With this chapter, the authors present the design of the HYDRA system, which uses a unified architecture that integrates multi-stream recording and retrieval in a coherent paradigm, and hence provides support for these emerging applications.

INTRODUCTION

Presently, digital continuous media (CM) are well established as an integral part of many applications. Two of the main characteristics of such media are that (1) they require real-time storage and retrieval, and (2) they require high bandwidths and space. Over the last decade, a considerable amount of research has focused on the efficient retrieval of such media for many concurrent users. Algorithms to optimize such fundamental issues as data placement, disk scheduling, admission control, transmission smoothing, etc., have been reported in the literature.

Almost without exception, these prior research efforts assumed that the CM streams were readily available as files and could be loaded onto the servers offline without the real-time constraints that the complementary stream retrieval required. This is certainly a reasonable assumption for many applications where the multimedia streams are produced offline (e.g., movies, commercials, educational lectures, etc.). In such an environment, streams may originally be captured onto tape or film. Sometimes the tapes store analog data (e.g., VHS video) and sometimes they store digital data (e.g., DV camcorders). However, the current technological trends are such that more and more sensor devices (e.g., cameras) can directly produce digital data streams. Furthermore, some of these new devices are network-capable, either via wired (SDI, Firewire) or wireless (Bluetooth, IEEE 802.11x) connections. Hence, the need arises to capture and store these streams with an efficient data stream recorder that can handle both recording and playback of many streams simultaneously and provide a central repository for all data.

The applications for such a recorder start at the low end with small, personal systems. For example, the "digital hub" in the living room envisioned by several companies will, in the future, go beyond recording and playing back a single stream as is currently done by TiVo and ReplayTV units (Wallich, 2002). Multiple camcorders, receivers, televisions, and audio amplifiers will all connect to the digital hub to either store or retrieve data streams. At the higher end, movie production will move to digital cameras and storage devices. For example, George Lucas' "Star Wars: Episode II, Attack of the Clones" was shot entirely with high-definition digital cameras (Huffstutter & Healey, 2002). Additionally, there are many sensor networks that produce continuous streams of data. For example, NASA continuously receives data from space probes. Earthquake and weather sensors produce data streams as do Web sites and telephone systems. Table 1 illustrates a sampling of continuous media types with their respective bandwidth requirements.

In this chapter, we outline the design issues that need to be considered for large-scale data stream recorders. Our goal was to produce a unified architecture that integrates multi-stream recording and retrieval in a coherent paradigm by adapting and extending proven algorithms where applicable and introducing new concepts where necessary. We term this architecture HYDRA: High-performance Data Recording Architecture.

Multi-disk continuous media server designs can largely be classified into two different paradigms: (1) Data blocks are striped in a *round-robin* manner across the disks and blocks are retrieved in *cycles* or *rounds* on behalf of all streams; and (2) Data blocks are placed *randomly* across all disks and the data retrieval is based on a *deadline* for each block. The first paradigm attempts to guarantee the retrieval or storage of all data. It is often referred to as *deterministic*. With the second paradigm, by its very nature of

Table 1. A sampling of different media types and their respective data transmission rates

Media Type	**Specifications**	**Data Rate (per second)**
CD-quality audio	2 channels, 16-bit samples at 44,100 kHz	1.4 Mb/s
MPEG-2 encoded video	NTSC-quality (720x480)	4 to 8 Mb/s
MPEG-2 encoded video	HDTV-quality (1920x1080)	19.4 Mb/s
DV	NTSC-quality (720x480)	25 to 36 Mb/s
DVCPRO50	NTSC-quality (720x480)	50 Mb/s
DVCPROHD	HDTV-quality (1920x1080)	100 Mb/s
HDCAM	HDTV-quality (1920x1080)	135 Mb/s

randomly assigning blocks to disks, no absolute guarantees can be made. For example, a disk may briefly be overloaded, resulting in one or more missed deadlines. This approach is often called *statistical*.

One might at first be tempted to declare the deterministic approach intuitively superior. However, the statistical approach has many advantages. For example, the resource utilization achieved can be much higher because the deterministic approach must use worst case values for all parameters, such as seek times, disk transfer rates, and stream data rates, whereas the statistical approach may use average values. Moreover, the statistical approach can be implemented on widely available platforms such as Windows or Linux that do not provide hard real-time guarantees. It also lends itself very naturally to supporting a variety of different media types that require different data rates — both constant (CBR) and variable (VBR) — as well as interactive functions such as pause, fast-forward, and fast-rewind. Finally, it has been shown that the performance of a system based on the statistical method is on par with that of a deterministic system (Santos, Muntz, & Ribeiro-Neto, 2000).

For these reasons, we base our architectural design on a statistical approach. It has been shown that the probability of missed deadlines in such a system follows roughly an exponential curve. Hence, a very low stream hiccup probability can be achieved up to a certain system utilization (say 80%). By the same token it is very important to know how every additional stream will affect the system utilization. Consequently, one of the major design features of our architecture is a comprehensive admission control algorithm that enables an accurate calculation of the stream hiccup probability and system utilization.

The *design goals* of our architecture can be summarized as follows:

- Provide support for the real-time recording of multiple, concurrent streams that are of various media types. For example, streams may be received at different average bit rates and be encoded with constant (CBR) or variable bit rate (VBR) techniques.
- Provide support for the synchronized recording of multiple streams.
- Be a modular, scalable architecture.
- Use unified algorithms (e.g., data placement and scheduling) that can accommodate both recording and playback simultaneously in any combination with low latency.

The organization of this chapter is as follows. The next section, Related Work, relates our work to prior research and commercial systems. The section, Architecture Design, presents our proposed architecture and describes many of the relevant issues. Furthermore, we present some preliminary algorithms for admission control, data placement, and disk scheduling. The Conclusion contains remarks about our future plans.

RELATED WORK

This chapter details the design of a unified systems architecture. Therefore, it relates to a considerable number of research topics. Several of the issues that we were faced with have been addressed by themselves in academic research. Rather than list them here, we will point to the prior academic research in the sections where the relevant issues are discussed.

The Multicast Multimedia Conference Recorder (MMCR) (Lambrinos, Kirstein, & Hardman, 1998) probably is the most related to our architecture. The purpose of this project was to capture and play back multicast (MBone) sessions. The authors list a number of interesting and relevant issues for such systems. They focus more on the higher level aspects such as indexing and browsing the available sessions, while assuming only a small number of concurrent sessions. Our design, on the other hand, is specifically concerned with a scalable, high performance architecture where resources (memory, disk space, and bandwidth) need to be carefully scheduled.

There are also commercial systems available that relate to our design. We classify them into the following three categories:

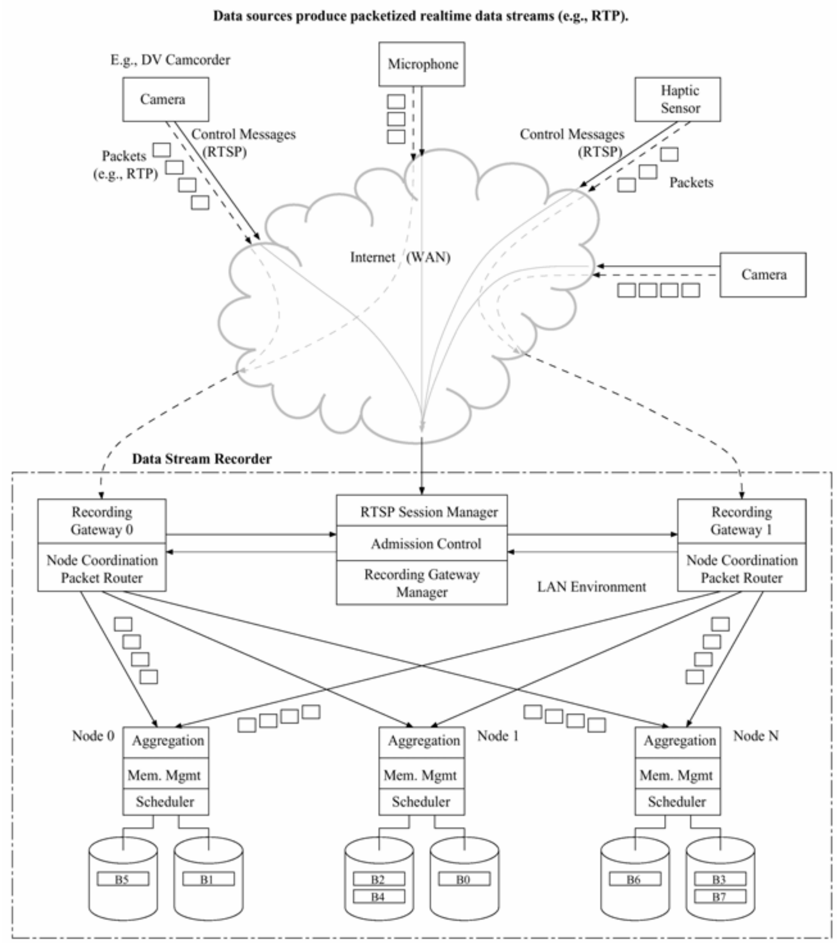
1. **Streaming media systems** (e.g., Microsoft's Windows Media, Apple's QuickTime, and RealNetwork's RealOne). These systems are optimized for streaming of previously (offline) stored content. Some of them also allow real- time live streaming (i.e., forwarding with no recording). They are designed for multi-user access and multiple media types. They cannot usually take advantage of a cluster of server nodes.
2. **Personal video recorders (PVR).** For example, the TiVo and ReplayTV models and the SnapStream software. These systems allow real-time recording and playback of standard broadcast quality video. Some of their limitations are that they are designed as single-user systems. Furthermore, they are optimized for a single media type (NTSC/PAL/SECAM video with two channels of audio). Local playback is supported, and with newer models file sharing is enabled over a network. However, they do not provide streaming playback over a network.
3. **Editing systems and broadcast servers.** These systems are the professional cousins of the PVRs. They are used for the production and distribution of video content (e.g., to TV stations), and they are designed to interface via professional I/O standards (usually not Ethernet). Their use is for local environments, not distributed streaming setups. Most of the time they handle only a few media types and one (or a few) streams at a time. Their special purpose hardware and elaborate control interfaces to other studio equipment places them into a price category that makes them not cost-effective for use as a more general purpose stream recorder.

As indicated, none of these categories encompasses the full functionality that we envision. Each one of them only provides a subset of the desired functionalities.

ARCHITECTURE DESIGN

Figure 1 illustrates the architecture of a scalable data stream recorder operating in an IP network environment. Multiple, geographically distributed sources, for example, video cameras, microphones, and other sensors, acquire data in real time, digitize it, and send it to the stream recorder. We assume that the source devices include a network

Figure 1. Data Stream Recorder Architecture (Multiple source and rendering devices are interconnected via an IP infrastructure. The recorder functions as a data repository that receives and plays back many streams concurrently. Note, playback streams are not shown to simplify the diagram.)



interface and that the data streams are transmitted in discrete packets. A suitable protocol for audio and video data traffic would be the Real-time Transport Protocol (RTP) (Schulzrinne, Casner, Frederick, & Jacobson, 1996) on top of the Universal Datagram Protocol (UDP). The client-recorder dialog that includes control commands such as record, pause, resume, and stop is commonly handled via the Real-time Streaming Protocol (RTSP) (Schulzrinne, Rao, & Lanphier, 1998).

The data stream recorder includes two interfaces to interact with data sources: (1) a *session manager* to handle RTSP communications, and (2) multiple *recording gateways* to receive RTP data streams. A data source connects to the recorder by initiating an RTSP session with the session manager, which performs the following functions: (1) admission control for new streams, (2) maintaining RTSP sessions with sources, and (3) managing the recording gateways. As part of the session establishment, the data source receives detailed information about which recording gateway will handle its data stream. Media packets are then sent directly to this designated gateway, bypassing the manager. Multiple recording gateways are supported by each stream recorder, providing scalability to a large number of concurrent streams and removing the bottleneck caused by having a single entry point for all packets. Specifically, each recording gateway performs the following functions: (1) handling of errors during transmissions, (2) timestamping of packets (see section on Packet Timestamping), (3) packet-to-storage-node assignment and routing, and (4) storage node coordination and communication. A recording gateway forwards incoming data packets to multiple *storage node*s. Each storage node manages one or more local disk storage devices. The functions performed in the storage nodes are (1) packet-to-block (P2B) aggregation, (2) memory buffer management, (3) block data placement on each storage device, (4) real-time disk head scheduling, and (5) retrieval scheduling for outgoing streams.

Some of these functions are also present in a playback-only system. However, the recording of streams requires new approaches or modifications of existing algorithms. Here is a summary of features of this architecture.

- Multi-node, multi-disk cluster architecture to provide scalability.
- Multiple recording gateways to avoid bottlenecks due to single-point recording.
- Random data placement for the following operations: block-to-storage node assignment, block placement within the surface of a single disk, and optionally packet-to-block assignment. These result in the harnessing of the average transfer rate for multi-zone disk drives and improve scalability.
- Unified model for disk scheduling: deadline-driven data reading and writing (fixed block sizes reduce complexity of file system).
- Unified memory management with a shared buffer pool for both reading and writing.
- Statistical admission control to accommodate variable bit rate (VBR) streams and multi-zone disk drives.

We will now discuss each function in turn. The discussion of the admission control algorithm is deferred to the Admission Control Section because it is an overarching component that relies on many of the other concepts that will be introduced first.

Session Management

The Real-Time Streaming Protocol (RTSP) provides a well-defined set of commands for managing recording sessions. Figure 2 shows a sample RTSP request-response exchange for establishing a recording with one audio and one video stream. Once an RTSP session is successfully set up, the session manager informs the recording gateways of session details such as port numbers, expected bandwidth, etc.

Figure 2. Sample RTSP request-response exchange to establish a recording session (S – source, R – recorder)

```
S->R:
ANNOUNCE rtsp://imsc.usc.edu/openhouse RTSP/1.0
CSeq: 90
Content-Type: application/sdp
Content-Length: 121
s=IMSC Open House Demo
u=http://imsc.usc.edu
t=3080271600 3080703600

R->S:
RTSP/1.0 200 OK
CSeq: 90

S->R:
SETUP rtsp://imsc.usc.edu/openhouse/audiotrack
RTSP/1.0
CSeq: 91
Transport:
RTP/AVP;multicast;destination=10.1.1.1;port=21010-
21011;mode=record;ttl=127

R->S:
RTSP/1.0 200 OK
CSeq: 91
Session: 50887676
Transport:
RTP/AVP;multicast;destination=10.0.1.1;port=21010-
21011;mode=record;ttl=127

S->R:
SETUP rtsp://imsc.usc.edu/openhouse/videotrack
RTSP/1.0
CSeq: 92
Session: 50887676
Transport:
RTP/AVP;multicast;destination=10.1.1.2;port=61010-
61011;mode=record;ttl=127

R->S:
RTSP/1.0 200 OK
CSeq: 92
Transport:
RTP/AVP;multicast;destination=10.1.1.2;port=61010-
61011;mode=record;ttl=127

S->R:
RECORD rtsp://server.example.com/meeting RTSP/1.0
CSeq: 93
Session: 50887676
Range: clock=19961110T1925-19961110T2015

R->S:
RTSP/1.0 200 OK
CSeq: 93
```

Recording Gateway Management

The recording gateways are the media stream entry points into the recorder. Each gateway maintains its own available network bandwidth. Different streams are assigned to different gateways based on the current workload and the resources available. The session manager informs a gateway whenever a new stream is assigned to it (gateways ignore packets that do not have a recognized session ID). If a session is paused, resumed, or stopped, the gateway is also notified by the session manager.

As part of the stream admission control, the session manager is aware of the resource utilization of every gateway. A newly entering stream must announce how much bandwidth it expects to use, and the session manager will assign it to the most appropriate gateway. In turn, the gateway will allocate the necessary resources to the incoming stream so that there is no loss of data because of resource over-utilization.

Transmission Error Recovery

The recorder architecture accepts data in the form of RTP packets, which are usually based on UDP datagrams. UDP is a best-effort delivery mechanism and does not provide any guarantees to ensure packet delivery. Since we may be recording content from an original source, lost packets are not acceptable as they will be permanently missing from the stored data. There are a number of methods to minimize losses during packet transmission, such as Forward Error Control (FEC) and Retransmission Based Error Control (RBEC) (Zimmermann, Fu, Nahata, & Shahabi, 2003). HYDRA uses a form of a selective retransmission protocol that is optimized for recording.

Packet Timestamping

With continuous data streams, packets need to be timestamped such that the temporal relationship between different parts of a stream can be preserved and later reproduced (intra-stream synchronization). Such timestamps also help to establish the synchronization between multiple streams (inter-stream synchronization).

Packets may be timestamped directly at the source. In that case, intra-stream synchronization will not be affected by any network jitter that a stream experiences during its network transmission to the recorder. However, inter-stream synchronization with other data originating from geographically different locations requires precise clock synchronization of all locations. One possible solution is to use clock information from Global Positioning System (GPS) receivers if very precise timing is needed (in the order of microseconds). For synchronization in the order of tens of milliseconds, a solution such as the network time protocol (NTP) may suffice.

If packets are timestamped once they reach the data recorder, then the temporal relationship is established between packets that arrive concurrently. Hence, if Stream A has a longer transmission time than Stream B, the time difference will be implicitly recorded, and if the transmission delays are not identical during playback, then any combined rendering of A+B will be out-of-sync. Furthermore, with this approach any packet jitter that was introduced by the network will be permanently recorded as part of the stream. For these reasons, it is preferable to timestamp packets directly at the source.

Packet-to-Block Aggregation

We discuss packet-to-block aggregation before the packet-to-storage-node assignment in the section, Block-to-Storage-Node Assignment, even though the two steps happen in reversed order in the actual system. However, we believe that from a conceptual point of view the discussion will be easier to understand.

Packet-switched networks such as the Internet generally use relatively small quanta of data per packet (for example, 1400 bytes). On the other hand, magnetic disk drives operate very inefficiently when data is read or written in small amounts. This is due to the fact that disk drives are mechanical devices that require a transceiver head to be positioned in the correct location over a spinning platter before any data can be transferred. Figure 3a shows the relative overhead experienced with a current generation disk drive (Seagate Cheetah X15) as a function of the retrieval block size. The disk parameters used are shown in Table 2. The overhead was calculated based on the seek

Figure 3. Disk characteristics of a high performance disk drive [Seagate Cheetah X15, see Table 2) (The transfer rate varies in different zones. Because of the very high transfer rates, a considerably large block size is required to achieve a reasonable bandwidth utilization (i.e., low overhead).]

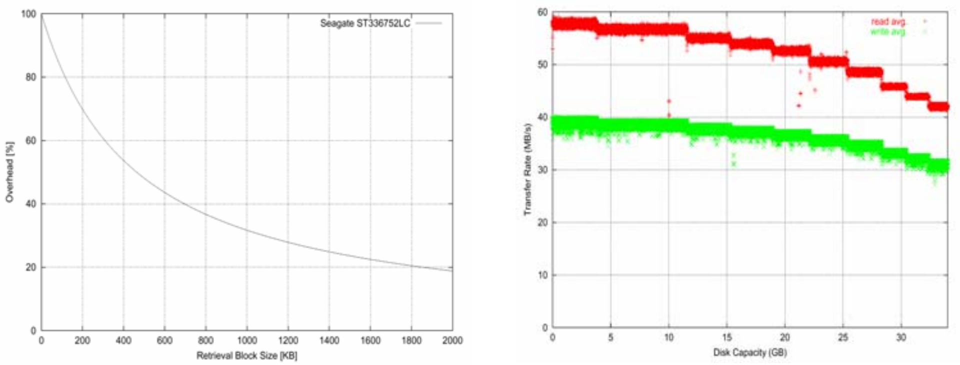


Figure 3a. Overhead in terms of seek time and rotational latency as a percentage of the total retrieval time, which includes the block transfer time.

Figure 3b. Maximum read and write rate in different areas (also called zones) of the disk. The write bandwidth is up to 30% less than the read bandwidth.

Table 2. Parameters for a current high performance commercial disk drive

Model **Series** **Manufacturer**	**ST336752LC** **Cheetah X15** **Seagate Technology, LLC**
Capacity C	37 GB
Transfer rate RD	See Figure 3
Spindle speed	15,000 rpm
Avg. rotational latency	2 msec
Worst case seek time	≈7 msec
Number of Zones Z	9

time needed to traverse half of the disk's surface plus the average rotational latency. As illustrated, only large block sizes beyond one or two megabytes allow a significant fraction of the maximum bandwidth to be used (the fraction is also called *effective bandwidt*h). Consequently, incoming packets need to be aggregated into larger data blocks for efficient storage and retrieval. There are two ways this can be accomplished.

- **Sequential:** With this scheme, incoming packets are aggregated in sequence into blocks. For example, if *m* packets fit into one block then the receiver routing algorithm will send *m* sequential packets to one node before selecting another node as the target for the next *m* packets. As a result, each block contains sequentially numbered packets. The advantage of this technique is that only one buffer at a time per stream needs to be available in memory across all the storage nodes.
- **Random:** With this scheme, each incoming packet is randomly assigned to one of the storage nodes, where they are further collected into blocks. One advantage of this technique is that during playback data is sent randomly from all storage nodes at the granularity of a packet. Therefore, load-balancing is achieved at a small data granularity. The disadvantage is that one buffer per node needs to be allocated in memory per stream. Furthermore, the latency until the first data block can be written to a storage device is about N times longer than in the sequential case, where N is the number of storage nodes.

Generally, the advantage of needing only $\frac{1}{N}$ times the memory for the sequential case outweighs the load-balancing advantages of random. When many streams are retrieved simultaneously, load balancing with the sequential approach (i.e., at the granularity of a block) should be sufficient. We plan to quantify the exact trade-offs between these two techniques as part of our future work.

Block-to-Storage-Node Assignment

To present a single point of contact for each streaming source, packets are collected at a recording gateway as indicated earlier. However, to ensure load balancing, these packets need to be distributed across all the storage nodes. Storing individual packets is very inefficient, and hence they need to be aggregated into larger data blocks as described in the Packet-to-Block Aggregation Section.

Once the data is collected into blocks, there are two basic techniques to assign the data blocks to the magnetic disk drives that form the storage system: in a *round-robin* sequence (Berson, Ghandeharizadeh, Muntz, & Ju, 1994), or in a *random* manner (Santos & Muntz, 1998). Traditionally, the round-robin placement utilizes a cycle-based approach to scheduling of resources to guarantee a continuous display, while the random placement utilizes a deadline-driven approach. There has been extensive research investigating both techniques in the context of continuous media stream retrievals. The basic characteristics of these techniques still apply with a mixed workload of reading and writing streams.

In general, the round-robin/cycle-based approach provides high throughput with little wasted bandwidth for video objects that are stored and retrieved sequentially (e.g., a feature-length movie). Block retrievals can be scheduled in advance by employing optimized disk scheduling algorithms (such as *elevator* [Seltzer, Chen, & Ousterhout, 1990]) during each cycle. Furthermore, the load imposed by a display is distributed evenly across all disks. However, the initial startup latency for an object might be large under heavy load because the disk on which the starting block of the object resides might be busy for several cycles. Additionally, supporting variable bit rate streams and interactive operations such as pause and resume are complex to implement. The random/deadline-driven approach, on the other hand, naturally supports interactive functions and VBR streams. Furthermore, the startup latency is generally shorter, making it more suitable for a real-time stream recorder.

- **Block Size:** The block size to use in a continuous media server is usually determined in one of two ways: (a) the block size represents a constant data length (CDL) or (b) the block size represents a constant time length (CTL). With CTL, the size in bytes varies if the media stream is encoded with a variable bit rate. Conversely, with CDL, the amount of playback time per block is variable. A system that utilizes a cycle-based scheduling technique works well with CTL, whereas a deadline-driven system can use either approach. For an actual implementation, the fixed block size of CDL makes the design of the file system and buffer manager much easier. Hence, a CDL design with random placement and deadline-driven scheduling provides an efficient and flexible platform for recording and retrieving streams.

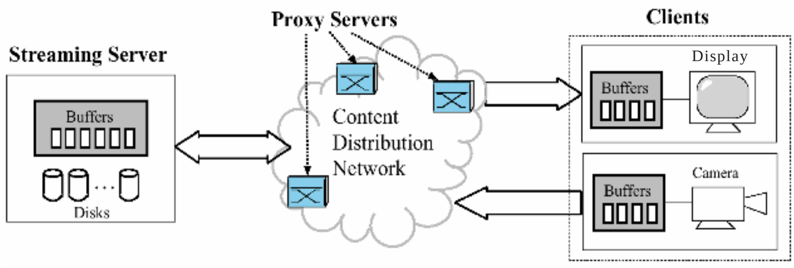
Memory Buffer Management

Managing the available memory efficiently is a crucial aspect of any multimedia streaming system. A number of studies have investigated buffer/cache management. These techniques can be classified into three groups: *server buffer management* (Lee, Whang, Moon, & Song, 2001; Makaroff & Ng, 1995; Shi & Ghandeharizadeh, 1997; Tsai & Lee, 1998; Tsai & Lee, 1999), *network/proxy cache management* (Chae et al., 2002; Cui & Nahrstedt, 2003; Ramesh, Rhee, & Guo, 2001; Sen, Rexford, & Towsley, 1999) and *client buffer management* (Shahabi & Alshayeji, 2000; Waldvogel, Deng, & Janakiarman, 2003). Figure 4 illustrates where memory resources are located in a distributed environment.

When designing an efficient memory buffer management module for a large-scale data stream recorder, we may classify the problems of interest into two categories: (1) *resource partitioning,* and (2) *performance optimizatio*n. In the resource partitioning category, a representative class of problems is — *What is the minimum memory or buffer size that is needed to satisfy certain streaming and recording service requirement*s? The requirements usually depend on the quality of service expectations of the end user or application environment. In the performance optimization category, a representative class of problems is — *Given certain amount of memory or buffer, how to maximize our system performance in terms of certain performance metric*s? Some typical performance metrics are as follows:

1. Maximize the total number of supportable streams.
2. Maximize the disk I/O parallelism, i.e., minimize the total number of parallel disk I/Os.

Figure 4. Buffer distribution in a large-scale streaming system



Approach: The assembly of incoming packets into data blocks and conversely the partitioning of blocks into outgoing packets requires main memory buffers. In a traditional retrieval-only server, double buffering is often used: one buffer is filled with a data block that is retrieved from a disk drive, while the content of the second buffer is emptied (i.e., streamed out) over the network. Once the buffers are full/empty their roles are reversed. In a retrieval-only system, more than two buffers per stream are not necessary. However, if additional buffers are available, they can be used to keep data in memory longer, such that two or more streams of the same content, started at just a slight temporal offset, may share the data (Shi & Ghandeharizadeh, 1997). As a result, only one disk stream is consumed and more displays can be supported.

With a stream recorder, double buffering is still the minimum that is required. However, with additional buffers available, incoming data can be held in memory longer and the deadline by which a data block must be written to disk can be extended. This can reduce disk contention and hence the probability of missed deadlines. Aref, Kamel, Niranjan, and Ghandeharizadeh (1997) introduced an analytical model to calculate the write deadline of a block as a function of the size of the available buffer pool. However, their model does not use a shared buffer pool between readers and writers. In a large- scale stream recorder, the number of streams to be retrieved versus the number to be recorded may vary significantly over time. Furthermore, the write performance of a disk is usually significantly less than its read bandwidth (Figure 3b). Hence, these factors need to be considered and the existing model modified (see also Admission Control Section).

Data Placement on the Disk Platters

The placement of data blocks on a magnetic disk has become an issue for real- time applications since disk manufacturers have introduced multi-zoned disk drives. A disk drive with multiple *zones* partitions its space into a number of regions such that each have a different number of data sectors per cylinder. The purpose of this partitioning is to increase the storage space and allocate more data to the outer regions of a disk platter as compared with the inner regions. Because disk platters spin at a constant angular velocity (e.g., 10,000 revolutions per minute), this results in a data transfer rate that is higher in the outer zones than it is in the inner ones.

Consequently, the time to retrieve or store a data block varies and real-time applications must handle this phenomenon. A conservative solution is to assume the slowest transfer rate for all regions. As a result, the scheduler need not be aware of the location where a block is to be stored or retrieved. However, this approach might waste a significant fraction of the possible throughput. The transfer rate ratio between the innermost and outermost zones sometimes exceeds a factor of 1.5.

A number of techniques have been proposed to improve the situation and harness more of a disk's potential. IBM's Logical Tracks (Heltzer, Menon, & Mitoma, 1993), Hewlett Packard's Track Pairing (Birk, 1995), and USC's FIXB (Ghandeharizadeh, Kim, Shahabi, & Zimmermann, 1996) all attempt to utilize the average transfer rate instead of the minimum. All these approaches were designed to work with deterministic scheduling techniques with the assumption that every block access must not exceed a given time span.

However, in the context of random assignments of blocks to disks and stochastic, deadline-driven scheduling, this assumption can be relaxed. By randomly placing blocks into the different zones of a disk drive, the average transfer rate can easily be achieved. However, now the block retrieval times vary significantly. By observing that the retrieval time is a random variable with a mean value and a standard deviation we can incorporate it into the admission control module such that an overall statistical service guarantee can still be achieved. An advantage of the random block-to-zone data placement is its simplicity and the elegance of using the same random algorithm both within a disk and across multiple disks.

Real-Time Disk Head Scheduling

Recall that the effective bandwidth of a magnetic disk drive depends to a large degree on the overhead (the seek time and rotational latency) that is spent on each block retrieval. The effect of the overhead can be reduced by increasing the block size. However, this will result in a higher memory requirement. Conversely, we can lower the overhead by reducing the seek time. Disk-scheduling algorithms traditionally achieve this by ordering block retrievals according to their physical locations and hence minimize the seek distance between blocks. However, in real-time systems such optimizations are limited by the requirement that data needs to be retrieved in a timely manner.

In a multimedia server that utilizes random data placement, deadline-driven scheduling is a well-suited approach. Furthermore, for a system that supports the recording and retrieval of variable bit rate streams, deadline-driven scheduling is an efficient way with medium complexity. Cycle-based scheduling becomes very complex if different media types are to be supported that require fractional amounts of blocks per round to be accessed.

For example, the SCAN-EDF (Reddy & Wyllie, 1993) algorithm combines earliest-deadline-first disk head scheduling with a scan optimization. Data blocks are retrieved in order of their deadlines and if some deadlines are identical then the scan order is used. The deadline of a block can be obtained from the timestamp of the last packet that it contains plus an offset. For data retrieval, the offset is commonly the time when the retrieval request was admitted into the system. For stream writing, the offset needs to be determined from the delay that a block is allowed to stay in a main memory buffer until it must be flushed to a physical disk drive. The SCANRT-RW (Aref et al., 1997) algorithm treats both reads and writes in a uniform way and computes the writing deadlines based on the amount of buffer memory available. However, it assumes a partitioned buffer pool where some space is allocated exclusively for writing. How a unified buffer pool affects the scheduling performance should be investigated.

Admission Control

The task of the admission-control algorithm is to ensure that no more streams are admitted than the system can handle with a predefined quality. A number of studies have investigated admission-control techniques in multimedia server designs. Figure 6 classifies these techniques into two categories: *measurement-based* and *parameter-based*. The parameter-based approach can be further divided into *deterministic* and *statistical* algorithms.

With measurement-based algorithms (Bao & Sethi, 1999; Kim, Kim, Lee, & Chung, 2001), the utilization of critical system resources is measured continually and the results are used in the admission-control module. Measurement-based algorithms can only work online and cannot be used offline to configure a system or estimate its capacity. Furthermore, it is difficult to obtain an accurate estimation of dynamically changing system resources. For example, the time window during which the load is measured influences the result. A long time window smooths out load fluctuations but may overlap with several streams being started and stopped, while a short measurement interval may over or underestimate the current load.

With deterministic admission control (Chang & Zakhor, 1996; Lee & Yeom, 1999; Makaroff et al., 1997; Narasimha & Wyllie, 1994), the worst case must be assumed for the following parameters: stream bandwidth requirements, disk transfer rate, and seek overhead. Because compressed streams, such as MPEG-2, may require quite variable bit

Figure 5. Parameter sets used for the admission control algorithm

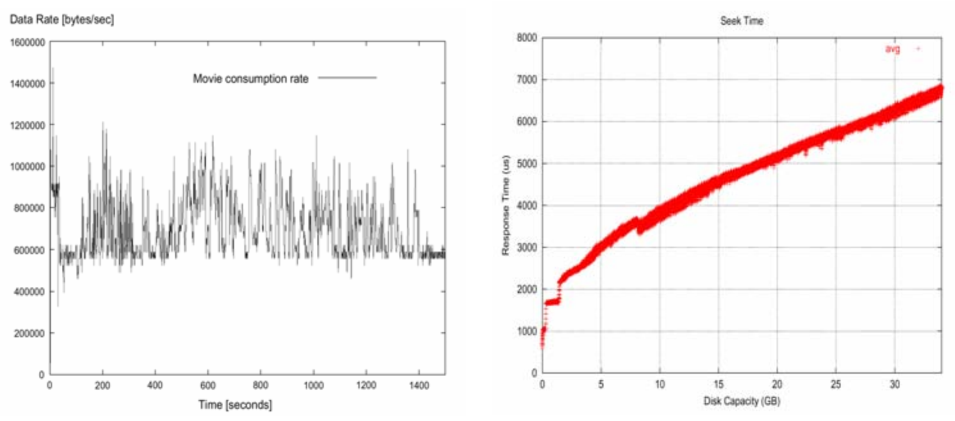
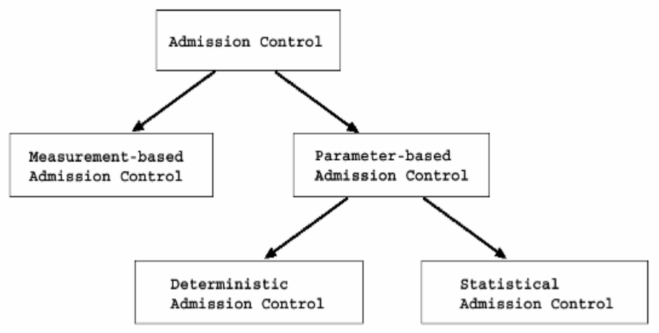


Figure 5a. The consumption rate of a movie encoded with a VBR MPEG-2 algorithm.

Figure 5b. The seek profile of a Seagate Cheetah X15 disk drive (see also Table 2).

Figure 6. Taxonomy of different admission control algorithms


Admission Control
Measurement-based Admission Control
Parameter-based Admission Control
Deterministic Admission Control
Statistical Admission Control

rates (Figure 5a) and the disk transfer rates of todays multi-zoned disk drives also vary by a factor of up to 1.5-to-1, assuming worst case parameters will result in a significant underutilization of resources in the average case. Furthermore, if an operating system without real-time capabilities is used service guarantees may be violated even with conservative calculations. Consequently, for a more practical approach we focus on statistical admission control where service is guaranteed with a certain probability to not exceed a threshold requested by the user.

Statistical admission control has been studied in a number of papers (Chang & Zakhor, 1994; Kang & Yeom, 2000; Nerjes, Muth, & Weikum, 1997; Vin, Goyal, & Goyal, 1994). Vin et al. (1994) exploit the variation in disk access times to media blocks as well as the VBR client load to provide statistical service guarantees for each client. Note that in Vin et al., the distribution function for disk service time is obtained through exhaustive empirical measurements. Chang and Zakhor (1994) introduce three ways to estimate the disk overload probability while Kang and Yeom (2000) propose a probabilistic model that includes caching effects in the admission control. Nerjes et al. (1997) introduce a stochastic model that considers VBR streams and the variable transfer rates of multi-zone disks.

Recently, the effects of user interaction on admission control have been studied (Friedrich, Hollfelder, & Aberer, 2000; Kim & Das, 2000). Kim and Das (2000) proposed an optimization for the disk and cache utilization while reserving disk bandwidth for streams that are evicted from cache. Friedrich et al. (2000) introduced a Continuous Time Markov Chains (CTMCs) model to predict the varying resource demands within an interactive session and incorporated it into the admission control algorithm.

Next, we describe a novel statistical admission control algorithm called Three Random variable Admission Control (TRAC) that models a much more comprehensive set of features of real-time storage and retrieval than previous work.

1. Support for variable bit rate (VBR) streams (Figure 3a illustrates the variability of a sample MPEG-2 movie).
2. Support for concurrent reading and writing of streams. The distinguishing issue for a mixed workload is that disk drives generally provide less write than read bandwidth (Figure 3b). Therefore, the combined available bandwidth is a function

of the read/write mix. We propose a dynamic bandwidth-sharing mechanism as part of the admission control.

3. Support for multi-zoned disks. Figure 3b illustrates that the disk transfer rates of current generation drives is platter location-dependent. The outermost zone provides up to 30% more bandwidth than the innermost one.
4. Modeling of the variable seek time and variable rotational latency that is naturally part of every data block read and write operation.
5. Support for efficient random data placement (Muntz, Santos, & Berson, 1997b).

Most of the previously proposed statistical admission-control algorithms have adopted a very simple disk model. Only Nerjes et al. (1997) consider the variable transfer rate of multi-zone disks. Theirs differs from our TRAC algorithm in that (1) it assumes that all zones have the same number of tracks, (2) it did not consider the variance of the seek time, and (3) it is based on round-robin data placement and round-based disk scheduling. Additionally, no previous study has considered the difference in the disk transfer rate for reading and writing (Figure 3b).

TRAC Algorithm: Consider a server recording n variable bit rate streams using deadline-driven scheduling and movie blocks that are allocated to disks using a random placement policy. The server activity is observed over time intervals with duration T_{svr}. Our model is characterized by three random variables: (1) $D(i)$ denotes the amount of data to be retrieved or recorded for client i during observation window T_{svr}, (2) $\overline{R_{Dr}}$ denotes the average disk read bandwidth during T_{svr} with no bandwidth allocation to writing, and (3) $\overline{T_{seek}}$ denotes the average disk seek time during each observation time interval T_{svr}.

Let $T_{seek}(i)$ denote the disk seek time for client i during T_{svr}[2]. Let n_{rs} and n_{ws} denote the number of retrieval and recording streams served respectively, i.e., $n = n_{rs} + n_{ws}$. Also, $\hat{R}_{Dw}$ represents the average disk bandwidth (in MB/s) allocated for writing during T_{svr}, while $\hat{R}_{Dr}$ represents the average bandwidth for reading. With such a mixed load of both retrieving and recording clients, the average combined disk bandwidth $\overline{R_{Dio}}$ is constrained by $\overline{R_{Dio}} = \hat{R}_{Dw} + \hat{R}_{Dr}$. Consequently, the maximum amount of data that can be read and written during each interval T_{svr} can be expressed by:

$$\overline{R_{Dio}} \times \left(T_{svr} - \sum_{i=1}^{n_{ns}+n_{ws}} T_{seek}(i) \right).$$

Furthermore, if

$$\sum_{i=1}^{n} D(i)$$

represents the total read and write bandwidth requirement during T_{svr} from all streams n, then the probability of missed deadlines, p_{iodisk}, can be computed by Equation 1.

$$p_{iodisk} = P\left[\sum_{i=1}^{n} D(i) > \left(\overline{R_{Dio}} \times \left(T_{svr} - \sum_{i=1}^{n} T_{seek}(i)\right)\right)\right] \quad \textbf{(1)}$$

Note that a missed deadline of a disk access does not necessarily cause a hiccup for the affected stream because data buffering may hide the delay. However, we consider the worst case scenario for our computations.

Recall that

$$\sum_{i=1}^{n} T_{seek}(i)$$

denotes the total seek time spent for all n clients during T_{svr}. Let $t_{seek}(j)$ denote the seek time for disk access j, where j is an index for each disk access during T_{svr}. Thus, the total seek time can be computed as follows:

$$\sum_{i=1}^{n} T_{seek}(i) = \sum_{j=1}^{m} t_{seek}(j) = m \times \overline{t_{seek}} \quad \textbf{(2)}$$

where m denotes the number of seeks and $\overline{t_{seek}}$ is the average seek time, both during T_{svr}. Because every seek operation is followed by a data block read or write, m can also be expressed by

$$m = \frac{\sum_{i=1}^{n} D(i)}{B_{disk}},$$

where B_{disk} is the block size. With the appropriate substitutions we arrive at our final expression for the probability of overcommitting the disk bandwidth, which may translate into missed I/O deadlines.

$$p_{iodisk} = P\left[\sum_{i=1}^{n} D(i) > \left(\frac{\overline{R_{Dio}} \times T_{svr}}{1 + \frac{\overline{t_{seek}} \times \overline{R_{Dio}}}{B_{disk}}}\right)\right] \le p_{req} \quad \textbf{(3)}$$

Table 3. Parameters used in the experiments and analysis

Parameters	**Configurations**
Test movie "Twister"	MPEG-2 video, AC-3 audio
Average bandwidth	698594 Bytes/sec
Length	50 minutes
Throughput std. dev.	140456.8
Test movie "Saving Private Ryan"	MPEG-2 video, AC-3 audio
Average bandwidth	757258 Bytes/sec
Length	50 minutes
Throughput std. dev.	169743.6
Test movie "Charlie's Angels"	MPEG-1 video, Stereo audio
Average bandwidth	189129 Bytes/sec
Length	70 minutes
Throughput std. dev.	56044.1
Disk Model	Seagate Cheetah X15 (Model ST336752LC)
Mixed-load factor α	1.0 (retrieval only experiments) 0.0 (recording only experiments) 0.4094 (retrieval and recording mixed experiments)
Relationship factor β (between R_{Dr} and R_{Dw})	0.6934
Mean inter-arrival time of streaming request	5 seconds
Server observation window T_{svr}	1 second
Disk block size B_{disk}	1.0 MB
Number of disks (ξ)	1, 2, 4, 8, 16, … , 1024

Because of space constraints, we do not include all the equations that produce the final calculation of the stream disruption probability (for detailed discussions, see Zimmermann & Fu, 2003). Rather, we include some of the experimental results that we obtained with a Seagate Cheetah X15 disk drive (see Table 3 for parameter values).

Figure 7(a) shows the measurement and theoretical retrieving experimental results for the DVD movie "Twister." The y-axis shows the probability for missed deadlines of all block requests. When the number of streams $n \leq 55$, then the probability is very small (< 1%). Above this threshold, the probability increases sharply and reaches 1 for $n = 62$. The analytical results based on our 3RV model follow the measurements very closely, except that the 1% transition is one stream higher at 55. The miss probability of the 1RV model is also shown and its transition point is 39. Consequently, not only does our 3RV model result in a 38% improvement over the simpler model (for $p_{req} = 1\%$), but it also tracks the physical disk performance much more accurately.

The retrieving experimental results for the VCD movie "Charlie's Angels" are shown in Figure 7(c). Because of the lower bandwidth requirement of this video, a much higher

Figure 7. Retrieval or recording only experimental results

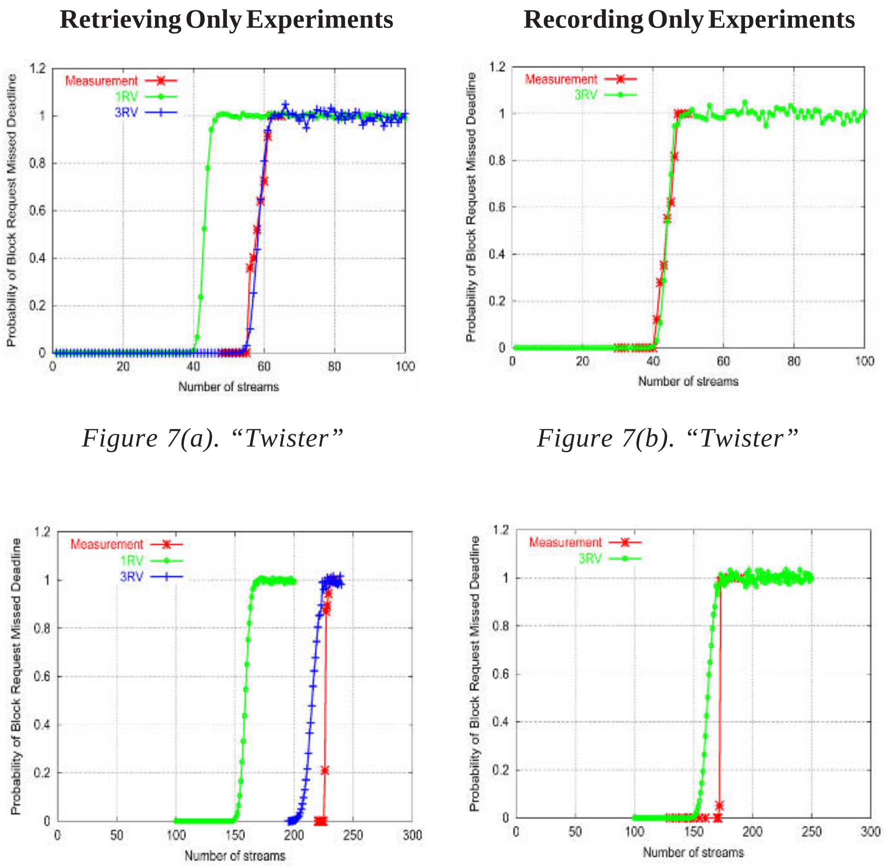


Figure 7(a). "Twister"

Figure 7(b). "Twister"

Figure 7(c). "Charlie's Angels"

Figure 7(d). "Charlie's Angels"

number of streams (150 resp. 200) can be supported. The improvement of 3RV over 1RV is similar to the "Twister" case, and we have omitted the graphs for "Saving Private Ryan" because the results were comparable.

Figures 7(b) and (d) show the miss probabilities for our recording experiments. Analogous to the retrieval case, the 3RV curve very closely matches the measured values. Since the disk write bandwidth is significantly lower than the read bandwidth (see Fig.3(b)), the transition point for, say "Twister," is $n = 40$ instead of $n = 55$ in the stream retrieval experiment.

Figure 8. Mixed workload recording and retrieval experiments

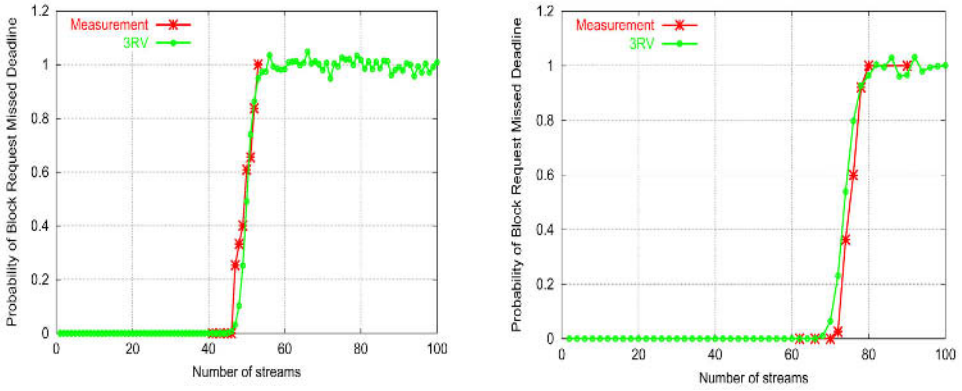


Figure 8(a). "Twister" (retrieval and recording).

Figure 8(b). A mix of "Saving Private Ryan" and "Charlie's Angels" (retrieval and recording).

Figure 8 shows the results of mixed workload experiments. Figure 8(a) shows an example graph with workload generated by a single media type (the DVD movie "Twister"; note that the other media types produced analogous results). As expected, the 1% transition point at 46 streams lies between the pure retrieval (54) and recording (40) values. Figure 8(b) shows the experimental results with workload generated by two different media types (the DVD movie "Saving Private Ryan" and the VCD movie "Charlie's Angels") and, once again, the miss probability computed by 3RV model closely matches the measured results.

Retrieval Scheduling for Outgoing Streams

Any stream recorder will naturally have to support the retrieval of streams as well. Hence, it needs to include all the functionality that traditional continuous media servers contain. The framework that we presented so far was deliberately designed to be very complementary to the functionality required for stream retrieval. The combination of random data placement and deadline driven scheduling has been shown previously to be an efficient platform for serving media streams (Berson, Muntz, & Wong, 1996; Muntz, Santos, & Berson, 1997a; Shahabi, Zimmermann, Fu, & Yao, 2002).

Other Functions

For a fully functional data stream recorder, a number of additional functions would need to be provided. Such functions include: authentication of users and devices, editing facilities to create new content, browsing facilities to quickly find the content of interest, and remote control of recorder functions. These functions are also necessary in a playback-only system and therefore are not discussed here in detail.

CONCLUSIONS

The need for high performance and scalable data stream recorders will arise for many applications. We have described an architecture that is flexible, scalable, and incorporates practical issues. We outlined an admission-control algorithm that considers the multi-zoning, the variable seek and rotational latency overhead, and the varying read and write performance of the current generation of disk drives. Additionally, incoming streams with variable bit rate requirements are supported.

We have also outlined additional issues such as the main memory buffer management with a unified pool of buffers, synchronization of multiple streams, and the write deadline determination of incoming data. Some of these ideas pose new research challenges and will require further investigation.

REFERENCES

Aref, W., Kamel, I., Niranjan, T. N., & Ghandeharizadeh, S. (1997). Disk scheduling for displaying and recording video in non-linear news editing systems. *Proceedings of the Multimedia Computing and Networking Conferenc*e, (pp. 228-239), San Jose, CA. SPIE Proceedings Series, Volume 3020.

Bao, Y., & Sethi, A.S. (1999). Performance-driven adaptive admission control for multimedia applications. In *IEEE International Conference on Communications, 1999 (ICC '99*), (vol. 1, pp. 199-203).

Berson, S., Ghandeharizadeh, S., Muntz, R., & Ju, X. (1994). Staggered striping in multimedia information systems. *Proceedings of the ACM SIGMOD International Conference on Management of Dat*a.

Berson, S., Muntz, R. R., & Wong, W. R. (1996). Randomized data allocation for real-time disk I/O. *Proceedings of the 41st IEEE International Computer Conference,* (pp. 286-290). Washington, DC: IEEE Computer Society.

Birk, Y. (1995). Track-pairing: A novel data layout for VOD servers with multi-zone recording disks. *Proceedings of the International Conference on Multimedia Computing and System*s, (pp. 248-255).

Chae, Y., Guo, K., Buddhikot, M. M., Suri, S., & Zegura, E. W. (2002). Silo, rainbow, and caching token: Schemes for scalable, fault tolerant stream caching. *Special Issue of IEEE Journal of Selected Area in Communications on Internet Proxy Service*s.

Chang, E., & Zakhor, A. (1994). Variable bit rate MPEG video storage on parallel disk arrays. In *First International Workshop on Community Networkin*g. San Francisco, CA.

Chang, E., & Zakhor, A. (1996). Cost analyses for VBR video servers. *IEEE Multi Medi*a, *3*(4), 56-71.

Cui, Y., & Nahrstedt, K. (2003). Proxy-based asynchronous multicast for efficient on-demand media distribution. In *The SPIE Conference on Multimedia Computing and Networking 2003 (MMCN 2003),* Santa Clara, CA, pp. 162-176.

Friedrich, M., Hollfelder, S., &Aberer, K. (2000). Stochastic resource prediction and admission for interactive sessions on multimedia servers. *Proceedings of ACM Multimedi*a, Los Angeles, CA (pp. 117-126).

Ghandeharizadeh, S., Kim, S. H., Shahabi, C., & Zimmermann, R. (1996). Placement of continuous media in multi-zone disks. In S.M. Chung (Ed.), *Multimedia Information Storage and Managemen*t, Chapter 2. Boston, MA: Kluwer Academic Publishers. ISBN: 0-7923-9764-9.

Heltzer, S. R., Menon, J. M., & Mitoma, M. F. (1993). *Logical data tracks extending among a plurality of zones of physical tracks of one or more disk devices.* U.S. Patent No.5,202,799.

Huffstutter, P. J., & Healey, J. (2002). Filming without the film. *Los Angeles Time*s, A.1.

Kang, S., & Yeom, H. Y. (2000). Statistical admission control for soft real-time VOD servers. In *ACM Symposium on Applied Computing (SAC 2000*).

Kim, I.-H., Kim, J.-W., Lee, S.-W., & Chung, K.-D. (2001). Measurement-based adaptive statistical admission control scheme for video-on-demand servers. In *The 15th International Conference on Information Networking (ICOIN'01),* Beppu City, Oita, Japan, pp. 472-478.

Kim, S.-E. & Das, C. (2000). A reliable statistical admission control strategy for interactive video-on-demand servers with interval caching. *Proceedings of the 2000 International Conference on Parallel Processing,* Toronto, Canada, August 21-24.

Lambrinos, L., Kirstein, P., & Hardman, V. (1998). The multicast multimedia conference recorder. *Proceedings of the International Conference on Computer Communications and Networks, IC3*N, Lafayette, LA.

Lee, K., &Yeom, H. Y. (1999). An effective admission control mechanism for variable-bit-rate video streams. *Multimedia System*s, *7*(4), 305-311.

Lee, S.-H., Whang, K.-Y., Moon, Y.-S., & Song, I.-Y. (2001). Dynamic buffer allocation in video-on-demand systems. *Proceedings of the International Conference on Management of Data (ACM SIGMOD 2001*), Santa Barbara, CA (pp. 343-354).

Makaroff, D. J., Neufeld, G. W., & Hutchinson, N. C. (1997). An evaluation of VBR disk admission algorithms for continuous media file servers. *ACM Multimedi*a, 143-154.

Makaroff, D. J., & Ng, R. T. (1995). Schemes for implementing buffer sharing in continuous-media systems. *Information Systems*, *20*(6), 445-464.

Muntz, R., Santos, J., & Berson, S. (1997a). RIO: A Real-time Multimedia Object Server. *ACM Sigmetrics Performance Evaluation Revie*w, *25*.

Muntz, R., Santos, J. R., & Berson, S. (1997b, Sept). Rio: A real-time multimedia object server. *ACM Sigmetrics Performance Evaluation Review*, *25*(2), 29-35.

Narasimha, A. R., & Wyllie, J. C. (1994). I/O issues in a multimedia system. *IEEE Compute*r, *27*(3), 69-74.

Nerjes, G., Muth, P., & Weikum, G. (1997). Stochastic service guarantees for continuous data on multi-zone disks. *Proc. of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1997*), Tucson, AZ (pp. 154-160).

Ramesh, S., Rhee, I., & Guo, K. (2001). Multicast with cache (mcache): An adaptive zero delay video-on-demand service. *IEEE INFOCOM '01*, 85-94.

Reddy, A. L. N., & Wyllie, J. C. (1993). Disk scheduling in a multimedia I/O system. *Proceedings of the ACM Multimedia Conferenc*e, Anaheim, CA, (pp. 225-233).

Santos, J. R., & Muntz, R. R. (1998). Performance analysis of the RIO multimedia storage system with heterogeneous disk configurations. In *ACM Multimedia Conferenc*e, Bristol, UK.

Santos, J. R., Muntz, R. R., & Ribeiro-Neto, B. (2000). Comparing random data allocation and data striping in multimedia servers. *Proceedings of the SIGMETRICS Conferenc*e, Santa Clara, CA.

Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (1996). RTP: A transport protocol for real time applications. Retrieved at: *http://ww.itef.org/rfc/rfc1889.txt*

Schulzrinne, H., Rao, A., & Lanphier, R. (1998). Real time streaming protocol (RTSP). Retrieved at: *http://ww.itef.org/rfc/rfc2326.txt*

Seltzer, M., Chen, P., & Ousterhout, J. (1990). Disk scheduling revisited. *Proceedings of the 1990 Winter USENIX Conferenc*e, Washington DC, (pp. 313-324). Usenix Association.

Sen, S., Rexford, J., & Towsley, D. F. (1999). Proxy prefix caching for multimedia streams. *IEEE INFOCOM '99*, 1310-1319.

Shahabi, C., & Alshayeji, M. (2000). Super-streaming: A new object delivery paradigm for continuous media servers. *Journal of Multimedia Tools and Application*s, *11*(1).

Shahabi, C., Zimmermann, R., Fu, K., & Yao, S.-Y. D. (2002). Yima: A second generation continuous media server. *IEEE Compute*r, *35*(6), 56-64.

Shi, W., & Ghandeharizadeh, S. (1997). Buffer sharing in video-on-demand servers. *SIGMETRICS Performance Evaluation Revie*w, *25*(2), 13-20.

Tsai, W.-J., & Lee, S.-Y. (1998). Dynamic buffer management for near video-on-demand systems. *Multimedia Tools and Applications, 6*(1), 61-83.

Tsai, W.-J., & Lee, S.-Y. (1999). Buffer-sharing techniques in service-guaranteed video servers. *Multimedia Tools and Applications*, *9*(2), 121-145.

Vin, H. M., Goyal, P., & Goyal, A. (1994). A statistical admission control algorithm for multimedia servers. *Proceedings of ACM International Conference on Multimedia* (pp. 33-40).

Waldvogel, M., Deng, W., & Janakiraman, R. (2003). Efficient buffer management for scalable media-on-demand. In *The SPIE Conference on Multimedia Computing and Networking 2003 (MMCN 2003)*, Santa Clara, CA.

Wallich, P. (2002). Digital hubbub. *IEEE Spectru*m, *39*(7), 26-29.

Zimmermann, R., & Fu, K. (2003). Comprehensive statistical admission control for streaming media servers. *Proceedings of the 11th ACM International Multimedia Conference (ACM Multimedia 2003*), Berkeley, CA (pp. 75-85).

Zimmermann, R., Fu, K., Nahata, N., & Shahabi, C. (2003). Retransmission-based error control in a many-to-many client-server environment. *Proceedings of the SPIE Conference on Multimedia Computing and Networking 2003 (MMCN 2003*), Santa Clara, CA.

ENDNOTES

[1] This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and IIS-0082826, and an unrestricted cash gift from the Lord Foundation.

[2] $T_{seek}(i)$ includes rotational latency as well.

Chapter III

Wearable and Ubiquitous Video Data Management for Computational Augmentation of Human Memory

Tatsuyuki Kawamura, Nara Institute of Science and Technology, Japan

Takahiro Ueoka, Nara Institute of Science and Technology, Japan

Yasuyuki Kono, Nara Institute of Science and Technology, Japan

Masatsugu Kidode, Nara Institute of Science and Technology, Japan

ABSTRACT

This chapter introduces video data management techniques for computational augmentation of human memory, i.e., augmented memory, on wearable and ubiquitous computers used in our everyday life. The ultimate goal of augmented memory is to enable users to conduct themselves using human memories and multimedia data seamlessly anywhere, anytime. In particular, a user's viewpoint video is one of the most important triggers for recalling past events that have been experienced. We believe designing augmented memory system is a practical issue for real world-oriented video data management. This chapter also describes a framework for an augmented memory albuming system named Sceneful Augmented Remembrance Album (SARA). In the SARA *framework, we have developed three modules for retrieving, editing, transporting, and exchanging augmented memory. Both the Residual Memory module and the* I'm

Here! module enable a wearer to retrieve video data that he/she wants to recall in the real world. The Ubiquitous Memories module is proposed for editing, transporting, and exchanging video data via real world objects. Lastly, we discuss future works for the proposed framework and modules.

INTRODUCTION

In this chapter, we introduce our wearable and ubiquitous video data management study for computational augmentation of human memory in everyday life. Scientific psychological analysis and engineering technologies for memory aid have been studied extensively in recent years. Psychological results show that the human brain can cause mistakes in either the encoding, storing, or retrieval process (Brewer & Treyens, 1981; Nickerson & Adams, 1979). The technology of the computational augmentation of human memory aims to integrate computationally recorded multimedia data named, "augmented memory" (Rhodes, 1995, 1997), into human memory. The ultimate goal of augmented memory is to enable users to conduct themselves using these memories seamlessly anywhere, anytime in their everyday life. In particular, video data provides the user with strong stimuli to recall past events that he or she has experienced. Our research consists of several studies used in developing a video-based augmented memory system.

In the field of computational memory aid, several representative works on wearable computers exist. Mann (1997), for example, described a user who wears a CCD camera and sensors to record his/her everyday life. This system allows the user to get information anytime and anywhere the user wants. This type of human-centered computer technology is called "wearable computing." Wearable computing technology must be aware of the user's internal (desire, emotion, health, action, etc.) and external (goings on, temperature, other people, etc.) state at any time. *Jimminy* (a Wearable Remembrance Agent) also supports human activities using just-in-time information retrieval (Rhodes, 2003). Kawashima, Nagasaki, and Toda (2002) and Toda, Nagasaki, Iijima, and Kawashima (2003) have developed an automatic video summarization system using visual pattern recognition methods for location recognition and a view tracking device for a user's action recognition. The *Mithril* platform has also advanced over the years (DeVaul, Pentland , & Corey, 2003). Kidode (2002) has developed an advanced media technology project named a *Wearable Information Playing Station* (WIPS). Our study of augmented memory in this chapter is a part of the WIPS project.

Lamming and Flynn (1994) have developed *Forget-me-not*, a prototype system as a portable episodic memory aid. This system records a user's action history using sensors implanted in a laboratory and active badges worn by users. The user can refer to his/her own history and easily replay a past event on a PDA. This *Forget-me-not* study is based on the concept of "ubiquitous computing" proposed by Weiser (1991). *The Aware Home Research Initiative* is also directly inspired by the same concept (Abowd & Mynatt, 2002; Kidd et al., 1999; Tran & Mynatt, 2003).

Augmented memory technologies using video data are divided into three simple topics as follows: a location-based memory aid, ubiquitous video data management, and a person's information-based augmented memory. In the case of location-based memory supporting systems, Hoisko (2000) developed a visual episodic memory prosthesis that retrieves video data recorded at the place attached to certain IR tags. The Global

Positioning System (GPS) and the Geographic Information System (GIS) are also used for a location-based memory-aid. Ueda, Amagasa, Yoshikawa, and Uemura. (2002) proposed an automatic video summarization method using a wearer position detected by the GPS, with the user's body direction showing his/her interests around various architectures. In the object-based augmented memory, Rekimoto, Ayatsuka, and Hayashi (1998) proposed a system for browsing information that includes the physical context of a user's current situation using *CyberCode* (Rekimoto & Ayatsuka, 2000) tags. Their aim was to develop interactive technologies in physical environments. A proposed video replay system, *DyPERS*, stores a user's visual and auditory scenes (Jebara, Schiele, Oliver, & Pentland, 1998; Schiele, Oliver, Jebara, & Pentland, 1999). This system can retrieve a video clip using a signal that was explicitly registered by a user who pushes a button while he/she looks at an interesting scene. The *VAM* system by Farringdon and Oni (2000) detects a human face recorded previously and displays information about the retrieved person. Kato, Kurata, and Sakaue (2002a; 2002b) proposed *VizWear-Active*, which includes a robust face-registration method and a stable face-tracking method. Kawamura, Kono, and Kidode (2003), however, proposed a managing system named the *Nice2CU* for recollection of a person's information such as profiles, messages, experiences, and human relations in the real world.

Other augmented memory systems are described in this section. We consider each system capable of being one of the modules for an ideal augmented memory system. Segmentation of information and summarization research has been studied because such research provides information necessary for wearers to be able to refer to their recorded viewpoint video data. Aizawa, Ishijima, and Shina (2001) proposed a system that summarizes a video scene from a head-worn camera by segmenting brain waves. Picard and Healey (1997) developed *StartleCam*, which records video data triggered by skin conductivity from a startled response from a user. In the *Affective Wearables*, various types of sensors were investigated to analyze and detect expressions of emotion in a certain environment (Healey & Picard, 1998). In addition, the following video segmentation methods have been proposed: a statistical trajectory segmentation method (Gelgon, & Tilhou, 2002), and a wearer's attention to a moving/stable object (Nakamura, Ohde, & Ohta, 2000). Augmented memory has been investigated from the point of view of important psychological perspectives by DeVaul, Pentland, and Corey (2003) and Czerwinski and Horvitz (2002). Context awareness (Clarkson & Pentland, 2000, 2001) and the modeling of event structure, as in *Ubi-UCAM* (Jang & Woo, 2003), are also important issues. Ueoka, Hirota, and Hirose (2001) studied what kinds of sensors are necessary. Murakami (2002) developed an editing system named the *Memory Organizer* that enables users to refer to memory with a multi-directional viewpoint of interests using weak information structures, as in *CoMeMo* (Maeda, Hirata, & Nishida, 1997). The *ComicDiary* (Sumi, Sakamoto, Nakao, & Mase, 2002) automatically produces a comic style diary of the users' individual experiences.

The rest of this chapter consists mainly of the following four sections. The first section introduces a framework design named *SARA* for wearable computational augmentation of human memory. The second section explains one of the memory retrieval modules of *SARA*, the *Residual Memory*. In addition, the third section discusses the memory retrieval module termed, *I'm Here!* The fourth section concludes the *Ubiquitous Memories* module, which is the base module for the memory retrieval, memory exchange, and memory transportation of *SARA*.

A FRAMEWORK FOR A VIDEO-BASED AUGMENTED MEMORY ALBUMING SYSTEM

In this section we introduce a framework for a computational memory aid useful in everyday life. The aim of this research is to develop a video-based albuming system for augmenting human memory. We propose a framework for an augmented memory albuming system named Sceneful Augmented Remembrance Album (SARA), used to realize computational memory aid in everyday life (Kawamura, Kono, & Kidode, 2002; Kono, Kawamura, Ueoka, Murata, & Kidode, 2003) in which:

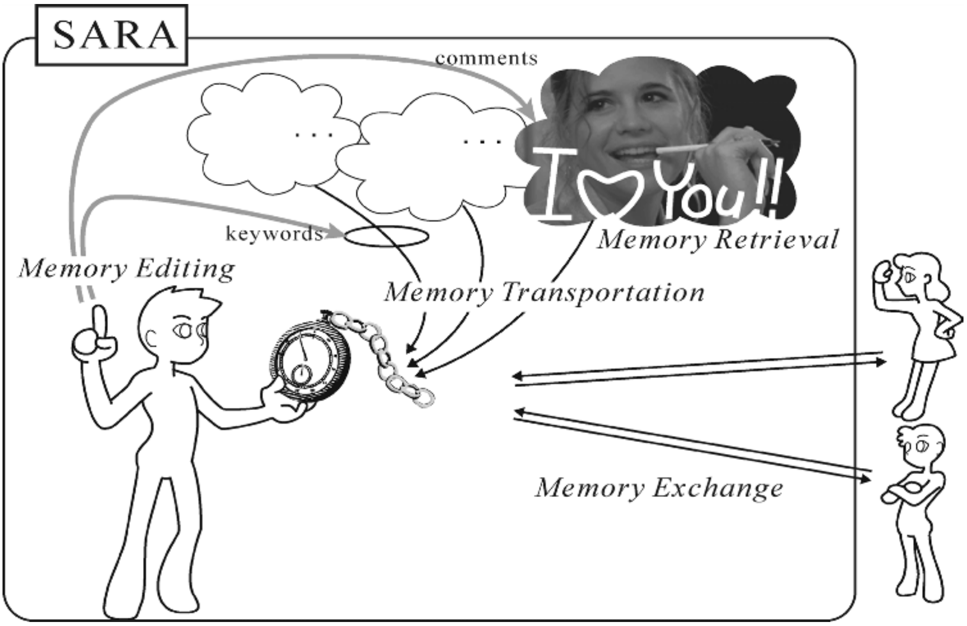
- A user's viewpoint images are always observed.
- The observed images along with the data observed by other wearable sensors are analyzed to detect contexts.
- The observed images are stored with the detected contexts as the user's augmented memories.
- The stored data can be additionally annotated/indexed by the user for later retrieval.
- The user can recall his/her experiences by reviewing a stored video, which is retrieved by consulting an index, via a Head-Mounted Display (HMD), that is, both annotating and indexing data automatically stored and manually annotated information.

SARA: A Sceneful Augmented Remembrance Album

Figure 1 illustrates the overall architecture of the proposed augmented memory albuming system. We believe that the following four functions are essential to realize *SARA*:

- **Memory Retrieval.** A memory retrieval function provides a user with the ability to retrieve appropriate multimedia data from a huge multimedia database termed "video memory." A human chooses his/her best remembrance strategy from various remembrance strategies and applies this strategy to the context. We have developed the following two modules for Memory Retrieval: "Residual Memory" for supporting a location-based user's recollection, and "I'm Here!" for an object-based one. "Ubiquitous Memories," mentioned in Memory Transportation, also works as an object-based memory retrieval module.
- **Memory Transportation.** A memory transportation function provides a user with the ability to associate either the event he/she is enacting or the augmented memory retrieved by one of the memory retrieval modules, with other features. By using such a function, the user is able to rearrange his/her memories for later retrieval. Using *Ubiquitous Memories*, the user can associate augmented memories with real world objects. He/she can also hold and convey the memories to the associated objects.

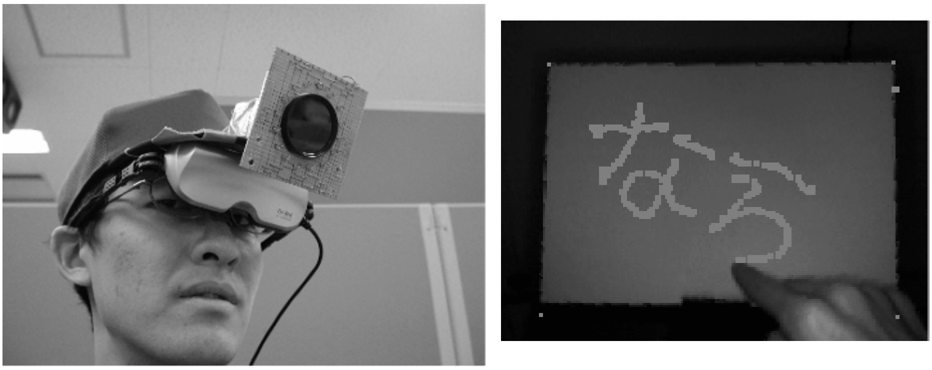
Figure 1. Basic functions for operating augmented memories on SARA



- **Memory Exchange.** A user can augment his/her problem-solving ability by referring to others' experience if that experience is properly associated with the given problem. *Ubiquitous Memories* helps its users exchange their experiences. By accessing a real-world object, a user can view all the augmented memories associated with the object if the owner of each memory has approved other users viewing the object.
- **Memory Editing.** A person remembers an event he/she has experienced with conceptual/linguistic indexes. A memory-editing function provides a user with the ability to make annotations to his/her memories by adding keywords or free-hand comments, or by reordering his/her memories based on his/her own intentions. Our research group has developed a vision interface, which is named a *Wearable Virtual Tablet* (Ukita, Terabe, Kono, & Kidode, 2002), using only the user's fingertips (Figure 2). In Figure 2 (right), the user wrote down "Nara" in Japanese.

Utilizing these functions, the user can reuse human experience by remembering his/her own augmented memories, or by viewing other users' augmented memories rearranged in a real world object. In this chapter, we introduce Residual Memory in the section on Location-Based Memory Retrieval. We also introduce I'm Here! in the section on Object-Base Memory Retrieval. Lastly, we introduce Ubiquitous Memories in the section on How to Manage Video Memory in the Real World.

Figure 2. A wearable virtual tablet for annotating and indexing



LOCATION-BASED VIDEO MEMORY RETRIEVAL

We have developed a location-based video retrieval module named *Residual Memory* (Kawamura, Kono, & Kidode, 2001; Kawamura, Ukita, Kono, & Kidode, 2002). *Residual Memory* module assists a user in remembering an event that happened at a particular location. The *Residual Memory* provides associable video retrieval in a previous data set triggered by current video data that is captured from the wearable camera. The module needs high speed and an appropriate location-based video retrieval method for a user's on-demand request using his/her viewpoint image. Generally, the users head frequently moves, and moving objects also move in his/her view. The module must divide captured information into moving information and location information. In order to achieve these methods, the module has to track the user's head movements, detect moving objects in a scene, and remove these two motions from the video in the user's activities. The *Residual Memory* module operates the following three methods (Figure 3).

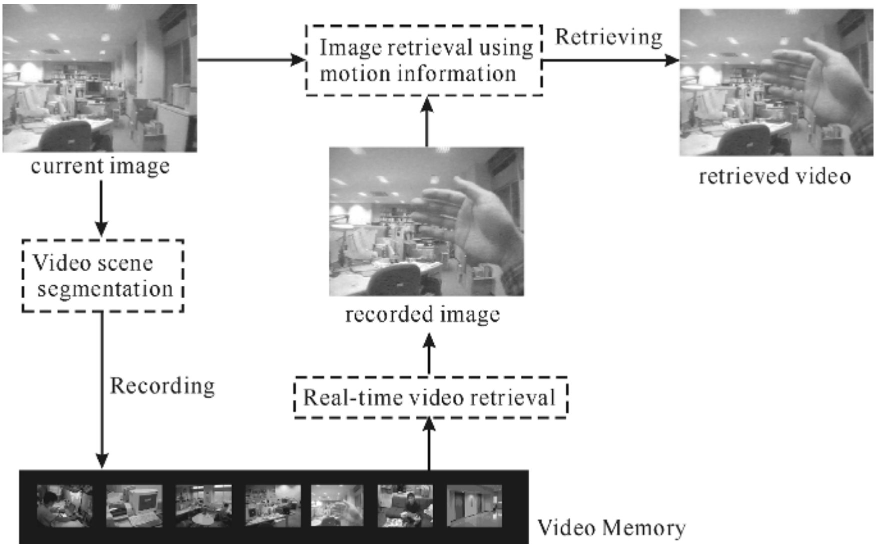
Motion information exclusion from video data is performed using the wearable camera by:

- tracking the yaw and pitch head movements with two mono-axis gyro sensors,
- tracking moving objects in a scene using a block matching method, and by
- excluding motion information by masking moving areas in the scene.

Video scene segmentation from continuous input video data is changed by:

- detecting scene changes continuously from current video data and two gyro data, and by
- indexing each scene for easy viewing.

Figure 3. Overview of residual memory process flow



Real-time video retrieval from large sequential video data is retrieved by:

- dividing small segments from video data for stable and high-speed video retrieval, and by
- retrieving an associable scene from a segment similar to the current video data.

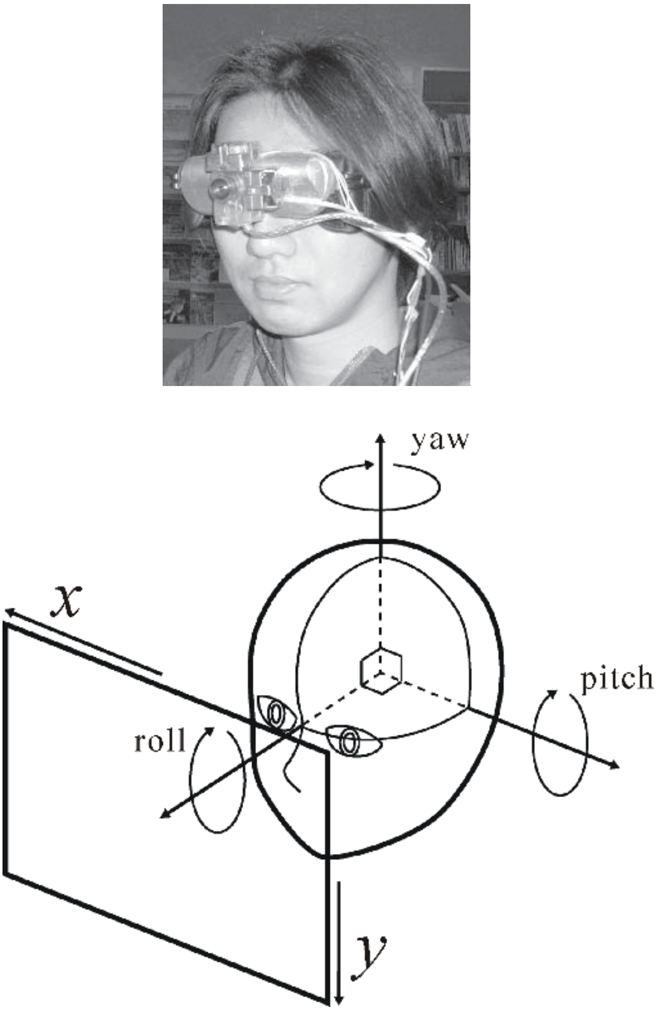
In our video retrieval module, query images from the user's viewpoint are updated with high frequency (the same frequency as the video input rate). By using a high-frequency input, the module can quickly track the changes of the user's request to refer a location-based associable video data. The user can always slightly change an image with his/her own body control and input a query image to the module at the same time. The module, however, must have the function of an on-the-fly adaptive video search. The module prepares itself using a background processing module for a user's unforeseen choices of a scene in the huge video data set.

Image Matching Except for Motion Information

Tracking Head Movements

A user's head motion prevents him/her from recognizing a moving object in a captured video from the wearable camera. We have employed two mono-axis gyro sensors and placed these sensors in the center of the wearable camera as shown in Figure 4 (top). The two gyros can detect yaw-axis motion and pitch-axis motion (Figure 4

Figure 4. A wearable camera, display, and two mono-axis gyro sensors

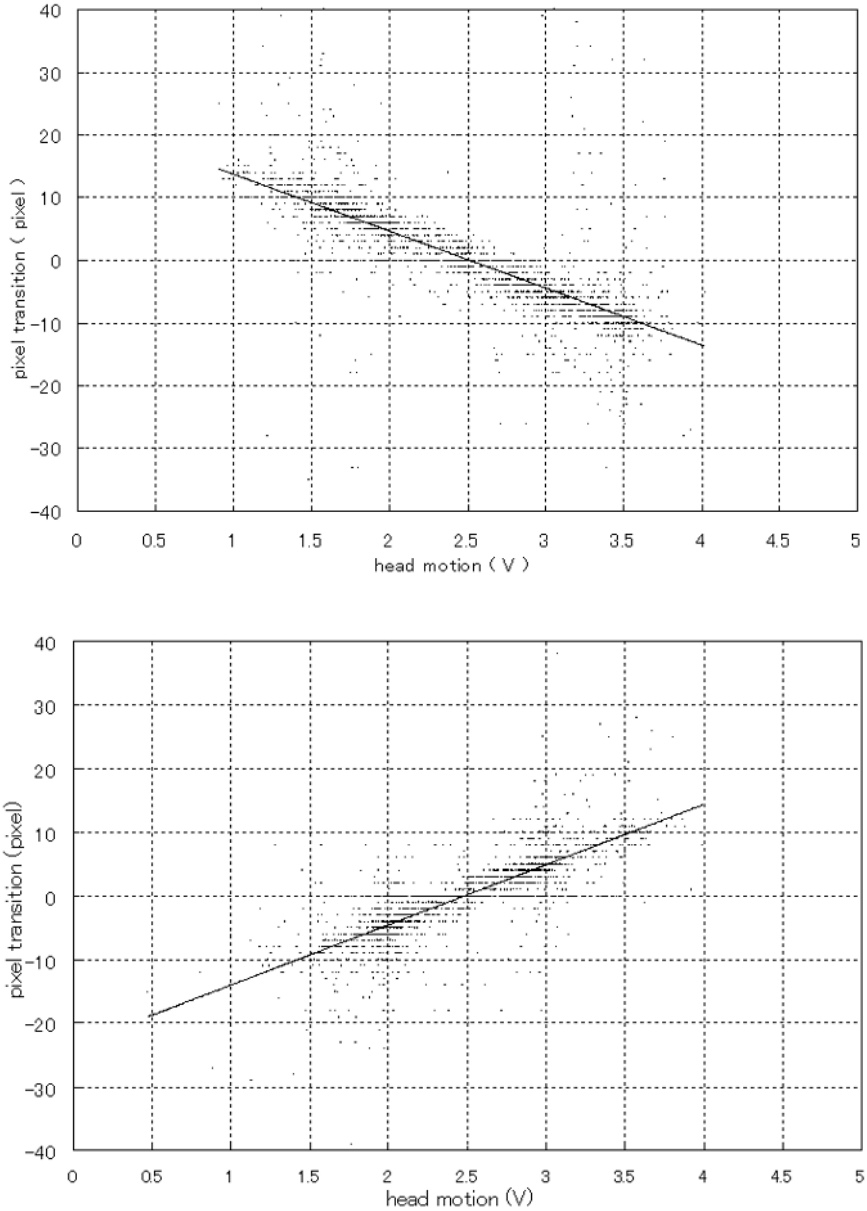


(bottom)). Figure 5 shows the amount relations of value transition, which occurred by head motion between the gyro sensor and an image shift. These results allow us to remove head-motion information from the video data.

Tracking Moving Objects

We have employed a conventional block-matching method. The matching method can detect motion areas, motion directions, and amounts of motion at the same time. This method divides an image into small blocks. Each block is an averaged and normalized RGB-value with ($I = r + g + b$, $I_r = r/I$, $I_g = g/I$, $I_b = b/I$). Our method is defined as the following formulae:

Figure 5. Relations between the Gyro Sensor Value and the Image Shift Value (yaw and pitch)



$$Mr_{i,j}(u,v) = \sum_{u,v}(I_r(i,j) - I_r(i+u, j+v))^2 ,$$

$$Mb_{i,j}(u,v) = \sum_{u,v}(I_b(i,j) - I_b(i+u, j+v))^2 , \tag{1}$$

$$Mb_{i,j}(u,v) = \sum_{u,v}(I_b(i,j) - I_b(i+u, j+v))^2 .$$

$$M_{i,j}(u_{\min}, v_{\min}) = \min_{u,v}\{Mr_{i,j}(u,v) + Mg_{i,j}(u,v) + Mb_{i,j}(u,v)\} \tag{2}$$

($u_{\min}$, $v_{\min}$) represents an estimated minimum motion vector value. The estimated motion block is then redefined into five simple states (up, down, left, right, and not-moving). If a motion vector is adequately small, this block is named, "not-moving."

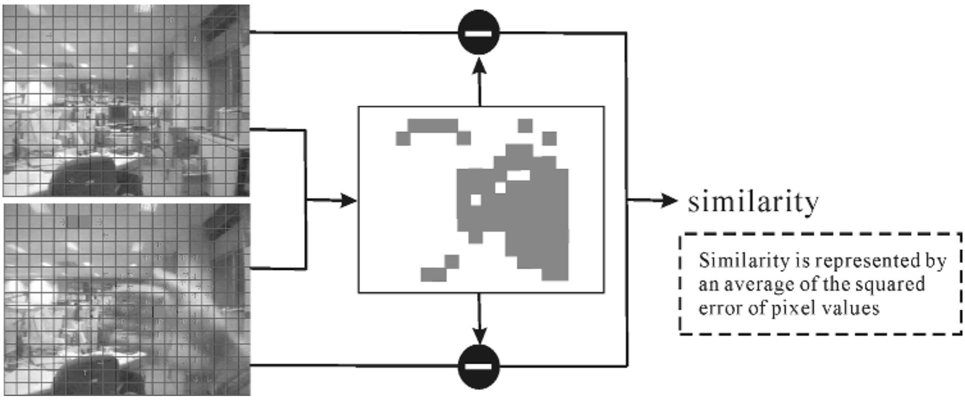
Excluding Motion Information

In order to calculate the similarity of the background as location information, the process has to compare a current query image to a previous image except for motion blocks (Figure 6). The current query image and previous image have eigen-motion information. In order to remove mutual motion blocks in each image from target searching blocks, a motion block mask should be made. The first step of the similarity estimation compares the same address block in two images with blocks in the "not-moving" state. The second step divides a value, from which values are summed and calculated by the previous process, by the number of "not-moving" blocks.

Video Scene Segmentation Using Motion Information

In order to segment a scene from continuously recorded video data, we employed a moving average method using two mono-axis gyro sensors. In the moving average method, continuously input gyro sensor values are added from past values in the T

Figure 6. Similarity of background as location information


similarity
Similarity is represented by an average of the squared error of pixel values

frames prior to the current values and the added value is divided by T. This method obtains a meta trend of captured data. The moving average method equation is as follows:

$$MA_T(t) = \frac{\sum_{i=t-T}^{t} f(i)}{T}. \tag{3}$$

In this study, four values are calculated by the moving average method: two values are calculated with the yaw-axis value, and the other two values are calculated with the pitch-axis gyro value. The following three states are defined to detect scene changes:

- **Stable state:** Both the moving average value of the short interval (e.g., T=30) and that of the long one (e.g., T=300) are within under a certain range.
- **Side1 state:** The moving average of the short interval is higher than that of the long one.
- **Side2 state:** The moving average of the long interval is higher than that of the short one.

If the difference and the amount of the value transition of gyro sensors are large, we choose a point at which to divide the scene. This step, however, does not make a new scene when a color difference between adjacent images shows under a certain threshold.

Real-time Video Retrieval

In this section, we introduce the HySIM algorithm. The HySIM algorithm consists of two processes for high-speed video retrieval. One process includes the construction process of hybrid space. The other process includes the process of a video scene tracking in the HySIM (Figure 7). In the construction of hybrid space, the module makes a scene-segmented, video data set and constructs hybrid space at the same time. In the video scene tracking process, the module searches for a video and then tracks a similar video using a current user's viewpoint query image from the scene-segmented video data set.

An overview of a hybrid space is illustrated in Figure 8. The algorithm uses two spaces. These spaces consist of representative images, each of which well represent all images in a single image segment; continuously observed images are segmented so that similar images are stored in the same segmented scene. One of the spaces is a time-sequential space, which is constructed from representative images. We named this space the C-space. The other space is a feature space that is also constructed from representative images. We named this discrete space the D-space.

Process of a Hybrid-space Construction

The construction process of hybrid space is shown in Figure 9:

1. Segmenting of a video scene and representation of an image from a segmented scene.

Figure 7. Overview of the HySIM mechanism

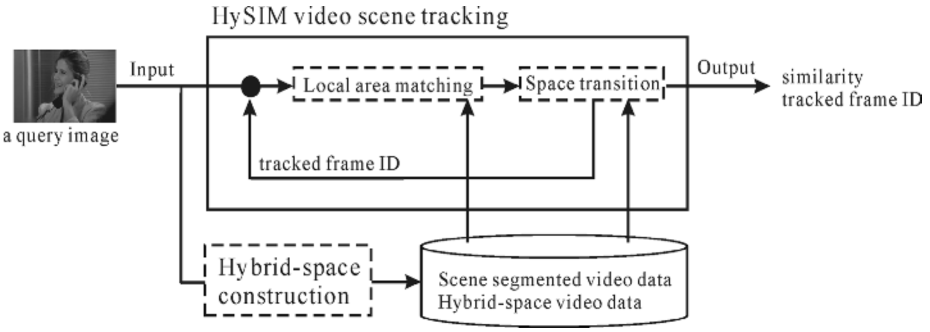
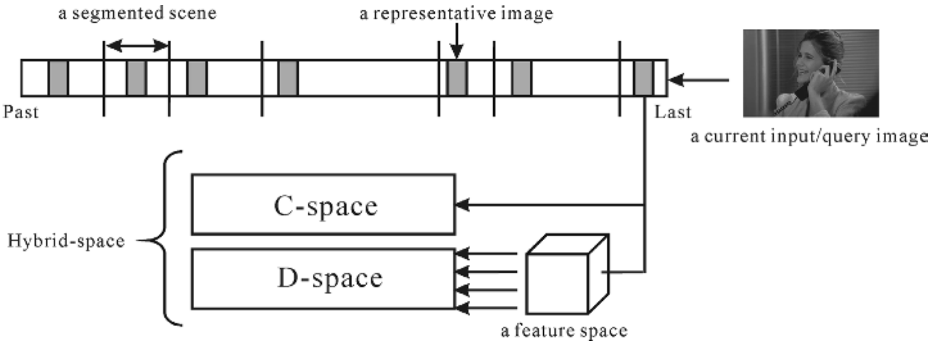


Figure 8. Overview of a Hybrid-space



2. Linking the selected representative image to the last selected image in C-space.
3. Categorizing and linking the selected representative image to the last categorized image in the same space of D-space.

In the *i*th segmentation and representation process, there are n numbers of a representative image candidate that include a current (*t*th) input/query image. A *j*th image in the i th segmentation process is shown $R_i(j)$. A temporal representative image R'_i is calculated by equation (4). Segmentation is performed when an error $\varepsilon_{\max}(i)$, which is evaluated from equation (5), is higher than a threshold *Th*.

$$R'_j = \min_j\{\max_k(|R_i(j) - R_i(k)|)\}, (t-n \le j, k \le t, j \ne k), \quad \textbf{(4)}$$

$$\varepsilon_{\max}(i) = \max_j(|R'_i - R_i(k)|)\}, (t-n \le k \le t). \quad \textbf{(5)}$$

After the selection of the representative image, a hybrid space is constructed using the representative image R_i. The C-space is reconstructed by linking representative

Figure 9. Construction of a Hybrid-space

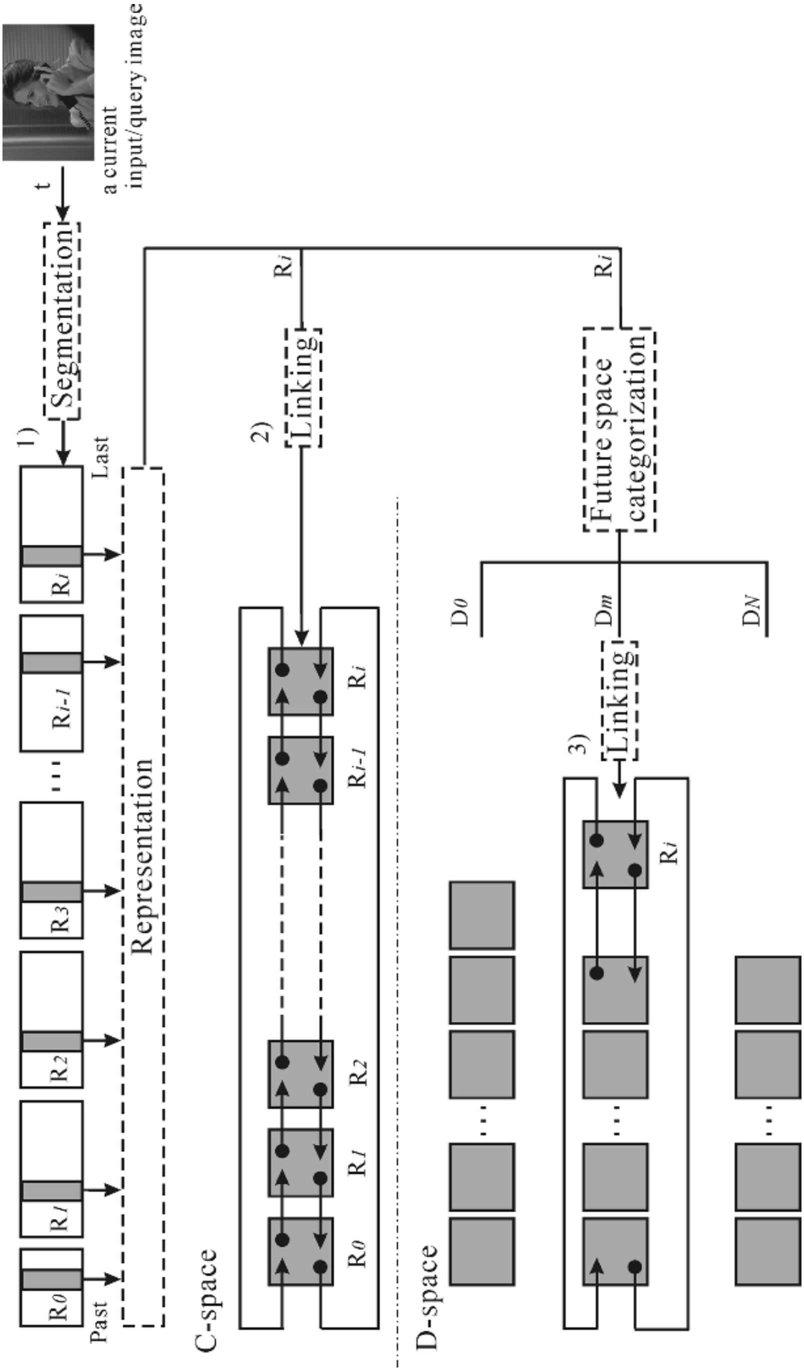
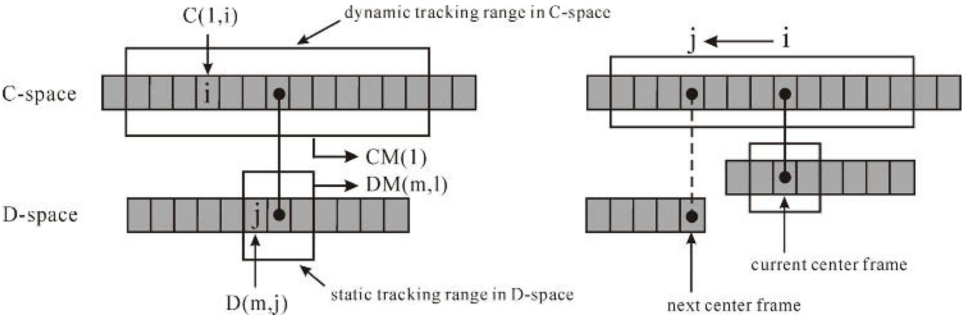


Figure 10. Construction of Hybrid-space



images. R_i is linked to R_{i-1} and R_O, and R_{i-1} is unlinked from R_O. In the construction of the D-space, the down sampled Rd_i is categorized in a feature space. The representative image is then linked to the past representative image and the last image in the local space D_m of the D-space.

The Process of a HySIM Video Scene Tracking

A basic calculation method for a HySIM video scene tracking is shown in Figure 10. The D-space has N dimension in this paper. A representative image is stored in both spaces. In the C-space and D-space, the tracking at a certain moment is performed within a dynamic range (*CR*) and a static range (*DR*), respectively. A value of the static tracking range in the D-space is set in advance. *D*(*m*, *j*) represents a similarity between a representative image *j* and a current query image in dimension *m*. *D*(*m*, *j*) is identical to *C*(*j*, *j*). Let *l* denote the center frame of a tracking range in C-space and D-space. *C*(*l*, *i*) represents a similarity between a representative image *i* and a current query image in the frame *l*. In this study, all similarities have a range from 0.0 to 1.0. In the tracking range determined by *l*, values of the *CM*(*l*) and *DM*(*m*,*l*) are calculated by the following equations:

$$CM(l) = \max_i (C(l,i)), (l + (CR+1)/2 \le i \le l - (CR+1)/2), \quad \textbf{(6)}$$

$$DM(m,l) = \max_i (CM(i)). (l + (CR+1)/2 \le i \le l - (CR+1)/2). \quad \textbf{(7)}$$

The dynamic tracking range *CR* depends on an entrance point *C*(*l*,*l*) similar to equation (8).

$$CR = \alpha \cdot A + (1-\alpha) \cdot A \cdot (1 - C(l,l))^2 . (\alpha, A : const) \quad \textbf{(8)}$$

Here, *A* represents a basic range and α represents a flexibility rate.

The next tracking step can set a new center frame by using the evaluation of the previous step and the following rule of a space transition of the tracking area.

$$l = \begin{cases} i & \exists_i \{C(l,i) \equiv CM(l),\ l-(CR+1)/2 \le i \le l+(CR+1)/2,\ i \ne j\} \\ j & \exists_i \{CM(j) \equiv DM(m,l),\ l-(CR+1)/2 \le j \le l+(CR+1)/2\} \end{cases} . \qquad (9)$$

Experiments and Results

Image Matching Excluding Motion Information

This experiment took place outdoors, in a hall, and in a room during the daytime. The experimental tasks were recorded four times in each place. The first image frame in the recorded video data is defined as the retrieval query. The experimental task consists of two parts. One task required the user to wave his/her hand several times. The other task required the user to turn his/her head to the left. A normal method that does not consider motion information was performed to compare with our proposed method. The result is shown in Figure 11. In this figure, a higher similarity corresponds to a lower evaluation value. Our proposed method clearly shows a higher similarity than the normal method in the hand waving section. In comparison, eliminating noises caused by turning head was less effective than that by moving a hand in our method.

Table 1 illustrates the precision and recall rates. The precision rate represents the rate of retrieved data for all correct data. The recall rate represents the rate of correct data for all retrieval data. Therefore, our proposed method is well suited to retrieving similar location-based images because the relevance rate of the proposed method performed 1.3 times better than the normal method.

Figure 11. Evaluation and targeting of video similarity between base image

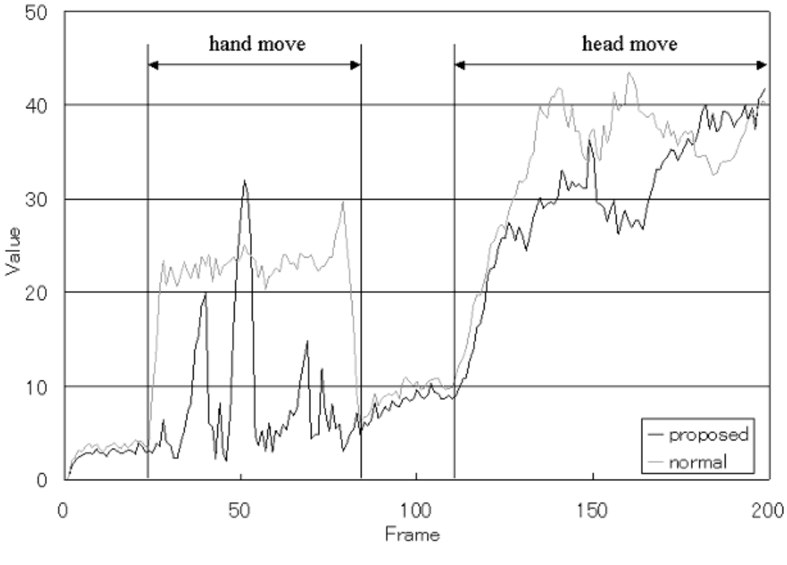


Table 1. Precision and recall rate

	Precision		Recall	
	Proposed	Normal	Proposed	Normal
Outdoor	90%	54%	98%	96%
Hall	97%	56%	92%	90%
Room	88%	57%	97%	96%
Total	92%	56%	96%	94%

Video Scene Segmentation Using Motion Information

Both image and gyro data, which consist of 5584 frames (191.36 seconds, 29.19 frame/second), were recorded for evaluation. A subject walked around a certain route two times in our laboratory (Figure 12). We set intervals of the moving average as $T = 30$ and $T = 300$. This process limits the minimum scene length to 30 frames. This remarkable result of the experiment is shown in Figure 13. The upper lines in the figure are the moving average values. The lower line shows the detected scene changes. These scene changes took place nine times in the figure.

The following results (5584 frames) were obtained: 41 total scenes were constructed from a video data set in the experiment. The average length of the constructed scenes was 119.51frames (4.09 seconds). The minimum and the maximum length of these scene were 31 frames (1.06 seconds) and 756 frames (25.90 seconds), respectively. The minimum scene was made when the subject walked quickly through a narrow path. The maximum scene change was constructed on a long, straight, and undiversified hallway. From this experimental result, we conclude that the user's motion information can be used to construct scene changes on a wearable computer environment.

Figure 12. The walking route of the test subject

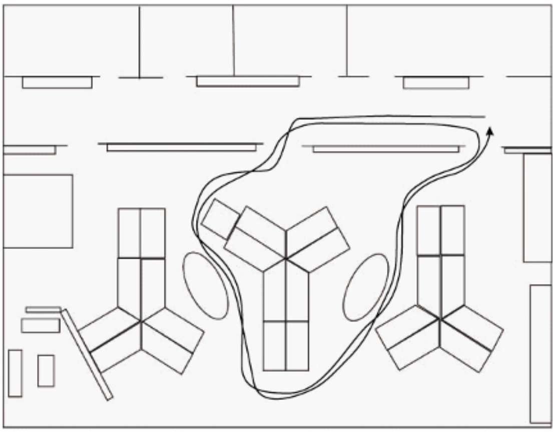
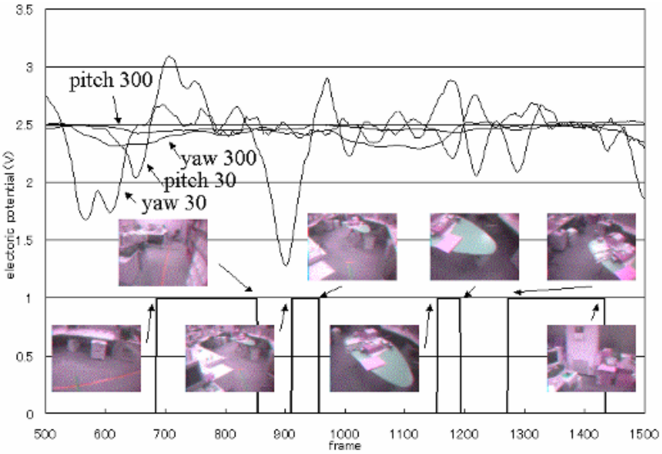


Figure 13. Making scene changes using two mono-asix Gyro sensors



Real-time Video Retrieval

The following experiments were conducted to evaluate the video retrieval performance of the HySIM algorithm. In these experiments, the HySIM algorithm is compared with a full-search algorithm, which uses only representative images. We have evaluated retrieval speed and retrieval accuracy.

In these experiments, one test subject wore a wearable computer, and continuously walked around for approximately an hour in a building with three halls and two laboratory rooms. Input query images were captured frame by frame. We used the same images for the query data set as for the recorded data set. 100,000 images were captured for approximately one hour. The parameter α is set at 50 in this experiment. The parameter *DR* value is equal to 15 per trial.

Table 2 shows the performance result of these experiments, in which the data column shows the number of images. The *R*-frame illustrates the amount of representative images

Table 2. A comparison between the full search and the HySIM search

Data (frame)	R-frame (frame)	Full (sec)	HySIM (sec)	Accuracy rate (%)
10000	1257	.0106	.0021	69.48
20000	2254	.0189	.0022	69.90
30000	6235	.0515	.0031	34.07
40000	10138	.0835	.0035	35.73
50000	11142	.0920	.0033	34.70
60000	12568	.1037	.0033	31.93
70000	13724	.1138	.0031	34.69
80000	15313	.1263	.0031	41.92
90000	16520	.1374	.0030	37.61
100000	21039	.1744	.0033	30.33

in a trial. Both full and HySIM methods show the video data processing times required to retrieve a similar image with an input query image, respectively. The accuracy rate is calculated by the number of frames that have a similarity over the threshold *Th* (set at 0.95 in the experiment) in the scene segmentation, or as the best similarity in all recorded images.

The processing time of the full algorithm shows a linear increase. The processing time of HySIM increases approximately 1.57 times when the data of 10,000 frames increases to 100,000 frames. The processing time of HySIM is 52.20 times as fast as that of the full algorithm in the data of 100,000 frames.

Summary

We introduced three methods for supporting a location-based user's recollection in the *Residual Memory* module. First, we proposed the stable image matching method involving head movement and a moving object in a scene. We employed two mono-axis gyro sensors to track a two-axis user's head movement. In addition, we proposed the video scene segmentation method to create comprehensible video scenes for the user. We introduced the moving average method by using two mono-axis gyro sensors. Last, we proposed a real-time video retrieval method named HySIM. We introduced two different types of feature spaces: a time-sequential space, and an image-feature space with a similar color feature value.

A further direction of this study will be to develop a faster, more stabilized, and more efficient video retrieval method. Additional types and numbers of perceptual sensors will need to be attached to achieve this goal.

OBJECT-BASED VIDEO MEMORY RETRIEVAL

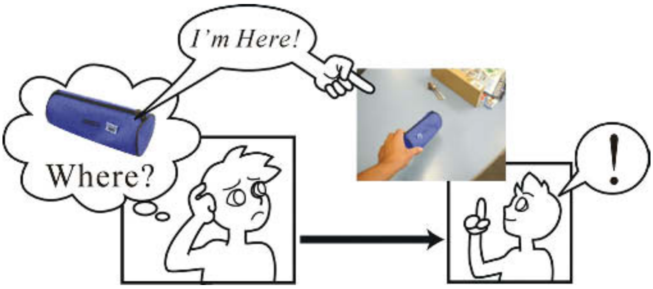
A Brief Concept

This section discusses a wearable module to support a user's ability to remember where he/she placed an object used in everyday life. As depicted in Figure 14, the proposed module named "*I'm Here!*" shows its user a video recorded when he/she last held a target object. Viewing the video, the user can remember where and when he/she placed the object. Ultimately, we expect that the module will act as if the object itself sends a message such as "*I'm Here!*" to the user (Ueoka, Kawamura, Kono, & Kidode, 2003).

Also supporting a user's ability to remember where objects are placed in the real world (Shinnishi, Iga, & Higuchi, 1999) is a study of an interface device called "Hide and Seek," a proposed module with small devices attached to real objects. When a user sends a message to an object, the object reacts to the user by a variable frequency sound. However, attaching devices that require electric power to real objects is a critical problem for everyday use.

The *I'm Here!* module, however, is a stand-alone, wearable module without any devices that need to be attached. Moreover, this module has achieved the simple registration of both portable and rigid objects in a user's everyday circumstances. In

Figure 14. The concept of 'I'm Here'



addition, the module automatically associates video memory with the IDs of the continuously observed objects if they have already been registered. The user then can view the video recorded when he/she last held the target object by simply identifying the object by its name.

Module Design

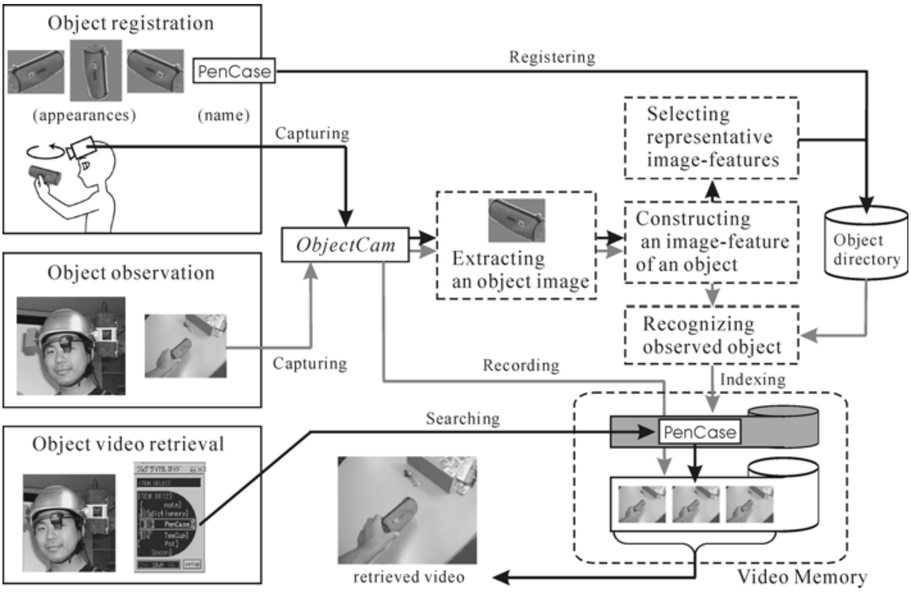
I'm Here! continuously captures a user's viewpoint video, and continuously identifies the object held by the user. The module then constructs and records the video memory, which is video data of a user's viewpoint along with the index of the detected objects. Using the video memory, the module can retrieve the latest video of the target object so that the user can remember where he/she last placed the object. To create a video memory, a dictionary, which includes the appearance-based image-features of the preliminary registered objects, is registered.

The module hardware consists of a wearable PC, a Jog-dial interface, and a wearable display with a camera device named "*ObjectCam*." The video captured by the *ObjectCam* is stored in the wearable PC.

As depicted in Figure 15, the module provides the user with the following three operation phases:

- **Object Registration.** In this phase, the user registers target objects by simply holding and gazing at the object with a rotational operation to construct the object dictionary. The dictionary consists of the name, the ID, and appearance-based image-features of each object.
- **Object Observation.** In this phase, the module automatically captures and continuously records a video of the user's viewpoint. The module simultaneously identifies the object held by the user to construct the video memory, which is the video data with the index of the observed object's IDs.
- **Object Video Retrieval.** In the object video retrieval phase, the user simply selects the name of the object he/she wants to remember. The module searches for the name of the object in the video memory and retrieves the last recorded video associated by the object.

Figure 15. Overview of operation phases



Object Extraction from a User’s Viewpoint Image

The image of a user’s viewpoint includes the object region, the region of the hand holding the object, and the background region. To construct an appearance-based image-feature of the object, the region of the object must be extracted from the image by eliminating both the hand region and the background region.

Many studies of extracting a target region by eliminating a background region exist. The background subtraction method has been used in many real-time vision systems with fairly good results (Horprasert, Harwood, & Davis, 1999). However, this method requires a static background image that can rarely be obtained by a wearable camera. On the other hand, top-down knowledge from object recognition that can be used for segmenting the target region from exterior regions has also been discussed (Leibe & Schiele, 2003b). This method is suitable for images with cluttered backgrounds and partial occlusions. However, applying this method to real-time applications is difficult because the temporal loads from preliminary learning of knowledge and iterative processing are required in this method.

ObjectCam

We have developed a new camera device named “*ObjectCam*” to extract only the object image from the user’s viewpoint image. This camera device enables low-cost object image extraction regardless of background complications.

Figure 16 illustrates the architecture of the *ObjectCam*. The camera device consists of a color camera and an infrared camera clamped at the same posture across a beam splitter. A frame of the image captured by the *ObjectCam* consists of a color field image

Figure 16. Wearable devices and the structure of ObjectCam

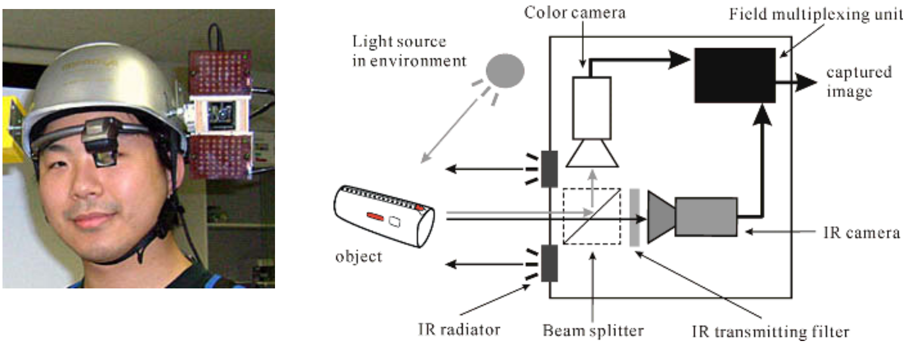
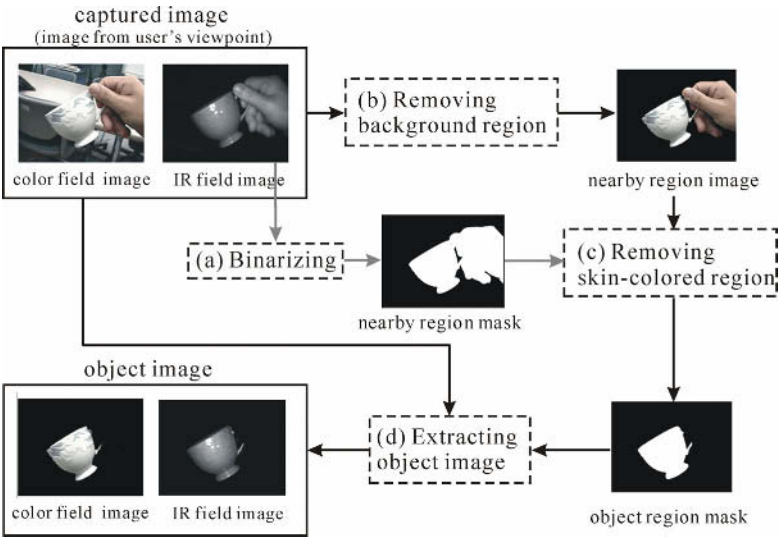


Figure 17. Blockgram of extracting the object image



and an infra-red (IR) field image. An IR field image displays the reflected IR luminance caused by the IR radiator on the front of the device.

Object Extraction Using the ObjectCam

Figure 17 illustrates a blockgram of the object extraction process. First, a nearby region mask is made from an IR field image with a luminance threshold in a binarizing process (Figure 17a). Second, by applying the mask to remove the background region from the color field image (Figure 17b), the module creates a nearby region image. Third, the object region mask is created by removing the region of the hand holding the object using the user's skin color (Figure 17c). Finally, by applying the object region mask to both color and IR field image of the user's viewpoint, the module creates an image of the object (Figure 17d).

Object-feature and Similarity

Both object registration and recognition processes are necessary for retrieving the last recorded video containing the target object from the video memory database. Many object recognition methods have been proposed. The appearance matching of three-dimensional objects using parametric eigenspace features, for instance, has been estimated with good results (Schiele & Crowely, 1996). However, a large load is necessary to compress large amounts of input images into the eigenspace features, and a strict facility to register the eigenspace features is required. On the other hand, probabilistic object recognition using the feature of multidimensional histograms has been applied to estimate the probabilistic presence of the registered objects in the scene (Nayar, Nene, & Murase, 1996). Because the experimental result mentions only the scale change and image-plane rotation of the input image, the variety of appearances of the three-dimensional object still remains a considerable problem.

Several problems regarding the recognition of the object held by a user in everyday circumstances still need to be considered; for example, (1) real-time registration and recognition of the object, (2) the ability to recognize a three-dimensional object in several appearances, and (3) the problem of achieving good performance of object recognition in the case of increasing the amount of registered objects. The study proposing an appearance matching of three-dimensional objects using parametric eigenspace features has the disadvantage of (1), and the other study of probabilistic object recognition using the feature of multidimensional histograms does not clearly address. (2) In the development of *I'm Here!*, a multi-dimensional histogram feature extracted from object images captured by *ObjectCam*, which includes color and IR luminance data is proposed to solve problems (1) and (2), and the proposed feature is estimated from the perspective of (3) in the section, Experiments and Result.

Constructing an Object Feature with Object Images

The object dictionary contains the appearance-based image features of the registered objects with their IDs. An object feature of a three-dimensional object is constructed from several images in representative appearances. To construct the object feature, the module captures several images of the object, makes an image feature representing each image, and integrates these image features into an object feature by grouping and integrating similar image features.

Extracting an Image Feature

An object image feature is denoted by a $\{H - Z - C\}$ three-dimensional histogram. This histogram consists of $\{H, Z, C\}$ elements extracted from each pixel of the object image. H and Z represent Hue and IR luminance value. C represents the pixel group ID divided by distance from the centroid of the silhouette of the object image. In addition to the hue values as a color feature, the $\{H - Z - C\}$ histogram includes IR luminance as a depth-like feature and C as a silhouette-like feature. Due to each feature's robustness for the rotation of the view axis and the low-cost processing of this histogram feature, the $\{H - Z - C\}$ histogram is expected to be a good image-feature in object recognition.

The hue value is based on HSV color representation converted from the RGB values of a pixel. The module extracts hue value *H* using the expressions below:

$$V = \max(R, G, B), \tag{10}$$

$$W = \min(R, G, B), \tag{11}$$

$$S = \alpha\left(\frac{V - Z}{V}\right), \tag{12}$$

$$H = \begin{cases} \beta\left(\dfrac{G - B}{V - Z}\right), & R \equiv V \\ \beta\left(2 + \dfrac{B - R}{V - Z}\right), & G \equiv V \\ \beta\left(4 + \dfrac{R - G}{V - Z}\right), & B \equiv V \end{cases} \tag{13}$$

S represents the saturation value, and *V* and *W* represent the maximal and minimal value among the {*R,G,B*} values of the pixel. These values are limited as $0 \le R,G,B,S,V,W < \alpha$ and $0 \le H \le \beta$.

The hue value with low saturation is sensitive to sensor noise, however. According to the sensor noise, the {*R,G,B*} and {*H,S,V,W*} color features oscillate on their own order. Furthermore, the hue value becomes uncertain as the saturation value decreases. This creates a problem.

To avoid the influence of the sensor noise, we apply the probabilistic sensor noise model to the hue value. The distribution of sensor noise is assumed to the Gaussian distribution, as denoted below when the average of the distribution is *u* and the dispersion is σ^2:

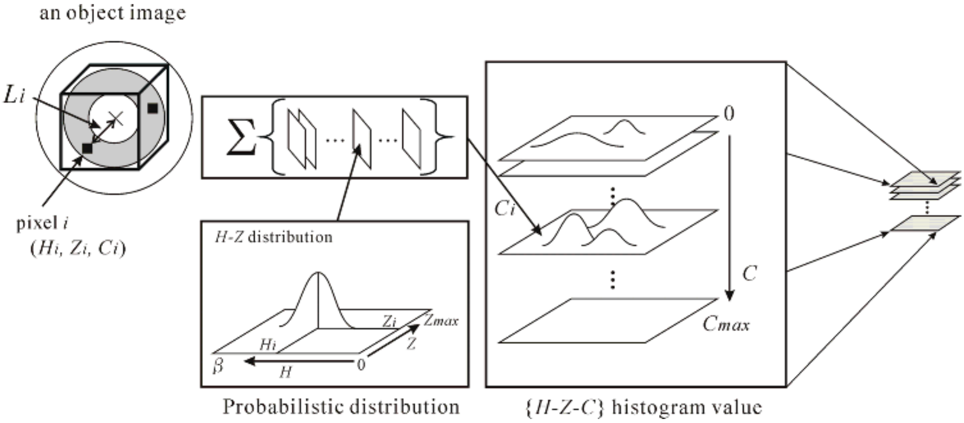
$$K(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-u)^2}{2\sigma^2}}. \tag{14}$$

Using equation 13, the dispersion of *H* is represented as:

$$\sigma_H = \begin{cases} \sigma_{RGB}(G, B), & R \equiv V \\ \sigma_{RGB}(B, R), & G \equiv V \\ \sigma_{RGB}(R, G), & B \equiv V \end{cases} \tag{15}$$

The element σ_{RGB} in equation (15) is represented as below:

Figure 18. Image-feature of an object as the {H - Z- C} three-dimensional histogram



$$\sigma_{RGB}(x,y) = \alpha\beta\sqrt{\frac{(x-y)^2(V^2\sigma_S^2 + S^2\sigma_V^2) + S^2V^2(\sigma_x^2+\sigma_y^2)}{V^4S^4}}. \tag{16}$$

The dispersion of the {*S*,*V*,*W*} value represented as {σ_S, σ_V, σ_W} can be directly derived from the statistical observation of the {*R*,*G*,*B*} distribution.

Using equation (14), (15), and (16), the uncertainty of the hue value with low saturation can be represented; in the case of zero saturation, the probabilistic hue distribution becomes flat.

An image feature of an object is the integrated distribution of all the pixels of the image represented as the {*H* - *Z* - *C*} histogram. Figure 18 represents the construction of the {*H* - *Z* - *C*} histogram from an object image when *i*th pixel of an object image has the hue value of H_i , the IR luminance value of Z_i, the group ID of C_i, and the distance from the centroid of the silhouette of the object image of L_i. The value of C_i is denoted as below:

$$C_i = \left[\frac{L_i}{L}\cdot\sqrt{\frac{n_0}{n}}\right]. \tag{17}$$

L represents a standard distance. *n* equals the entire amount of all pixels of the object image and n_0 represents the normalized amount of pixels. As depicted in Figure 18, the distribution has a hue axis, an IR luminance axis, and an axis of the pixel group number. The values of *Z* and *C* are limited as $0 \le Z \le Z_{\max}$ and $0 \le C \le C_{\max}$. The distribution value in {*H* = *j*,*Z* = *k*,*C* = *l*} is represented as below:

$$D(j,k,l) = \sum_{i=1}^{n} f_H(i,j)\cdot Z_{ik}\cdot C_{il}\ , \tag{18}$$

$$Z_{ik} = \begin{cases} 1, & k = Z_i \\ 0, & k \neq Z_i \end{cases}, \qquad \textbf{(19)}$$

$$C_{il} = \begin{cases} 1, & l = C_i \\ 0, & l \neq C_i \end{cases}. \qquad \textbf{(20)}$$

$f_H(i,j)$ represents a probabilistic contribution of ith pixel for the j th bin of the hue value represented as below:

$$f_H(i,j) = \frac{1}{n}\int_{h(j-1)}^{hj} K_i^H(x)dx, \qquad \{ j \mid j = 1, 2, ..., \frac{\beta}{h}\}. \qquad \textbf{(21)}$$

h is the breadth of the bin of the hue (Kawamura, Ueoka, Kiuchi, Kono, & Kidode, 2003).

Grouping Image-features for Object Registration

In the object registration phase, the module constructs an object feature from several appearances of an object as a set of selected representative image features (Figure 19). The selection of representative image features is based on a grouping of similar image features of an object.

The module captures several images of the object and calculates the image features. We can reduce the amount of the elements of an object feature by grouping similar image features into a representative image feature to avoid a redundant comparison in identifying the object.

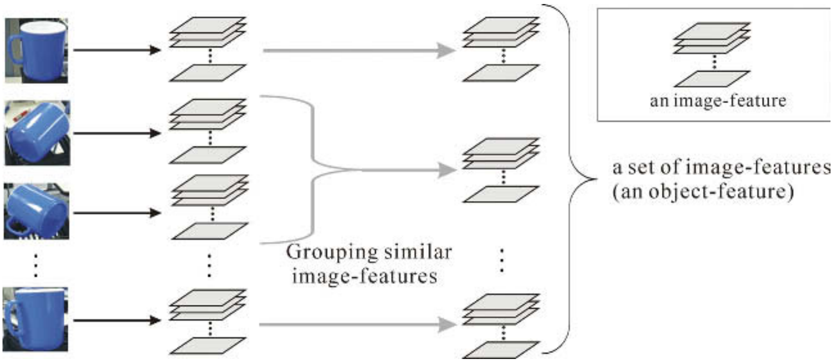
The similarity of image features D_i and D_j is based on the Sum of Absolute Difference (SAD) of three-dimensional histograms, as denoted by SAD* (D_i, D_j) in the equations below:

$$SAD^*(D_i, D_j) = \begin{cases} \sum_{h,z,c} \left| D_i(h,z,c) - D_j\left(h, \left(z + z_m(i) - z_m(j)\right), c\right) \right|, & 0 \leq \left(z + z_m(i) - z_m(j)\right) \leq Z_{\max}, \\ \sum_{h,z,c} \left| D_i(h,z,c) \right|, & other. \end{cases} \qquad \textbf{(22)}$$

When $D_x(h,z,c,)$ is reintegrated into $D_x^Z(z)$ on the basis of IR luminance, $z_m(x)$ represents the centroid of $D_x^Z(z)$. $z_m(x)$ and $D_x^Z(z)$ are denoted as follows:

$$\begin{aligned} z_m(x) &= \frac{1}{N}\sum_z D_x^Z(z), \\ D_x^Z(z) &= \sum_{h,c} D_x(h,z,c). \end{aligned} \qquad \textbf{(23)}$$

Figure 19. From several appearances to an object-feature



The SAD* is derived by aligning distributions with their centroid on the basis of IR luminance values, because the reflected IR luminance value is experimentally observed in linear relation to the distance from the object's surface.

The following equations define the clustered structure of the images of the object:

$$\begin{aligned} &\mathrm{A}^{Uj} = \{A_{ij} \mid 1 \le i \le N_j\}, \\ &\mathrm{A}^{U1} = \{\mathrm{Q}_i \mid 1 \le i \le N_1\}, \\ &\mathrm{A}^{U(j+1)} = A^{Uj} - \mathrm{Q}_j. \end{aligned} \qquad \textbf{(24)}$$

A^{Uj} represents the *j*th set of images of the object. A^{Uj} includes all images of the object. A_{ij} represents the *i*th image included in A^{Uj}. N_j represents the amount of images included in A^{Uj}, and is bounded in $1 \le N_j \le N_1$. Q_i represents the *i*th cluster of similar images included in A^{U1}. When R_j is the selected representative image of A^{Uj}, R_j is denoted as follows:

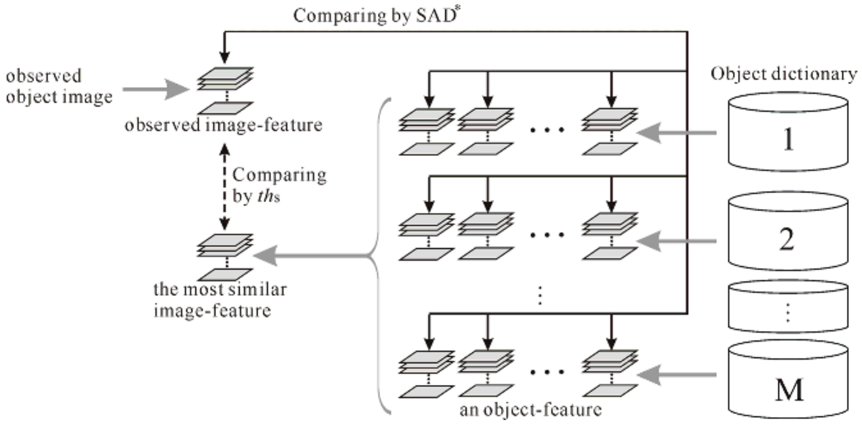
$$R_j = A_{1j}. \qquad \textbf{(25)}$$

When the image feature of A_{kj} is D_{kj} and the image feature of a representative image R_j is $D1_j$, the following equations define the contents of Q_j with equation (22):

$$\begin{cases} A_{kj} \in \mathrm{Q}_j, & SAD^*(D_{1j}, D_{kj}) < th_q, \\ A_{kj} \notin \mathrm{Q}_j, & other. \end{cases} \qquad \textbf{(26)}$$

th_q represents the threshold for Q_j to content images similar to R_j. The set of selected representative images R, is defined as follow:

Figure 20. Overview of the comparison between features for object recognition



$$R = \{R_i \mid 1 \le i \le N_R\}. \quad (27)$$

N_R represents the amount of selected representative images of the object bounded in $1 \le N_R \le N_1$.

Comparing Image-features for Object Recognition

In the object observation phase, the module tries to recognize an observed object along with any of the registered objects. In the recognition process, the observed object feature is compared with all registered object features in terms of similarity (Figure 20).

We defined the similarity between observed object *o* and a registered object *r* using equation (22) as below:

$$e_r = \min\{\mathrm{SAD}^*(D_o, D_p) \mid p = 1,2,\ldots.., N_r\}, \quad (28)$$

D_o represents the image feature of the observed object *o*, and D_p represents the *p*th image feature of the registered object *r*. N_r represents the amount of selected representative images of object *r*.

The result of object recognition is represented as below:

$$Obj = \begin{cases} r, & \min\{e_r \mid r = 1, 2, \ldots, M\} < th_s \\ 0, & other. \end{cases} \quad (29)$$

M represents the amount of registered objects, and th_S the threshold of minimal similarity. The result 0 shows that there are no registered objects matched with the observed object.

Video Retrieval with Video Memory

Through the object observation phase, the module creates the video memory, which is the video data with the index of observed objects. The index is associated with every frame of the video memory. If a frame of captured video contains a registered object, the index explicitly annotates to the frame with the ID of the object.

In the object video retrieval phase, the module has to display the segmented video using the video memory. To segment the recorded video, the module sets a start point p_s and an end point p_e on the recorded video sequence on the condition that the last observed point of the target object is p_o, and the time length of the segmented video is U:

$$p_s = p_o - \frac{U}{2}, \qquad \textbf{(30)}$$

$$p_e = p_o + \frac{U}{2}. \qquad \textbf{(31)}$$

Using the segmentation rule, the retrieved video includes both forward and backward information from when the object was observed.

Experiments and Results

Since the object recognition performance affects the accuracy of retrieved video, we have estimated the object recognition performance of the proposed method by a few experiments. These experiments were performed in an indoor environment with fluorescent light. The module is estimated in offline usage, by setting *ObjectCam* on its base and by setting a target object on a turntable.

The sets of target objects are depicted in Figure 21. Set (a) consists of ten objects, and Set (b) consists of twenty objects including the objects in (a) and an additional ten objects. Each object is rigid, the appropriate size to hold, and used in everyday life.

In off-line object registration, the module captures twenty images for each object as input images for object registration. Each of the twenty images is in a different configuration of distance and perspective. The module creates an object dictionary with selected representative image features and an ID of each object.

Instead of the object observation phase, the module calculates an object recognition rate with every registered image to be observed; that is, the object dictionary consists of all the registered object features in these experiments. The module rejects matching between different objects and allows matching between any configuration patterns in the same object.

To demonstrate the advantage of the proposed method, we have compared the $\{H\text{-}Z\text{-}C\}$ histogram method with other methods using $\{H\}$, $\{H\text{-}Z\}$, and $\{H\text{-}C\}$ histogram methods in these experiments. The parameters are set as follows: $\alpha = 256$, $\beta = 360$, $Z_{\max} = 255$, $C_{\max} = 10$, $L = 10$, $n_0 = 1000$ and $h = 4$. Table 3 displays the result of the experiments. We found that in the proposed $\{H\text{ - }Z\text{ - }C\}$ histogram method, the recognition rate decreases less than in other methods even when the amount of objects in the target set increases. In everyday use of *I'm Here!*, the amount of registered objects increases. The

Figure 21. Sets of target objects

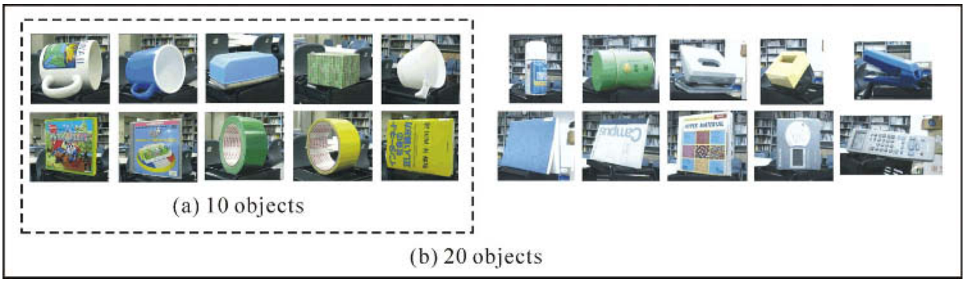


Table 3. Experimental results

Object feature value	{*H*}	{*H-C*}	{*H-Z*}	{*H-Z-C*}
(a) 10 objects	99.2%	99.2%	91.7%	96.7%
(b) 20 objects	81.7%	88.8%	87.5%	94.1%

experimental result demonstrates that the proposed method is useful for online object recognition in *I'm Here!*.

Summary

We have proposed a wearable module named "*I'm Here!*" to support a user's ability to remember where he/she placed a target object using video memory. Automatic construction of the video memory using the proposed {*H* - *Z* - *C*} histogram method requires real-time and accurate recognition of the observed object. To estimate the accuracy of the recognition method, off-line experiments have been performed. The experimental results suggest that the proposed method is useful for object recognition of the module.

Some issues remain for achieving an online module of *I'm Here!*:

- **Speeding up the recognition process.** We define the goal of the processing speed as 15fps (a quarter of the video rate).
- **Performing experiments with more registered objects.** Ultimately, we expect that the module will be able to identify 100 objects.
- **Segmenting more appropriate retrieved video for a user.** his problem is closely associated with the contents of the recorded video. The solution requires extensive analyses of the contents as well as a field trial of the module using questionnaires.
- **Producing a polished interface for object video retrieval.** The current interface is based on immediate direction to the name of the target object. Related studies have shown that knowledge of object categorization is useful for estimating and improving object recognition methods (Leibe & Schiele, 2003a). We expect that knowledge of object categorization is also useful for supporting the selection of the target object so that the user can select the category of the target object.

HOW TO MANAGE VIDEO MEMORY IN THE REAL WORLD

This section introduces the *Ubiquitous Memories* module to support Memory Retrieval, Memory Transportation, and Memory Exchange functions in *SARA* (Kawamura, Kono, & Kidode, 2002; Kono et al,, 2003). The primary motivation of the study is to enable wearers to manage everyday memories in the real world. In order to accomplish this motivation, we have proposed the concept of *Ubiquitous Memories* and developed a prototype module that can associate augmented memories with a physical object in the real world using a "touching" operation (Kawamura, Fukuhara, Takeda, Kono, & Kidode, 2003).

Conceptual Design

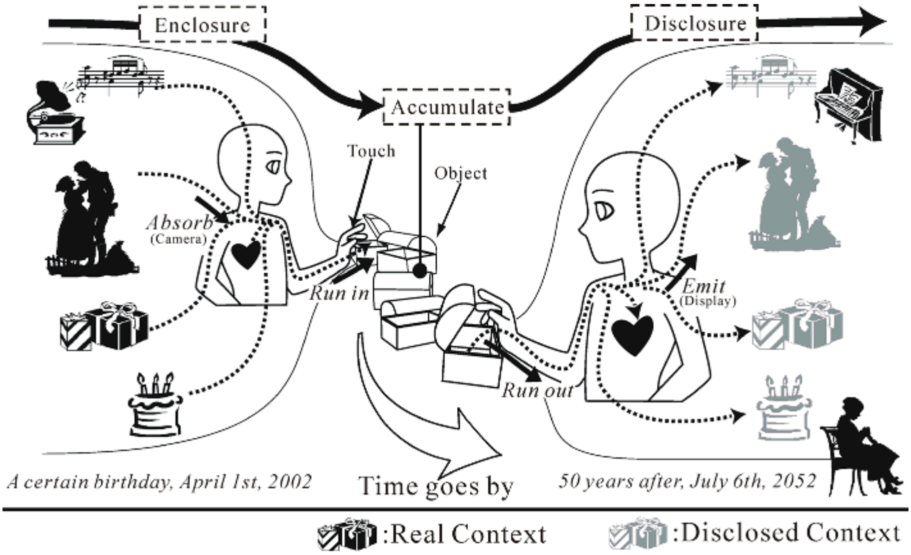
We propose a conceptual design to ideally and naturally correspond augmented memory to human memory. Conventionally, a person often perceives and understands a new event occurring in the real world by referring to his/her experiences and knowledge, and then storing the memory of the event into his/her brain. He/she then obtains a novel and natural action for the event by analogically and metaphorically associating the event with previously occurring events. We believe that the acquisition of natural actions is important for realizing augmented memory. This acquisition positively establishes a "conceptual design" for seamless integration between human memory and augmented memory. In addition, the "Hand" interface has the potential for integrating augmented memory into objects.

Below we introduce the conceptual design of the *Ubiquitous Memories* module. The following procedures illustrate the conceptual design:

1. A person perceives an event via his/her body.
2. The perceived event is stored into his/her brain as a memory.
3. The human body is used as media for propagating memories, i.e., the memory travels all over his/her body like electricity, and the memory runs out of his/her hands. (Imitating this feeling, he/she can transfer the memory from his/her body to a physical object by "touching").
4. The transferred memory remains in the object.
5. He/she transfers the memory from the object to his/her body when he/she is interested in the object and touches it again.
6. Finally, he/she can recall the event.

In this chapter, we define "Context" as information the human can sense in the real world, for example, atmosphere, the observer's emotional condition, and the biometric states of the observer. Note that a context is not data like a video memory. The "Human Body" and an "Object" are important for our concept in realizing ubiquitous memories. Both the "Human Body" and "Real-World Objects" are essential device/media for augmenting human memory in *Ubiquitous Memories*; that is, the human body behaves as a device/media that associates video memory with objects. The terms of the conceptual actions shown in Figure 22 are defined as follows:

Figure 22. Concept of Ubiquitous Memories



- **Enclosure** action is shown by two steps of behavior: (1) a person implicitly/explicitly gathers current context through his/her own body, and (2) he/she then arranges contexts as ubiquitous augmented memory with a real-world object using a touching operation. The latter step is functionally similar to an operation that records video data to a conventional storage media, for example, a videotape, a CD-R/W, or a flash memory. The two steps mentioned above are more exactly defined as the following actions:
  - **Absorb:** A person's body acquires contexts from an environment, his/her own body, and his/her mind, as moisture penetrates into one's skin. Such an operation is called "Absorb" and is realized by employing real world sensing devices, e.g., a camera, a microphone, and a thermometer.
  - **Run in:** When a person touches a real-world object, an augmented memory flows out from his/her hand and runs into the object. A "Run in" functionally associates an augmented memory with an object. In order to actualize this action, the module must recognize a contact between a person's hand and the object and must identify the object.
- **Accumulation** denotes a situation in which video memories are enclosed in an object. Functionally, this situation represents how the augmented memories are stored in storages with links to the object.
- **Disclosure** action is a reproduction method where a person recalls the context enclosed in an object. "Disclosure" has a meaning similar to that of replay (for example, the way a DVD player runs a movie). This action is composed of the following actions: "Run out" and "Emit."

- **Run out:** In contrast to "Run in," video memory runs out from an object and travels into a person's body. Computationally, the "Run out" identifies the storage space where the augmented memories' linked objects are stored, and retrieves these memories from the Internet to a user's wearable PC. In order to achieve this action, the module needs contact and object identification functions such as "Run in." In addition, the module must have a retrieval function to refer to augmented memories associated with an object.
- **Emit:** A wearer can restore contexts in an environment to his/her body, and mind. The module should employ devices, for example, a video display and a headset that can play back an augmented memory.

Enclosing an augmented memory in an object memory-seeking behavior directly corresponds to an object-searching behavior where the object is associated with the memory in some scene. This correspondence gives a wearer more intuitive power to seek for an augmented memory using the principle of human memory encoding (Tulving & Thomson, 1973). For example, suppose that a person won first prize in the 100-meter dash at an athletic festival and then got a plaque. A person can easily recall the event when he/she simply looks at the plaque because he/she has associated the event with the plaque in his/her mind. This associative ability is called the *Encoding Specificity Principle*. Two detailed characteristic traits exist for the principle when expressed in an object-seeking action. One characteristic trait is the ability to recall an event or feeling or emotion by simply looking at or thinking about an object. This associative trait allows one to decide quickly what object he/she should seek. Another trait is the remembrance of placed location of the object. This trait allows a person to remember where he/she placed an object. These associative traits illustrate how we can easily recall an event by seeking out an object related to that event.

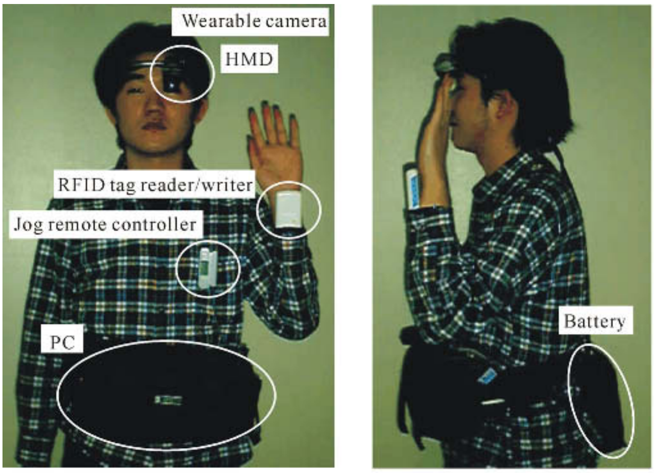
A person's touching operation is employed not only for realizing metaphors that a human hand implies ("Run in" and "Run out"), but also for naturally controlling an augmented memory module. Nonetheless, explicitly selecting an object for both human and computational devices makes it easy to for the user to express his/her distinct intentions by "touching."

Hardware

Wearable Equipment for the Ubiquitous Memories Module

Figure 23 shows the equipment worn with *Ubiquitous Memories*. The user wears a Head-mounted Display (HMD; SHIMADZU, DataGlass2) to view video memories and a wearable camera (KURODA OPTRONICS, CCN-2712YS) to capture video memory of his/her viewpoint. The user also wears a Radio Frequency Identification (RFID; OMRON, Type-V720) tag reader/writer on his/her wrist. Additionally, the wearer uses a VAIO jog remote controller (SONY, PCGA-JRH1). To control the module, the wearer attaches RFID operation tags to the opposite side of wrist from the RFID tag reader/ writer. The wearer carries a wearable computer on his/her hip. The RFID device can immediately read an RFID tag data when the device comes close to the tag. The entire module connects to the World Wide Web via a wireless LAN.

Figure 23. Ubiquitous Memories equipment



Real-World Object Attached to an RFID Tag

We currently assume that an RFID tag is attached to/implanted in each real-world object. We have employed a short-range type RFID device for identifying each real-world object, and for controlling the states of the module. The readable range of the RFID strongly depends on the RFID tag size. A small size tag of less than 1cm allows the reader to read the tag. Data can also be retrieved from a large tag of about 3cm.

Figure 24 depicts a sample image of an RFID tag attached to a cup. The *Ubiquitous Memories* module reads information from an IC of the RFID tag. Table 4 illustrates a tag data protocol to manage video memory. There are two facets of the RFID tag. One facet concerns identifying a certain object by attaching an RFID tag. We have employed a Serial Number (SRN) that is unique to each RFID tag before shipping as an object identification number. Another facet of the RFID tag is the data needed to address a server URL for storing and retrieving video memory by touching a real-world object, and send a command to the module by touching one of operation tags.

Figure 24. RFID tag attached to a cup

Table 4. RFID tag information

Purpose	Object Identification Number	Data
Address for storage	SRN	URL
System Control	SRN	Operation Code

Module Design

A Module Configuration Diagram

Figure 25 shows the module configuration of *Ubiquitous Memories*. This module is composed of a client and multiple servers for a wearer. In this section, we have termed the server a Ubiquitous Memories Server (UMS). The wearable computer plays the role of a client. The core process is UM Control on the client. The Video Buffer Control temporally manages a video memory before it is stored in a database. The Ring Buffer enables the wearer to choose two types of an enclosure operation. As a basic strategy, the wearer encloses a 10 second length video memory from the moment he/she wants to enclose it. On the other hand, the wearer can enclose a 10 second length video memory to a time when he/she wants to enclose it. This module has two types of databases. One is a privacy policy-oriented database that is placed in a wearable computer. We have named that database the Ubiquitous Memories Client Database (UMCDB). Another database is a public/group policy-oriented database. This database represents a server, which is termed the Ubiquitous Memories Server Database (UMSDB), in order to exchange video memory with other wearers.

We have employed a special protocol named the Ubiquitous Memories Transfer Protocol (UMTP) to transport a video memory between a client and a server. Table 5 illustrates the types of message and data used for UMTP. Table 6 shows an actual example of the message and data. In the *Ubiquitous Memories* module, three types of data are transported: Message, Video Memory, and List Data. In Table 5, a parenthesis shows that (Command), (Video Memory), or (List Data) are not needed in all processing. In the Message Type, the module can distinguish three states: “Only Message Information,” “Attaching Video Memory,” and “Attaching List Data.” OID means an object identification number (SRN) recorded in an RFID tag. The UID is a user identification number. The AT shows an attribute of publication to other users. This attribute of publication is equal to a permission used to limit a user who is able to enclose video memory. The GID is a group identification number used for sharing video memories with friends, families, and co-workers. TIME is the time when the wearer first encloses a video memory. Command is equal to an operation code registered in an RFID tag. List Data is written down information (UID, AT, GID, TIME, and a filename of the video memory) of video memories that are enclosed to the same objects (OID).

Figure 25. Ubiquitous memories module configuration

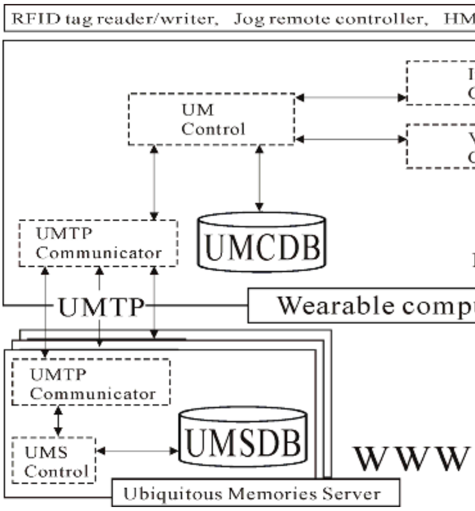


Table 5. Type of message and data transportation

Identification Part	Message Type
Message Part	OID, UID, AT, GID, TIME, (Command)
Data Part	(Video Memory)/(List Data)

Table 6. An example of message and data transportation

Identification Part	DATA
Message Part	0B8BE72400000009, 1000,1,9001, 20030909101231, OENC
Data Part	Data.avi

Operations Using Operation Tags

The *Ubiquitous Memories* module has six operational modes: ENCLOSURE, DISCLOSURE, DELETE, MOVE, COPY, and NONE. Note that the NONE mode means that the module reacts to a wearer's actions only when one of operation tags is touched. There are two basic operation tags and three additional operation tags for changing the mode. The wearer can select one of the following types:

- **ENCLOSURE:** By touching the "Enclosure" tag and an object sequentially, the wearer encloses video memory to an object. In this mode, the functions of "Absorb" and "Run in" are sequentially operated.

- **DISCLOSURE:** The wearer can disclose a video memory from a certain real-world object. In this mode, the "Run out" function and the "Emit" function are sequentially operated.

Using additional operation tags, the wearer can treat a video memory in the real world like paper documents or data on a PC by using the following types of tag:

- **DELETE:** The wearer can delete a video memory enclosed in a certain object in the "DELETE" mode. This mode is used when he/she accidentally encloses an incorrect video memory, or when he/she thinks that a certain video memory is not needed anymore. (First, a video memory is run out from an object. He/she then emits contexts to the real world. Lastly, the video memory passes out of the object.)
- **MOVE:** This mode is useful when the wearer wants to move a video memory from a certain object to another object. For example, the wearer encloses a video memory to a notebook in advance when he/she is on a business trip. He/she rearranges video memories to each appropriate object after he/she comes back to his/her office. (First, a video memory is run out from an object. He/she then emits contexts to the real world. Lastly, he/she runs in the video memory to another object.)
- **COPY:** In this mode the wearer can copy a video memory to other objects. An event often has contextual relations with multiple real-world objects. This mode enables the user to disperse a video memory to appropriate objects. (First, a video memory is run out from an object . The same video memory, however, remains in the object. He/she then emits context to the real world. Lastly, he/she runs in the video memory to another object.)

Operations Using a Jog Remote Controller Interface

A wearer is allowed two ways to use a jog remote controller: one way is to set permission for publication by referring video memories that are enclosed to an object, the other way is to seek for an appropriate video memory from retrieved candidates.

Selecting Publication/Reference Level:

A wearer must set a publication level attribute to a video memory to limit the people who can refer to the video memory when the user encloses it to an object. This publication attribute is particularly important when a wearer encloses a highly private video memory. Additionally, the wearer can select the reference level indicating the type of candidate video memories retrieved in a disclosure process. We have defined the following attributes:

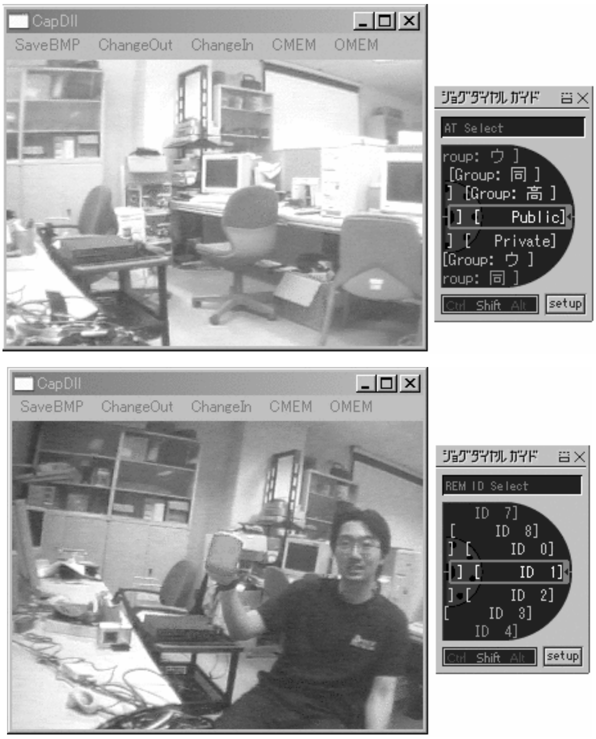
- **Publication Level:** A publication level is set when the wearer encloses a video memory to an object. Figure 26 (top) shows an example; he/she selects the attribute "Public." Three types of publication levels exist:
  - **Private:** Only the wearer who enclosed the video memory can disclose it.
  - **Group:** Wearers who belong to a certain group can disclose it.
  - **Public:** All users can disclose it.

- **Reference Level:** This level is selected when a wearer discloses a video memory from an object. In order to retrieve a desired video memory, he/she can reduce the video memories into disclosure candidates that are set at a certain publication level. Three types of reference levels exist:
    - **Personal:** The wearer can disclose his/her own video memories.
    - **Group:** The wearer can disclose his/her group's video memories published by the "Group."
    - **Public:** The wearer can disclose all video memories that he/she is allowed to refer to by permission of the owners.

Finding an Appropriate Video Memory:

In the disclosure process a wearer can easily find the desired video memory if the number of memory candidates were enclosed in the touched object using the jog controller. As the number of enclosed memories increases, however, it becomes more and more difficult for the wearer to find the memory to be disclosed among the candidates even if a wearer is limited by selecting one of the reference levels. Figure 26 (bottom) depicts how to find a video memory. In the example, the wearer views a snapshot of a video memory, which is activated by the controller, in the HMD. The wearer can change an activated snapshot when he/she turns the dial of the controller around. The wearer finally

Figure 26. Selecting publication level and the required video



discloses a video memory by pushing down the controller when he/she finds a video memory of interest.

Experiments and Results

We conducted an experiment to evaluate the effect of employing real world objects as media for augmenting human memory.

Methods

This experiment was conducted at the Nara Institute of Science and Technology (NAIST) in Nara, Japan, among graduate students of the Information Science Department. Twenty test subjects were included in this experiment.

For materials, we used 10 physical objects that had no contextual relation to each other. We also used 10 portraits of unfamiliar persons, and two sets of 10 playing cards composed of the numbers 1 through 10. We conducted the experiment under laboratory conditions. One experiment was composed of a memory test and a recall test. In the memory test, the subject memorized 10 trials. In the recall test, the subject answered a questionnaire.

In a trial of the memory test, the subject was first shown a pair consisting of an object and a portrait. The subject then selected one of the corners of the portrait. Finally, the subject was shown the predetermined pair of playing cards. The subject was allowed to look at these numbers for 30 seconds. The subject had to memorize the object and portrait pair, and the corner of the portrait and two card numbers as a real-world experience including narrative contexts. The subject continuously tried to memorize all trials. All subjects had to do two experiments within the following four conditions:

C1: Use only human memory (learn by heart)
C2: Use only facial characteristics (record with a paper and a pen)
C3: Refer to photo album type portraits that were used in the memory trial
C4: Use the *Ubiquitous Memories* module to refer to portraits in the recall test

Test subjects were divided into four groups. Group 1 did two experiments using conditions C1 through C3. Group 2 experimented using conditions C3 through C1. Group 3 experimented two times using conditions C2 through C4. Group4 did two experiments in conditions C4 through C2.

In the recall test, a questionnaire contained 10 recall questions. The subject was given one object image in each question. There were three empty boxes (portrait, corner, and card numbers) in a question. The subject then selected a portrait ID from a list having 40 portraits, marked a corner (Left-Top, Left-Bottom, Right-Top, Right-Bottom), and wrote down two card numbers. The subject was then given a list of 10 portraits used in the memory test only in condition C3. All subjects filled in some or all answers within 10 minutes. The question sequence was changed from the trial sequence in the memory test. All subjects ware allowed to answer the questions in a random order.

Table 7. Recall rate

	C1	C2	C3	C4
N**	24.0%	31.0%	10.0%	2.0%
PB'F'	11.0%	8.0%	19.0%	19.0%
P'BF'	12.0%	9.0%	5.0%	3.0%
P'B'F	23.0%	20.0%	32.0%	31.0%
PB'F	8.0%	4.0%	3.0%	1.0%
PB'F	4.0%	9.0%	11.0%	19.0%
P'BF	5.0%	1.0%	3.0%	0.0%
PBF	13.0%	18.0%	17.0%	25.0%
P**	51.0%	55.0%	79.0%	94.0%
B	53.0%	48.0%	57.0%	59.0%
F	30.0%	32.0%	34.0%	45.0%

Results

The results were taken of the 20 questionnaires collected from the graduate students of the Information Science Department in NAIST. Table 7 illustrates the recall rates from the 20 questionnaires. In this section we defined N, P, B, F and '. N (which is a percentage of errors), as follows: no answers regarding a portrait, a card and card numbers were correct answers on several questions. P shows that the answer regarding the portrait was correct. B shows that the answer of a corner was correct. F represents the answers of card numbers that were correct. X' (X is either P, B or F) represents the answer of a question X that was not correct.

In Table 7, N and P show a significant difference among the four test conditions ($p < 0.001$). P by C4 is, however, not 100% because of module error. In the sum of P'BF', P'B'F, and R'BF, we can see the influence on the difference among the test conditions (C1: 25.0%, C2: 14.0%, C3: 11.0%, C4: 4.0%, $p < 0.001$). The sum of PBF' and PBF shows the transparency in the different test conditions (C1: 36.0%, C2: 38.0%, C3: 49.0%, C4: 56.0%, $p > 0.1$).

Discussion

We need to investigate which kind of memory aid strategy performed best. Table 7 shows that our proposed module was the most effective. The differences are especially clear in the result of N and P. In the result of PBF (C1: 13.0%, C2: 18.0%, C3: 17.0%, C4: 25.0%), C4 showed the relationship of an object associated with a portrait. Additionally, the sum of P'BF', P'B'F, and R'BF (C1: 25.0%, C2: 14.0%, C3: 11.0%, C4: 4.0%, $p < 0.001$) shows a good result for the *Ubiquitous Memories* module. The sum of PBF' and PBF (C1: 36.0%, C2: 38.0%, C3: 49.0%, C4: 56.0%, $p > 0.1$), however, represents a lack of effectiveness for unrecorded events even if a person chose any of the memory-aid strategies.

In the experiment, the module showed the following two significant results:

1. People tend to use a module similar to conventional externalized memory-aid strategies, such as a memorandum (C2), and a photo album (C3).

2. The result shows that “Enclosure” and “Disclosure” operations, which enable wearers to directly record/refer to a video memory into/from an object, have enough effectiveness to make ubiquitous memories in the real world.

We believe that the module is more useful than conventional externalized memory-aid strategies. Increasing the workload adds knowledge about how best to conduct oneself in a certain situation or events in our increasing complicated lives. The former result means that a wearer can make ubiquitous memories without special overloads using our proposed module.

Summary

We introduced a *Ubiquitous Memories* module that enables wearers to make ubiquitous video memories using real-world objects. A prototype of the *Ubiquitous Memories* module has been developed on a wearable camera, with an RFID tag reader and anRFID tag attached to a real-world object. This module has enough basic operations to directly enclose/disclose a video memory into/from an object.

The shortcomings of this study include the necessity for retrieving a proper video memory from a huge collection of enclosed video memories in an object. An object selection problem also remains in both operations of “Enclosure” and “Disclosure.” In the case of object selection, a wearer might worry about selecting an object to enclose a video memory. In order to resolve these problems, we should structuralize contextual relations among persons, objects, and contexts (Michael, Hans, & Albrecht, 2001; Michael, Tobias, & Christian, 2002), and investigate the similarity of enclosure and disclosure patterns between similar objects and between non-related objects. Lastly, we believe that the “Memory Exchange” function will be an important issue in the near future because cooperative recall is superior to individual recall in the memory process (Takatori, 1980).

CONCLUSIONS

The research for augmented memory system has grown in recent years. However, most augmented memory researchers have dedicated themselves to resolving problems similar to previous problems using conventional techniques and have analyzed simplified human traits of memory activities. Recently, such research has been directed primarily toward clearing up issues related to wearable computing such as: investigating relationships between video data and brain wave data and supporting memory activity via a subliminal effect. Exchanging experiences is also a key topic in realizing augmented memory system. We believe that the study of augmented memory is an excellent test bed for integrating information spaces in such fields as psychology, sociology, and information science and technology into the real world.

The future research direction of the *SARA* framework includes connection-ability problems among super-distributed augmented memory modules. One of the most pressing problems for consideration is the ultimate realization of the operation of “*Association*” (surfing on the augmented memory). In order to accomplish “*Association,*” the augmented memory system would require standard notation rules to connect a huge amount of modules. Another difficult issue is the exchange of augmented memories among various people. In order to exchange augmented memories, the system has to convert information from strongly personalized augmented memories into a public domain for all desired users. Most studies for augmented memories, however, considered all human experience in a more or less homogeneous context. Each person has a unique perspective even if he/she shares an experience with another person at the same time and in the same place. We believe that exchanging useful augmented memories to realize information conversion requires contributions of relations or associations between multimedia data management strategies and human memory traits.

ACKNOWLEDGMENTS

This research is supported by Core Research for Evolutional Science and Technology (CREST) Program “Advanced Media Technology for Everyday Living” of Japan Science and Technology Agency (JST).

REFERENCES

Abowd, G.D., & Mynatt, E.D. (2002, March). The human experience. *IEEE Pervasive Computing*, 8-57.

Aizawa, K., Ishijima, K., & Shina, M. (2001). Automatic summarization of wearable video-indexing subjective interest. *Proceedings of the 2nd IEEE Pacific-Rim Conference on Multimedia, Springer LNCS2195*, 16-23.

Brewer, W.F. & Treyens, J.C. (1981). Role of schemata in memory for places. *Cognitive Psychology*, *13*, 207-230.

Clarkson, B., & Pentland, A. (2000). Framing through peripheral perception. *Proceedings of the 2000 IEEE International Conference on Image Processing*, 10-13.

Clarkson, B., & Pentland, A. (2001). Predicting daily behavior via wearable sensors. Technical Report Vismod TR451, Massachusetts Institute of Technology.

Czerwinski, M. & Horvitz, E. (2002). An investigation of memory for daily computing events. *Proceedings of the 16th Annual Human-Computer Interaction Conference*, 230-245.

DeVaul, R.W., Pentland, A., & Corey, V.R. (2003). The memory glasses: Subliminal vs. overt memory support with imperfect information. *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, 146-153.

Farringdon, J., & Oni, V. (2000). Visual augmented memory (VAM). *Proceedings of the 4th IEEE International Symposium on Wearable Computers*, 167-168.

Gelgon, M., & Tilhou, K. (2002). Automated multimedia diaries of mobile device users need summarization. *Proceedings of the 4th International Symposium on Human Computer Interaction with Mobile Devices and Services*, 36-44.

Healey, J., & Picard, R.W. (1998). StartleCam: A cybernetic wearable camera. *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, 42-49.

Hoisko, J. (2000). Context triggered visual episodic memory prosthesis. *Proceedings of the 4th IEEE International Symposium on Wearable Computers*, 185-186.

Horprasert, T., Harwood, D., & Davis, L.S. (1999). A statistical approach for real-time robust background subtraction and shadow detection. *Proceedings of the IEEE International Conference on Computer Vision FRAME-RATE Workshop*.

Jang, S., & Woo, W. (2003). Ubi-UCAM: A unified context-aware application model. *Proceedings of the 4th International and Interdisciplinary Conference on Modeling and Using Context*, 178-189.

Jebara, T., Schiele, B., Oliver, N., & Pentland, A. (1998). DyPERS: Dynamic personal enhanced reality system. MIT Media Laboratory, *Perceptual Computing Technical Report #463*.

Kato, T., Kurata, T., & Sakaue, K. (2002a). VizWear-Active: Towards a functionally-distributed architecture for real-time visual tracking and context-Aware UI. *Proceedings of the 6th IEEE International Symposium on Wearable Computers*, 162-163.

Kato, T., Kurata, T. , &Sakaue, K. (2002b). Face registration using wearable active vision system for augmented memory. *Proceedings of Digital Image Computing Techniques and Applications (DICTA2002)*, 252-257.

Kawamura, T., Fukuhara, T., Takeda, H., Kono, Y. , & Kidode, M. (2003). Ubiquitous Memories: Wearable interface for computational augmentation of human memory based on real-world objects. *Proceedings of the 4th International Conference on Cognitive Science*, 273-278.

Kawamura, T., Kono, Y. , & Kidode, M. (2001). A novel video retrieval method to support a user's recollection of past events aiming for wearable information playing. *Proceedings of the 2nd IEEE Pacific-Rim Conference on Multimedia*, 24-32.

Kawamura, T., Kono, Y., & Kidode, M. (2002). Wearable interfaces for a video diary: Towards memory retrieval, exchange, and transportation. *Proceedings of the 6th IEEE International Symposium on Wearable Computers*, 31-38.

Kawamura, T., Kono, Y., & Kidode, M. (2003). Nice2CU: Managing a person's augmented memory. *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, 100-101.

Kawamura, T., Ueoka, T., Kiuchi, Y., Kono, Y., & Kidode, M. (2003). Image similarity measurement by an integrated probabilistic histogram. *Technical Report of NAIST Information Science #NAIST-IS-TR2003015*.

Kawamura, T., Ukita, N., Kono, Y., & Kidode, M. (2002). HySIM: A hybrid-space image matching method for a high speed location-based video retrieval on a wearable computer. *Proceedings of the 2002 IAPR Workshop on Machine Vision Applications*, 94-97.

Kawashima, T., Nagasaki, T., & Toda, M. (2002). Information summary mechanism for episode recording to support human activity. *Proceedings of the International Workshop on Pattern Recognition and Understanding for Visual Information Media*, 49-56.

Kidd, C.D., Orr, R., Abowd, G..D., Atkeson, C.G., Essa, I.A., MacIntyre, B., Mynatt, E.D., Starner, T.E., & Newstetter, W. (1999). The aware home: A living laboratory for ubiquitous computing research. *Proceedings of the 2nd International Workshop on Cooperative Buildings*, Position Paper.

Kidode, M. (2002). Design and implementation of wearable information playing station. *Proceedings of the 1st CREST Workshop on Advanced Computing and Communicating Techniques, Techniques for Wearable Information Playing*, 1-5.

Kono, Y., Kawamura, T., Ueoka, T., Murata, S., &Kidode, M. (2003). SARA: A framework for augmented memory albuming systems. *Proceedings of the 2nd CREST Work-*

shop on Advanced Computing and Communicating Techniques for Wearable Information Playing, 20-34.

Lamming, M., & Flynn, M. (1994). Forget-me-not: Intimate computing in support of human memory. *IN FRIEND21: International Symposium on Next Generation Human Interface*, 125-128.

Leibe, B. , & Schiele, B. (2003a). Analyzing appearance and contour based methods for object categorization. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Leibe, B., & Schiele, B. (2003b). Interleaved object categorization and segmentation, *Proceedings of the British Machine Vision Conference*.

Maeda, H., Hirata, T., & Nishida, T. (1997). CoMeMo: Constructing and sharing everyday memory. *Proceedings of the 9th International Conference on Tools with Artificial Intelligence*, 23-30.

Mann, S. (1997). Wearable computing: A first step toward personal imaging. *Computer*, *30*(2), 25-31.

Michael, B., Hans, W.G.., & Albrecht, S. (2001). MediaCups: Experience with design and use of computer augmented everyday artefacts. *Computer Networks*, *35*(4), 401-409.

Michael, B., Tobias, Z, & Christian, D. (2002). A location model for communicating and processing of context. *Personal and Ubiquitous Computing*, *6*(5-6), 341-357.

Murakami, H. (2002). Information acquisition and reorganization from the WWW by using memory-organizer. *Bulletin of Osaka City University Media Center*, *3*, 9-14.

Nakamura, Y., Ohde, J., & Ohta, T. (2000). Structuring personal experiences – Analyzing views from a head-mounted camera. *Proceedings of the IEEE International Conference on Multimedia and Expo 2000*.

Nayar, S.K., Nene, S.A., & Murase, H. (1996). Real-time 100 object recognition system, *Proceedings of the ARPA Image understanding Workshop*, 22-28.

Nickerson, R.S., & Adams, M.J. (1979). Long-term memory for a common object. *Cognitive Psychology*, *11*, 283-307.

Picard, R.W., & Healey, J. (1997). Affective wearables. *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, 90-97.

Rekimoto, J., & Ayatsuka, Y. (2000). CyberCode: Designing augmented reality environments with visual tags. *Proceedings of Designing Augmented Reality Environment*.

Rekimoto, J., Ayatsuka, Y., & Hayashi, K. (1998). Augment-able reality: Situated communication through physical and digital Spaces. *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, 68-75.

Rhodes, B. (1995). Augmented memory. Online: *http://www.media.mit.edu/wearables /lizzy/augmented-memory.html*

Rhodes, B. (1997). The wearable remembrance agent: A system for augmented memory. *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, 123-128.

Rhodes, B. (2003). Using physical context for just-in-time information retrieval. *IEEE Transactions on Computers*, *52*(8), 1011-1014.

Schiele, B., & Crowely, J.L. (1996). Probabilistic object recognition using multidimensional receptive field histograms. *Proceedings of the 13th Conference on Pattern Recognition*, Vol. B, 50-54.

Schiele, B., Oliver, N., Jebara, T., & Pentland, A. (1999). An interactive computer vision system DyPERS: Dynamic Personal Enhanced Reality System. *Proceedings of International Conference on Vision Systems*, 51-65.

Shinnishi, M., Iga, S., & Higuchi, F. (1999). Hide and seek: Physical real artifacts which respond to the user. *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, 4, 84-88.

Sumi, Y., Sakamoto, R., Nakao, K., & Mase, K. (2002). Comic diary: Representing individual experiences in a comic style. *Proceedings of the 4th Annual Conference on Ubiquitous Computing*, 16-32.

Takatori, K. (1980). The role of communication in the memory process – A comparative study of individual recall and cooperative recall. *Educational Psychology, XXVIII*(2), 108-113 (in Japanese).

Toda, M., Nagasaki, T., Iijima, T., & Kawashima, T. (2003). Structural representation of personal events. *Proceedings of the ISPRS International Workshop on Visualization and Animation of Reality-based 3D Models*.

Tran, Q.T. , & Mynatt, E.D. (2003). What was I cooking? Towards deja vu displays of everyday memory. *The Technical Report* #GIT-GVU-03-33.

Tulving, E., & Thomson, D.M. (1973). Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, *80*, 352-373.

Ueda, T., Amagasa, T., Yoshikawa, M., & Uemura, S. (2002). A system for retrieval and digest creation of video data based on geographic objects. *Proceedings of the 13th International Conference on Database and Expert Systems Applications*, 768-778.

Ueoka, R., Hirota, K., & Hirose, M. (2001). Wearable computer for experience recording. *Proceedings of the 11th International Conference on Artificial Reality and Telexistence*.

Ueoka, T., Kawamura, T., Kono, Y., & Kidode, M. (2003). I'm Here!: A wearable object remembrance support system. *Proceedings of the 5th International Symposium on Human Computer Interaction with Mobile Devices and Services*, 422-427.

Ukita, N., Terabe, A., Kono, Y., & Kidode, M. (2002). Wearable virtual tablet: Fingertip drawing interface using an active-infrared camera. *Proceedings of IAPR Workshop on Machine Vision Applications*, 98-101.

Weiser, M. (1991, September). The computer for the 21st century. *Scientific American*, *265*(3), 94-104.

Chapter IV

Adaptive Summarization of Digital Video Data

Waleed E. Farag, Zagazig University, Egypt

Hussein Abdel-Wahab, Old Dominion University, USA

ABSTRACT

As multimedia applications are rapidly spreading at an ever-increasing rate, efficient and effective methodologies for organizing and manipulating these data become a necessity. One of the basic problems with such systems is finding efficient ways to summarize the huge amount of data involved. In this chapter, we start by defining the problem of key frames extraction, then review a number of proposed techniques to accomplish that task, showing their pros and cons. After that, we describe two adaptive algorithms proposed to effectively select key frames from segmented video shots where both apply a two-level adaptation mechanism. These algorithms constitute the second stage of a Video Content-based Retrieval (VCR) system that has been designed at Old Dominion University. The first adaptation level is based on the size of the input video file, while the second level is performed on a shot-by-shot basis to account for the fact that different shots have different levels of activity. Experimental results show the efficiency and robustness of the proposed algorithms in selecting the near-optimal set of key frames required to represent each shot.

INTRODUCTION

As multimedia applications are rapidly spreading at an ever-increasing rate, novel techniques for organizing and abstracting the data they produce become a necessity. One of the basic problems with such systems is finding efficient ways to summarize the huge amount of data involved. To solve this problem, digital video streams need to be analyzed by first dividing each stream into a set of meaningful and manageable units — a task that was the focus of Video Shot Boundary Detection chapter. After dividing a video sequence into a number of shots, each shot still contains a large number of frames. As a result, the second stage in any video analysis system is the process of Key Frames (KFs) selection (Rui, Huang, & Mehrotra, 1998) that aims to abstract the whole shot using one frame or more. This process is known also as video summarization. Ideally, we need to select the minimal set of KFs that can faithfully represent each shot. KFs are the most important frames in a shot since they may be used to represent the shot in the browsing system as well as be used as access points. Moreover, one advantage of representing each shot by a set of frames is the reduction in the computation burden required by any content analysis system to perform similarity matching on a frame-by-frame basis.

In the segmentation chapter, we have explained how an input MPEG stream is processed by first extracting the DC sequence (Yeo & Liu, 1995a) from it. Subsequently, that sequence is used as an input to a neural network module to perform the segmentation, shot boundary detection, task. The output of the segmentation stage is a group of distinct shots with clear marks to the beginning and end of each segment. This output is then fed to our KFs selection module (Farag & Abdel-Wahab, 2002a, 2002b), which is the focus of this chapter.

We start by introducing a fairly simple and efficient algorithm that uses the accumulated summation of DC frames luminance differences in order to select KFs from segmented video shots. Once the accumulated summation exceeds a certain threshold, the current frame is added to the representative set in addition to the first frame (which is chosen by default), then the algorithm proceeds. To improve the basic algorithm introduced above, we propose a first-level adaptation mechanism that is based upon the dimension of the input video file. After that, a second level of adaptation based on a shot activity criterion is introduced to further improve the performance and accuracy of the selection module. At the end, another algorithm is proposed that uses a direct comparison between the summation of the luminance channel DC terms of the current frame and the corresponding summation of the last chosen key frame. The algorithm then selects the current frame into the representative set if the absolute difference is above a certain threshold. This algorithm uses a first-level adaptation policy similar to that used by the first group, but it employs a statistical criterion for the shot-by-shot adaptation level. Analyzing the results produced by both algorithms showed their efficiency and accuracy in selecting the near- optimal set of key frames required to represent each shot in a video stream. We then conclude the chapter by comparing the performance of both algorithms.

This chapter is organized as follows. In the next section, we review briefly a number of related approaches for selecting KFs. The first proposed set of algorithms is then introduced, starting with the most straightforward one and going towards the more effective ones. After that, we introduce the second proposed algorithm along with its criteria for the adaptation processes. Performance comparisons of the two proposed mechanisms are given next, followed by the Conclusion at the end of the chapter.

RELATED WORK

Key frames extraction is one of the active areas of research in visual information retrieval (Bimbo, 1999; Lew, 2001). A review of the major approaches that have been proposed by different researchers in the field to tackle this problem is given below.

A shot boundary-based approach was proposed in Nagasaka and Tanaka (1991) that uses the *nth* frame in each shot as the key frame. The main disadvantage to this method is that it uses only one key frame that may not be stable and may not capture the major visual content of the shot. Zhang, Kankanhalli, Smoliar, and Tan (1993) proposed a visual content-based approach where the first frame is used as a key frame, but in addition to it other frames could be selected as additional key frames if they have significant content change compared to the first one. Motion-based criteria are also considered in this method. Another approach that is based on motion analysis was proposed in Wolf (1996). That technique calculates the optical flow (Singh, 1991) for each frame, then computes a motion metric. It finally analyzes that metric and selects KFs at the local minima of motion.

A clustering algorithm has been proposed in Zhuang, Rui, Huang, and Mehrotra (1998). The algorithm assumes that there are N frames within a shot divided into M clusters. The similarity between frames is performed using color histograms, and the algorithm works as follows:

- The first frame is considered as the centriod of the first cluster, then subsequent frames are compared with existing clusters centriods (if any).
- If the maximum similarity value is less than a threshold (*d*), the current frame is put into a new cluster, otherwise it is put into the cluster with the maximum similarity.
- Finally, the algorithm adjusts the clusters centriods.
- After the process of clusters formation described above, the algorithm chooses KFs from representative clusters (those having the number of frames greater than N/M). The frame that is closest to the cluster centriod is chosen as a key frame.

In Yeung and Liu (1995), temporal sampling — the selection of representative frames — is achieved by using a nonlinear sampling process in which every frame is compared with the last chosen representative frame. If the difference is above a certain threshold, that frame is added to the set of representative frames for that particular shot. An alternative method is given in Yeo and Liu (1995b), where a set of KFs is used to represent a shot and the first frame in that shot is always selected as a KF. A different approach to represent the shot is proposed in Chen, Taskiran, Albiol, Delp, and Bouman (1999), where frames in the shot are organized in a tree structure in such a way that the root node is the most representative frame in the shot. As one progresses down the tree, frames are organized into representative groups. This tree representation is obtained through agglomerative (bottom-up) clustering, where color, texture, and edge histograms are the components of the feature vector and L1 norm is used to measure the feature distance.

Tonomura, Akutsu, Otsuji, and Sadakata (1993) proposed a system that represents video sequences by using evenly spaced key frames while ignoring shot boundaries. The major problem with this system is that it selects more than the necessary number of key frames especially in long inactive shots. In Girgensohn and Boreczky (1999), a technique is introduced to detect key frames to represent the whole video without also doing any shot boundary detection. The general idea is to use a clustering algorithm to divide the frames into a number of clusters, each with similar frames, then choose a frame from each cluster. The algorithm works as follows:

- At first, candidate frames are selected then the clusters are formed.
- Clusters that have small numbers of frames are filtered out (by using a time constraint condition).
- Then another time constraint is used to guide the process of selecting key frames from clusters.
- At the end, some classes of images are emphasized as key frames, for example close-ups on people are preferred over long shots.

An illumination invariant approach is proposed in Drew and Au (2000) to select key frames. The technique starts with an off-line processing stage that uses a training video set. It then normalizes the color channels, calculates the spherical chromaticity, applies the wavelet compression to the histogram produced by the previous step, and then uses the DCT transform to produce 21 DC coefficients. Finally, it employs singular-value decomposition to produce 12 basis vectors. For any new video, the following online processing steps are performed:

- Each frame is processed as described in the off-line processing stage to produce 21 DCT coefficients that are used with the 12 basis vectors to provide 12-coefficient features vectors for that frame.
- Hierarchical clustering is then followed.
- Frames closest to clusters centriods are selected as key frames.

A hierarchical color and motion segmentation scheme that is based on a multi-resolution implementation of the recursive shortest spanning tree is proposed in Avrithis, Doulamis, Doulamis, and Kollias (1999). All segment features produced by that hierarchical segmentation are then collected together using a fussy multidimensional histogram to reduce putting similar segments into different classes. Afterwards, the extraction of key frames is performed for each shot in a content-based rate-sampling approach.

ACCUMULATED FRAMES SUMMATION ALGORITHMS

The produced shots from the shot boundary detection module designed in the segmentation chapter need to be summarized by extracting KFs from them. Although a number of KFs extraction techniques were proposed in the literature, as discussed in the previous section, they have the following shortcomings:

- Some of the proposed methodologies for KFs extraction use only one frame to represent a shot. In many cases, one frame is not sufficient to semantically represent the shot, especially if the shot is complex or contains a lot of motion.
- A different group employs algorithms that are oversimplified so that they cannot adapt to different changing situations; for instance, various input frame sizes and various activity levels within shots.
- Other techniques use mechanisms that calculate the optical flow or other complex models to select KFs. Although they may give more accurate results than simpler approaches, they are computationally expensive that renders them unsuitable for online processing.

To avoid these shortcomings, we propose mechanisms that attempt to take a balanced approach between the two extremes. They are not oversimplified like some of the surveyed techniques, yet they attempt to be more accurate as well as computationally efficient. As mentioned before, the use of one key frame is, in general, not sufficient to represent a shot unless that shot is a completely still one. As a result of this fact, our proposed approaches use a set of frames to represent each shot. This set may contain only one frame if that frame is capable of representing the salient characteristics of the shot.

In this section, we present a group of algorithms to select KFs that uses the Accumulated Frames Summation (AFS) of DC terms. We start by explaining the first version of the proposed algorithm, illustrating its performance and commenting on its shortcomings. Subsequently, two enhancements to the basic algorithm are introduced along with the experimental results obtained by implementing each enhancement. All the proposed algorithms in this section and in the next one work on the DC terms of the luminance channel of DC frames extracted by the shot boundary detection module.

Using Accumulated Summation Without any Adaptation

The first developed algorithm to select KFs could be described as follows:

- *Initialize the representative set to be empty and select the first frame as the first element in that set.*
- *Initialize sum = 0 and flashIndex to the last detected flash position plus one.*
- *For j = 0 To N-2 do {*

```
if ( j = flashArray[flashIndex] )    /* only if ((j-1) >= 0) */
          sum = sum + diffArray[j-1]
else if ( j = (flashArray[flashIndex]+1) )
                sum = sum + diffArray[j-2]
            flashIndex = flashIndex+1
else
          sum = sum + diffArray[j]
if ( sum > $\delta_i$ )
          sum = 0
```

```
        Add j to the set of selected KFs
        Increment the number of selected KFs
    }
```

Where

$$diffArray[j] = \sum_{i=0}^{M-1} \left| DC_i(j) - DC_i(j+1) \right| \qquad \textbf{(1)}$$

N: The number of frames in a shot.
flashArray[]: An array containing indexes to flashlight positions.
δ_i: An initial threshold that determines the frequency of sampling.
DC_i: The DC of the luminance channel at location i.
M: The number of DC terms in a DC frame.

It is important to note that the occurrences of flashlights within shots are detected by the shot segmentation algorithm, and those points are passed to the key frames selection module into *flashArray[]*. The KFs selection module employs this information to avoid the inclusion of these pseudo peaks (see the Segmentation chapter) into the calculation of the accumulated summation used by the above algorithm. Instead, the algorithm substitutes each of these high-valued peaks by the last considered frame difference value (if any) as an approximation. In that way, the algorithm robustly excludes the effect of lighting changes that occurred as results of flashlights.

In order to evaluate the performance of the above algorithm, a number of video files were used during the tests. These files were segmented as previously described. Moreover, in our choice of these clips, we attempted to select various types of contents, amount of activities, and sizes of the captured video that would prove the applicability of the proposed algorithms over a range of various circumstances. The next important issue that needs to be addressed for the above algorithm to be effective is the choice of the threshold value (δ_i) that gives the best results for all these clips. While choosing the appropriate value of the threshold and in evaluating the proposed algorithms, we depend upon human opinion in comparing the selected KFs with those that should be generated (ground truth). To achieve these goals, we start by trying various values of the threshold with almost all of those clips and investigate which threshold value gives the best results in terms of the minimal expressive set of key frames. Table 1 and Table 2 show the numbers of selected KFs, their averages per shot, and the total average as a function of threshold value for those 11 clips.

The relation between threshold value and the percentage of selected key frames to the total number of frames is plotted in Figure 1 for three clips. The general and intuitive trend is the decrease in the number of selected key frames with the increase of the threshold values. This behavior is obvious for all three curves in Figure 1 and also can be observed by investigating Table 1 and Table 2. Our objective is to select the minimal number of key frames that is able to accurately describe and summarize the whole video stream. Selecting a very large threshold will certainly reduce the number of selected KFs but, at the same time, will discard many necessary frames that are supposed to be selected in order to properly summarize the video data. On the other hand, small threshold values

Table 1. Number of selected KFs as a function of threshold values (25000<=δ_i<=100000)

Video name	**δ_i = 25000**		**δ_i = 50000**		**δ_i = 100000**	
	# of KFs	**Ave/shot**	**# of KFs**	**Ave/ shot**	**# of KFs**	**Ave/ shot**
soccer	56	18.7	31	10.3	18	6.0
racing-boats	37	6.2	22	3.7	14	2.3
action-movie	20	6.7	11	3.7	6	2.0
carton	50	5.6	29	3.2	18	2.0
celebration	42	21.0	23	11.5	15	7.5
comedy	46	9.2	25	5.0	13	2.6
ads	79	7.2	44	4.0	24	2.2
class	27	3.9	14	2.0	10	1.4
news-cast	71	3.1	43	1.9	27	1.2
conf-discussion	93	4.7	52	2.6	31	1.6
tv-show	167	2.3	107	1.5	81	1.1
Total Average		**8.03**		**4.48**		**2.72**

Table 2. Number of selected KFs as a function of threshold values (150000<=δ_i<=250000)

Video name	**δ_i = 150000**		**δ_i = 200000**		**δ_i = 250000**	
	# of KFs	**Ave / shot**	**# of KFs**	**Ave / shot**	**# of KFs**	**Ave / shot**
soccer	13	4.3	10	3.3	8	2.7
racing-boats	9	1.5	8	1.3	8	1.3
action-movie	5	1.7	4	1.3	3	1.0
carton	15	1.7	12	1.3	11	1.2
celebration	8	4.0	7	3.5	6	3.0
comedy	10	2.0	8	1.6	7	1.4
ads	19	1.7	15	1.4	14	1.3
class	8	1.1	8	1.1	7	1.0
news-cast	25	1.1	24	1.0	23	1.0
conf-discussion	26	1.3	24	1.2	23	1.2
tv-show	75	1.0	74	1.0	73	1.0
Total Average		**1.95**		**1.65**		**1.64**

will produce many redundant key frames and render the similarity matching operation very inefficient. Bearing in mind that trade-off, we attempt to select a moderate value of the threshold that gives us the best results in terms of representation accuracy and the number of selected KFs. The value that balances this compromise is 150,000, and for that reason we will use this value in any further experimentation.

After selecting the value of the threshold that gives the best results, let us study in detail the behavior of the algorithm when applied to a representative subset of the clips under investigation. KFs selection results for the action-movie clip are shown in Table 3. Shots 0 and 1 are more active than shot 2 (Figure 2), and thus the algorithm chooses more KFs to represent these shots, although the last shot has more frames than either of the other two. The basic factor to determine how many KFs are needed to abstract a shot is the amount of activity in that shot, but the shot length is another factor induced by the way the algorithm functions. It is important to note that if the algorithm chooses three KFs for shot 0, they would abstract that shot better than only two frames, as shown in Figure 3.

Figure 1. Percentage of selected KFs as a function of the threshold value

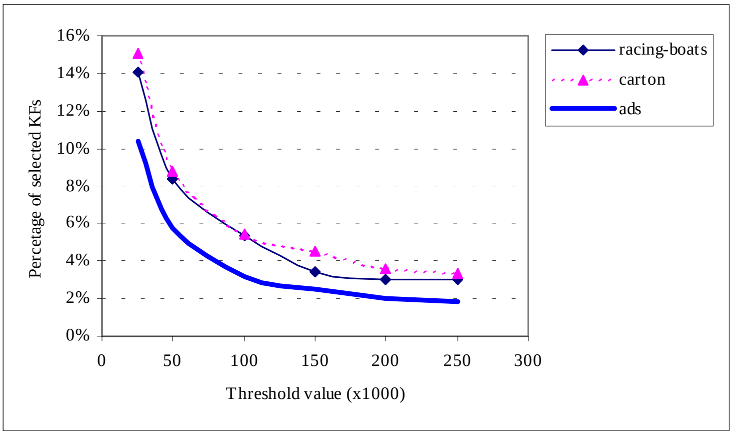


Table 3. KFs selection for the action-movie clip using δ_i =150,000 without any adaptation

Shot index	Range	# of frames	# of KFs	Index of selected KFs
0	0-51	52	2	0, 40
1	52-104	53	2	52, 96
2	105-177	73	1	105

For the carton clip, the moderate activity of Shots 1 and 2 in Table 4 allows the algorithm to select two KFs for each one of them. Shot 3 has a considerable amount of activity that can be noticed by investigating Figure 4, but the algorithm chooses only one KF to represent it. One more remark — the algorithm chooses the largest number of KFs for Shot 4, the most active shot in this clip, but regardless of that fact, there is still a need for more KFs to properly represent that shot due to its high activity.

Generally in TABLE 5, the result for the comedy clip whose frame difference graph is given in Figure 5. One KF is selected every about 115 frames for low-activity shots like Shot 1 (as in the case between 49 and 163). This can change due to a local increase in the activity and reach about 60 frames (163-226) because of the entrance of a new character into the scene (see the second picture in Figure 6). Ideally, three KFs are needed to properly describe Shot 1: one should contains the two characters in the scene, another KF should reflect the entrance of a third character, and the last one should feature the presence of only one character at the end of that shot (see Figure 6). The algorithm chooses 4 KFs for Shot 1 as a result of its length. In Shot 3, the selection of Frame 581 as a KF in addition to Frame 501 is justified by the entrance of a new character into the scene. The low-activity nature of Shots 0 and 4 can be noticed in Figure 5, thus the selection of one KF is proper. The choice for Shot 2 is also appropriate.

A close investigation to the algorithm's results for the class clip listed in Table 6 shows that a lot of needed KFs were not selected. This problem is the result of using a large threshold value with respect to the size of this MPEG clip. For instance, Shot 0 has

Figure 2. Frame difference graph for the action-movie video

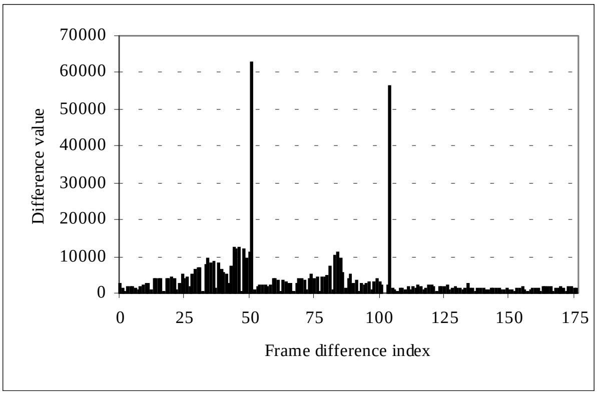


Figure 3. Three key frames better abstract the first shot of the movie clip than the use of two KFs because of the various positions the character takes

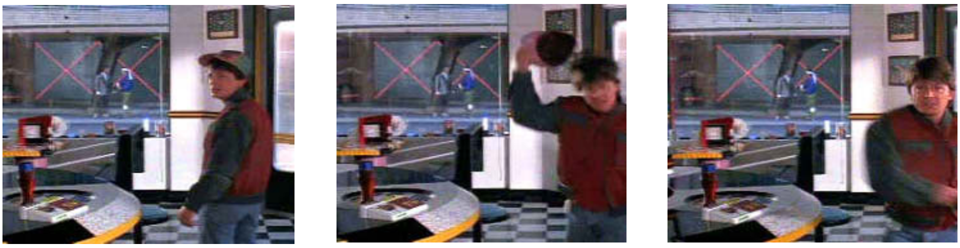


Table 4. KFs selection for the carton clip using δ_i =150,000 without any adaptation

Shot index	Range	# of frames	# of KFs	Index of selected KFs
0	0-48	49	1	0
1	49-83	35	2	49, 80
2	84-133	50	2	84, 127
3	134-151	18	1	134
4	152-234	83	5	152, 164, 182, 203, 220
5	235-244	10	1	235
6	245-254	10	1	245
7	255-266	12	1	255
8	267-330	64	1	267

camera panning at its end that necessitates the use of two to three KFs to faithfully represent the shot; this need can be noticed by investigating the clip frames difference in Figure 7. Reducing the value of the threshold is not a solution because this will affect the number of chosen key frames for large-dimension clips.

We can observe the main problem with the above-described algorithm through a study of its results, especially for small-dimension clips (176x120 and less). The source of the problem is the use of a fixed threshold value for all the clip sizes. Our criterion to

Figure 4. Frame difference graph for the carton video

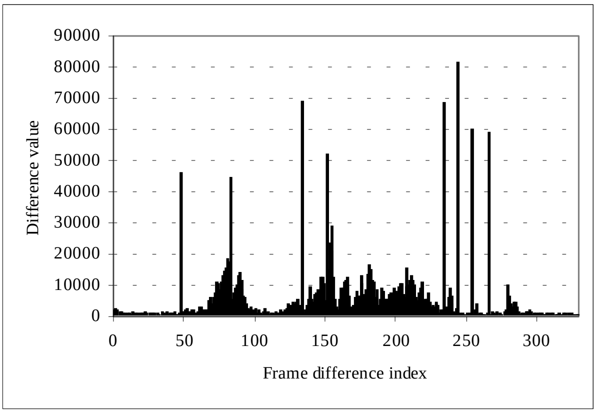
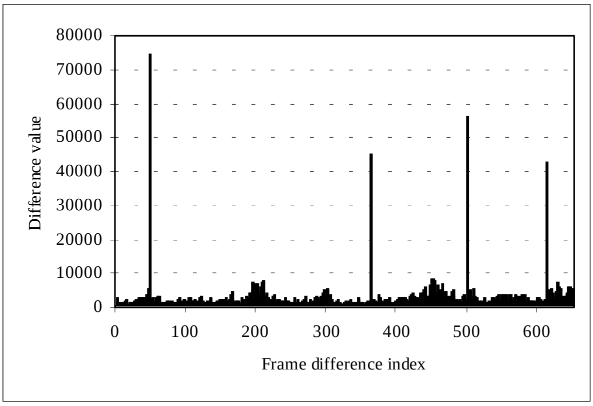


Table 5. KFs selection for the comedy clip using δ_i =150,000 without any adaptation

Shot index	Range	# of frames	# of KFs	Index of selected KFs
0	0-48	49	1	0
1	49-364	316	4	49, 163, 226, 331
2	365-500	136	2	365, 448
3	501-614	114	2	501, 581
4	615-654	40	1	615

Figure 5. Frame difference graph for the comedy video



determine if a new frame can be added to the representative set is the accumulated frames differences, and this value is a function of the video dimension. The chosen threshold works well for large-dimension clips (320x240 and larger), but it performs badly for small ones. In a general-purpose video retrieval database, we cannot assume any control over the dimension of input video streams. To be general and flexible enough, any new stream should be accepted, and the chosen threshold has to be adapted according to the input

Figure 6. Three key frames are ideally needed to abstract Shot 1 in the comedy clip

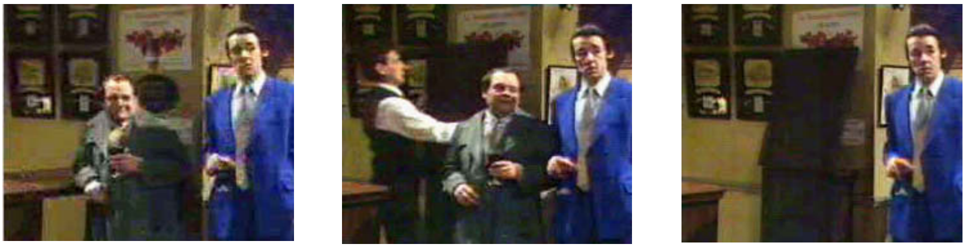
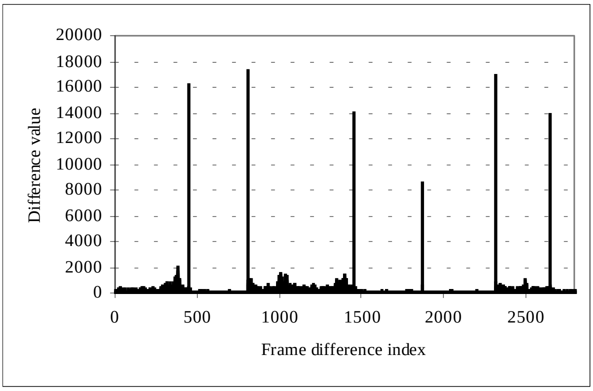


Table 6. KFs selection for the class clip using δ_i =150,000 without any adaptation

Shot index	Range	# of frames	# of KFs	Index of selected KFs
0	0-442	443	1	0
1	443-804	362	1	443
2	805-1452	648	2	805, 1214
3	1453-1866	414	1	1453
4	1867-2306	440	1	1867
5	2307-2641	335	1	2307
6	2642-2792	151	1	2642

Figure 7. Frame difference graph for the class video



video dimension. This enhancement to the basic algorithm presented above is the topic of the next section.

Using the First Level of Threshold Adaptation

As was shown in the previous section, there is a shortcoming in the proposed algorithm that can be stated as follows — using a single threshold for all input video files is not appropriate in all cases. The selection algorithm depends upon the accumulated differences between DC frames, and the difference itself is function of the size of the DC frame. So different sizes need different thresholds in order to reliably select the most appropriate and minimal set of representative frames. This problem motivates us to

Table 7. Number of selected KFs as a function of threshold values with first-level adaptation

Video name	δ_i = 50000		δ_i = 150000		δ_i = 250000		δ_i = 350000	
	# of KFs	Ave / shot	# of KFs	Ave / shot	# of KFs	Ave / shot	# of KFs	Ave / shot
soccer	31	10.3	13	4.3	8	2.7	7	2.3
racing-boats	29	4.8	14	2.3	8	1.3	8	1.3
action-movie	13	4.3	5	1.7	4	1.3	3	1.0
carton	40	4.4	18	2.0	14	1.6	11	1.2
celebration	29	14.5	11	5.5	7	3.5	5	2.5
comedy	29	5.8	11	2.2	8	1.6	6	1.2
ads	51	4.6	22	2.0	15	1.4	12	1.1
class	61	8.7	24	3.4	14	2.0	12	1.7
news-cast	148	6.4	63	2.7	42	1.8	32	1.4
conf-discussion	193	9.7	71	3.6	47	2.4	38	1.9
tv-show	353	4.8	142	1.9	106	1.5	92	1.3
Total Average		**7.14**		**2.88**		**1.91**		**1.54**

propose an enhancement to the basic algorithm in which the threshold used to determine the selection frequency (how many KFs are selected) becomes a function of the input video frame size. The algorithm gets the size information from the segmentation module and uses it to adapt the chosen threshold so that a different threshold is used for different frame sizes.

The proposed enhanced algorithm chooses the new threshold using a linear function of the size of the input file. This enhancement was implemented and a similar set of experiments was performed to judge the effectiveness of that approach. As mentioned in the previous section, threshold choice is the first step and hence we list, in Table 7, the overall results in terms of the total number of selected KFs. Again a threshold value equal to 150,000 seems to be a good choice, so we use that value as an initial threshold value (δ_i). Then, the algorithm chooses an adapted threshold value, δ_{a1}, for each clip in accordance with its dimension.

One important observation that comes from comparing Table 7 with Table 1 and Table 2 is that the total averages of the number of selected key frames per shot (taken over all the 11 clips) are larger in Table 7 than their corresponding values in the other two tables. The above observation is proof of the success of the proposed first-level adaptation algorithm in selecting more key frames (especially in the case of small-dimension clips) to better represent the video data. To illustrate the outcome of applying this modification, a set of tables similar to those given in the last section is presented in this section reporting the selection results for each individual clip.

Table 8. KFs selection for the action-movie clip using δ_{a1} =125,000 with first-level adaptation

Shot index	Range	# of frames	# of KFs	Index of selected KFs
0	0-51	52	2	0, 36
1	52-104	53	2	52, 86
2	105-177	73	1	105

Table 9. KFs selection for the carton clip using δ_{a1} =100,757 with first-level adaptation

Shot index	Range	# of frames	# of KFs	Index of selected KFs
0	0-48	49	1	0
1	49-83	35	2	49, 76
2	84-133	50	2	84, 97
3	134-151	18	2	134, 149
4	152-234	83	7	152, 156, 172, 182, 197, 209, 222
5	235-244	10	1	235
6	245-254	10	1	245
7	255-266	12	1	255
8	267-330	64	1	267

Table 10. KFs selection for the comedy clip using δ_{a1} =125,000 with first-level adaptation

Shot index	Range	# of frames	# of KFs	Index of selected KFs
0	0-48	49	1	0
1	49-364	316	4	49, 146, 206, 285
2	365-500	136	3	365, 441, 482
3	501-614	114	2	501, 570
4	615-654	40	1	615

The size of the soccer clip is taken as a reference size in which all other adapted thresholds are calculated with respect to it. For Table 8 through Table 10, the algorithm detects the relatively small size of these clips and hence reduces the threshold value, which leads to a general increase in the number of selected KFs (compared with corresponding numbers in the last section). This is true particularly for active shots such as Shot 4 in Table 9.

KFs selection results for the class clip are given in Table 11 where the total number of selected KFs is 24, compared to only 8 in TABLE 6. This confirms the effectiveness of the proposed first-level adaptation algorithm which avoids skipping required KFs for small size clips like this one. More comments are worth noting. Shot 0 has some camera work that calls for more than one KF, but the length of that shot is the main factor that causes the choice of five KFs. It is obvious from the frame difference diagram, Figure 7, that there is some activity in Shot 2 that requires more than one KF, but again the length of that shot biases the algorithm to choose many key frames (nine) for such a medium-activity shot. The same analysis applies to Shots 3, 4, and 5. Shots 1 and 6 are completely still so the choice of one KF for each is the right one, and the algorithm succeeds in making this choice, although Shot 1 is a relatively long one.

In the last section, many of the necessary KFs have been skipped, especially for small-dimension clips, but this is not the case when applying the first-level adaptation algorithm. In a nutshell, the use of the first-level adaptation mechanism solves the problem of the input size, but there are other problems that are still unsolved. The topic of the next section is to address the problems with the first-level adaptation algorithm and propose an effective and efficient strategy to overcome these shortcomings.

Table 11. KFs selection for the class clip using δ_{a1} =29,166 with first-level adaptation

Shot index	Range	# of frames	# of KFs	Index of selected KFs
0	0-442	443	5	0, 171, 305, 364, 388
1	443-804	362	1	443
2	805-1452	648	9	805, 870, 990, 1028, 1093, 1204, 1325, 1384, 1439
3	1453-1866	414	2	1453, 1814
4	1867-2306	440	2	1867, 2280
5	2307-2641	335	4	2307, 2399, 2494, 2622
6	2642-2792	151	1	2642

Using the Second Level of Threshold Adaptation

As discussed in the previous section, although the use of one-level adaptation solves the problem with various input file sizes, the algorithm still has two main problems:

- It tends to choose more key frames in cases of shots that have a large number of frames, even if they are inactive.
- It sometimes fails to select the required number of KFs in order to faithfully represent very active shots.

To tackle the above problems, we propose a second enhancement to the algorithm introduced in the last section by introducing a second level of adaptation. This second level is a shot-by-shot adaptation strategy that changes each shot threshold on the fly. That is to say, each shot in a certain video stream will get a threshold value in proportion to its activity level. So active shots get lower threshold values in order to increase the selection pressure, thus choosing more KFs to represent them. On the other side, low-activity shots get higher threshold values to reduce the sampling rate and prevent selecting too many redundant key frames. We first measure the shot activity before invoking the selection algorithm, then an adaptation heuristic is used to adapt each shot threshold based on the amount of activities found in that shot. As will be illustrated, the results of applying this technique are very good, and it solves the problems with the use of only one level of adaptation.

Again threshold selection is the first step and some of the results are listed in Table 12. We select δ_i = 150,000 and the algorithm is allowed to adapt it to a value called δ_{a2}.

To illustrate the effectiveness of the two-level adaptation mechanism, we start by discussing the differences between Table 12 and Table 7. For the first four clips and using δ_i=150,000, the two-level adaptation algorithm chooses more KFs (for each of these clips) than the corresponding numbers in Table 7. The algorithm detects the amount of activity in each shot and hence adjusts the selection frequency on a shot-by-shot basis. Each of the last five clips got total numbers of KFs in Table 7 that are larger than the corresponding values obtained in Table 12 that uses the two-level adaptation algorithm. Most of the last five clips are low-activity ones and the decision of the algorithm to reduce the selection frequency is quite appropriate. The total average of Table 12 is 2.7, smaller

Table 12. Number of selected KFs as a function of threshold values with second-level adaptation

Video name	**δ_i = 150000**		**δ_i = 250000**	
	# of KFs	**Ave/shot**	**# of KFs**	**Ave/shot**
soccer	16	5.3	10	3.3
racing-boats	15	2.5	10	1.7
action-movie	6	2.0	4	1.3
carton	22	2.4	16	1.8
celebration	11	5.5	7	3.5
comedy	7	1.4	6	1.2
ads	19	1.7	14	1.3
class	14	2.0	10	1.4
news-cast	51	2.2	37	1.6
conf-discussion	49	2.5	33	1.7
tv-show	121	1.7	93	1.3
Total Average		**2.66**		**1.82**

than the total average we got in Table 7 (2.9) that uses only one level of adaptation. Thus, the two-level adaptation solves the two shortcomings of the first level while producing at the same time a near optimal set of representative frames. Consequently, the two-level adaptation algorithm balances the trade-off and manages to select the minimal number of key frames capable of accurately representing the whole video stream.

To study the behavior of the algorithm for each clip, a set of tables similar to those given in the last two sections is presented here. For all the considered clips, the first-level threshold (δ_{a1}) will be the same as in the previous section and the second-level threshold (δ_{a2}) will be computed for each shot based on its level of activity. Moreover, in these tables we include another parameter, the Activity Index (AI) of each shot. This parameter is equal to the summation of frame differences all over a specific shot divided by the number of frame differences in that shot. Flashlights are discarded as explained before.

The heuristic used in the second-level adaptation process categorizes a shot into one of three categories according to its activity index. These categories are low activity, medium activity, and high activity. Through a large number of experiments and observations of the video clips and their frame difference graphs, we define the value of two thresholds. Any shot with normalized activity index (normalized with respect to its size) less than the first threshold is categorized as a low-activity one. A shot with a normalized activity index between the two thresholds is considered a medium-activity shot. Otherwise, the shot is considered a high-activity one. The difference between the activity levels of two shots of the carton clip is shown in Figure 8.

Table 13. KFs selection for the action-movie clip using δ_{a1} =125,000 with second-level adaptation

Shot index	**Range**	**# of frames**	**# of KFs**	**Index of selected KFs**	**AI**	**δ_{a2}**
0	0-51	52	3	0, 31, 45	4621	93750
1	52-104	53	2	52, 86	3204	125000
2	105-177	73	1	105	1261	250000

Figure 8. Activity diagram for Shots 0 and 4 of the carton clip

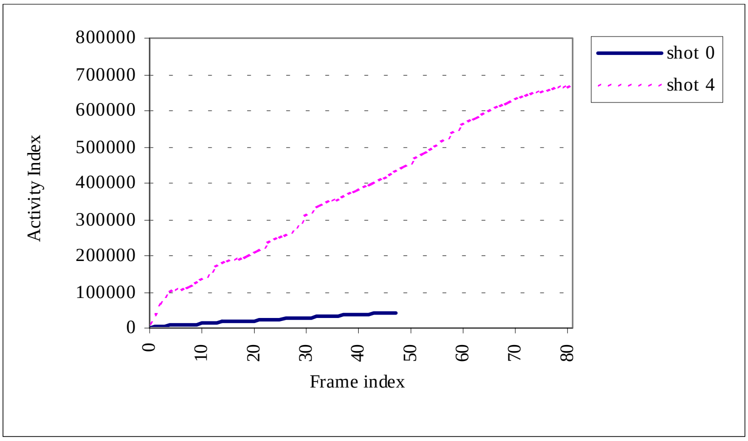


Table 14. KFs selection for the carton clip using δ_{a1} =100,757 with second-level adaptation

Shot index	**Range**	**# of frames**	**# of KFs**	**Index of selected KFs**	**AI**	δ_{a2}
0	0-48	49	1	0	916	201514
1	49-83	35	3	49, 73, 80	5826	75567
2	84-133	50	3	84, 91, 128	3523	75567
3	134-151	18	2	134, 147	6425	75567
4	152-234	83	9	152, 155, 165, 177, 184, 196, 205, 212, 223	8157	75567
5	235-244	10	1	235	3996	75567
6	245-254	10	1	245	681	201514
7	255-266	12	1	255	996	201514
8	267-330	64	1	267	1286	201514

The relative high activity of Shot 0 in Table 13 causes a reduction in that shot threshold and an increase in the number of selected KFs compared to the same shot in Table 8. For the last shot, the algorithm increases its threshold upon the detection of its low-activity nature but the number of selected KFs remains the same because it is only one frame (the minimum).

In Table 14, the remarkable increase in the activity of Shots 1, 2, and 4 causes the algorithm to increase the number of select KFs for each one of them compared to the same shots in Table 9 obtained by using the single level of adaptation. Other shots got the same number of KFs as their counterparts in Table 9. The difference in activity is remarkable from investigating the activity diagram in Figure 8, where the curve representing Shot 4 has a much higher monotonic rate than that of Shot 0. The activity diagram shows this activity gap between the two shots; thus, supporting the decision made by the algorithm to increase the sampling frequency for higher-activity shots.

The efficiency of the proposed shot-by-shot adaptation mechanism is evident in Table 15, in which the algorithm decreases the sampling frequency for low-activity shots

Table 15. KFs selection for the comedy clip using δ_{a1} =125,000 with second-level adaptation

Shot index	**Range**	**# of frames**	**# of KFs**	**Index of selected KFs**	**AI**	δ_{a2}
0	0-48	49	1	0	1418	250000
1	49-364	316	2	49, 206	1557	250000
2	365-500	136	2	365, 481	2077	250000
3	501-614	114	1	501	1730	250000
4	615-654	40	1	615	2966	125000

Table 16. KFs selection for the class clip using δ_{a1} =29,166 with second-level adaptation

Shot index	**Range**	**# of frames**	**# of KFs**	**Index of selected KFs**	**AI**	δ_{a2}
0	0-442	443	3	0, 304, 386	297	58332
1	443-804	362	1	443	79	58332
2	805-1452	648	5	805, 990, 1092, 1323, 1435	367	58332
3	1453-1866	414	1	1453	80	58332
4	1867-2306	440	1	1867	72	58332
5	2307-2641	335	2	2307, 2493	279	58332
6	2642-2792	151	1	2642	102	58332

and hence reduces the number of KFs selected for them. For instance, the number of chosen KFs for Shot 2 is reduced from 3 (Table 10) to 2. The total KFs selected for this clip is 7 compared to 11 without using the second level of adaptation that gives us a total reduction of about 45%. This was achieved while keeping the sufficient number of KFs required to properly representing each shot.

The reduction of the total number of selected KFs from 24 in Table 11 to 14 in TABLE 16 is obvious proof of the success of the proposed shot-by-shot adaptation policy. Take Shot 2 as an example, the algorithm selects only 5 KFs compared to 9 in Table 11, which is about a 45% reduction for that specific shot while the total reduction for the whole clip is about 42%. This reduction is mainly due to the adaptive nature of the algorithm in which it increases the selection frequency upon the detection of high activity and reduces it in cases of low-activity shots such as most of the shots in this clip.

USING LUMINANCE DIFFERENCES OF SUCCESSIVE FRAMES

In this section, another algorithm to select key frames from segmented video shots is introduced. The input video stream is segmented using the methodology described before, then the algorithm is applied to choose key frames that are able to represent each shot. The algorithm could be described using the following steps:

```
•   Initialize the representative set to be empty and select the first frame as the first
    element in that set.
•   Initialize i = 0 and flashIndex to the last detected flash position plus one.
•   For  j = 1  To  N-1  do  {
if ( j = (flashArray[flashIndex]+1) )
            flashIndex = flashIndex+1
  continue
if ( Diff(i , j) > $\delta_i$ )
            i = j
            Add j to the set of selected KFs
            Increment the number of selected KFs
      }
```

Where

$$Diff(i, j) = \left| \sum_{k=0}^{M-1} DC_k(i) - \sum_{k=0}^{M-1} DC_k(j) \right| \tag{2}$$

N: The number of frames in a shot.
flashArray[]: An array containing indexes to flashlight positions.
δ_i: An initial threshold that determines the frequency of sampling.
DC_i: The DC of the luminance channel at location i.
M: The number of DC terms in a DC frame.

This algorithm bears some similarity to the algorithm proposed previously but it uses direct frame differences instead of the accumulated frame differences, thus we call it the Absolute Luminance Difference (ALD) algorithm. ALD has been implemented and based on the previous experience from implementing the AFS set of algorithms so that the same first level of adaptation was included. The second level of adaptation in ALD uses a different criterion to adapt the threshold on a shot-by-shot basis. In the first algorithm, we used the shot activity in the second level of adaptation, but here we use a frame Variance Index (VI), the shot standard deviation, instead. The decision of using a frame variance comes from the nature of the selection method used. We use the direct difference between two consecutive frames and select another frame as a key frame if that difference is larger than a certain threshold. The problem here is that we cannot use the same value of that threshold all over the whole video stream because different shots exhibit different lighting conditions and amounts of activity. In an attempt to avoid the effect of these different conditions on the accuracy of the selection algorithm, we calculate the standard deviation of the shot (we call it the variance index) while discarding flashlight values as done before. The variance index is then used to categorize the shot into a number of categories; e.g., very high-variance and very low-variance shots. Then, the shot threshold is adapted according to the category to which each shot belongs. For instance, the shot threshold is reduced in case of low variance in order to select more KFs to properly represent the shot and to account for the fact that there are small differences between DC frames of that shot.

Table 17. Number of selected KFs with δ_i = 15,000 using the ALD algorithm

Video name	**δ_i = 150000**	
	# of KFs	**Ave / shot**
soccer	7	2.3
racing-boats	18	3.0
action-movie	5	1.7
carton	16	1.8
celebration	9	4.5
comedy	5	1.0
ads	19	1.7
class	13	1.9
news-cast	66	2.9
conf-discussion	57	2.9
tv-show	97	1.3
Total Average		**2.3**

Table 18. KFs selection for the action-movie clip with δ_{a1} =12,500 using the ALD Algorithm

Shot index	**Range**	**# of frames**	**# of KFs**	**Index of selected KFs**	**VI**	**δ_{a2}**
0	0-51	52	2	0, 50	539	10000
1	52-104	53	2	52, 85	829	10000
2	105-177	73	1	105	111	10000

Table 19. KFs selection for the carton clip with δ_{a1} =10,075 using the ALD algorithm

Shot index	**Range**	**# of frames**	**# of KFs**	**Index of selected KFs**	**VI**	**δ_{a2}**
0	0-48	49	1	0	182	8060
1	49-83	35	1	49	583	8060
2	84-133	50	2	84, 129	794	8060
3	134-151	18	2	134, 151	1049	8060
4	152-234	83	5	152, 153, 156, 178, 183	1156	10075
5	235-244	10	1	235	1277	10075
6	245-254	10	1	245	338	8060
7	255-266	12	1	255	124	8060
8	267-330	64	2	267, 286	578	8060

We start by performing similar experiments to those in the previous sections to determine the best values for the threshold. Then we use the best value obtained (15,000) and apply the two-level adaptation algorithm to the 11 clips used before. The overall results are shown in Table 17 as the total number of selected KFs for the best threshold value. Detailed results for each clip are given the same way they were presented in the last section.

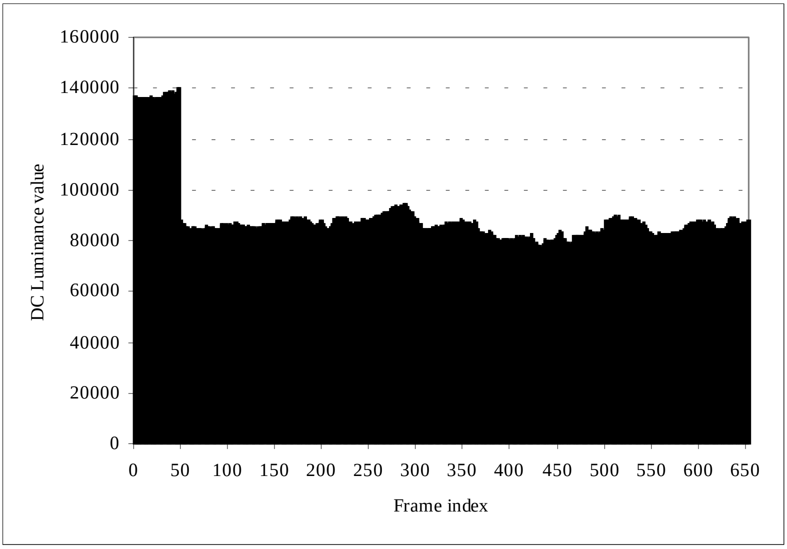
Table 18 and Table 19 report comparable results to their counterparts in the last section. We can notice the general tendency of this algorithm towards selecting fewer KFs in which their distribution may not be uniform over the length of the shot, as in Shot 4 of Table 19.

All shots in Table 20 were categorized as slightly low-variance shots, and hence their respective shot thresholds were reduced in order to capture more KFs into representative sets. Nevertheless, the algorithm selects only one KF to represent each

Table 20. KFs selection for the comedy clip with δ_{a1} *=12,500 using the ALD algorithm*

Shot index	Range	# of frames	# of KFs	Index of selected KFs	VI	δ_{a2}
0	0-48	49	1	0	202	10000
1	49-364	316	1	49	156	10000
2	365-500	136	1	365	178	10000
3	501-614	114	1	501	279	10000
4	615-654	40	1	615	307	10000

Figure 9. Value of the DC luminance frames in the comedy clip



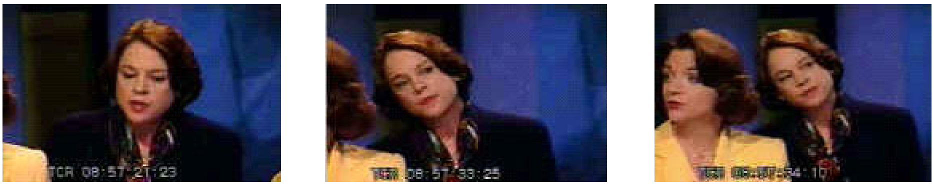
shot, and these results render the performance of the algorithm bad in this clip. For instance, Shot 1 required 3 KFs to be properly represented (Figure 6). The low- variance nature (small variation around the average) of all the shots in the comedy clip can be depicted from investigating Figure 9.

In Table 21, the algorithm chooses three KFs to represent Shot 0 which is a good choice as can be observed from Figure 10. Moreover, the results in Table 21 are almost similar to those in TABLE 16 except that the number of KFs selected for Shot 2 is just 4 instead of 5. The decision to select only 4 KFs to represent Shot 2 is an appropriate one

Table 21. KFs selection for the class clip with δ_{a1} *=2,916 using the ALD algorithm*

Shot index	Range	# of frames	# of KFs	Index of selected KFs	VI	δ_{a2}
0	0-442	443	3	0, 362, 377	513	2332
1	443-804	362	1	443	32	1458
2	805-1452	648	4	805, 810, 1369, 1420	176	2332
3	1453-1866	414	1	1453	24	1458
4	1867-2306	440	1	1867	19	1458
5	2307-2641	335	2	2307, 2484	84	1749
6	2642-2792	151	1	2642	57	1458

Figure 10. Three KFs selected by ALD to represent Shot 0 of the class video



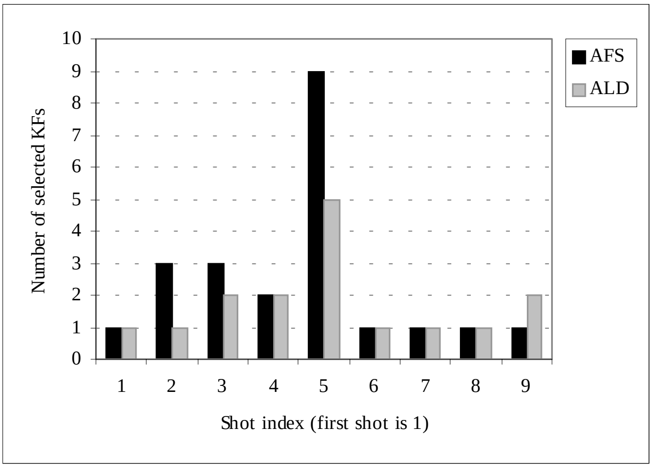
because of the intermediate activity of that shot, but the distribution of the selected KFs is not uniform all over the length of the shot.

PERFORMANCE COMPARISONS

We have proposed two sets of algorithms in this chapter, the AFS and the ALD algorithms, and in this section we will attempt to give the pros and cons of each one, focusing on the last algorithm of each set that uses two-level adaptation.

- The AFS is generally biased towards selecting more KFs as the length of the shot increases. We introduce the second level of adaptation to effectively alleviate this natural bias. On the other hand this bias is not exhibited by the ALD.
- In general, ALD has a tendency to select less key frames per shot compared to those selected by the AFS. This is obvious from investigating Table 12 and Table 17. Figure 11 shows the number of selected KFs for each shot of the carton clip using both AFS and ALD.
- The ALD is more sensitive to lighting changes than the AFS. An investigation of the results for the ads clip supports that conclusion.

Figure 11. Number of selected KFs for each shot of the carton clip using AFS and ALD



- The distribution of KFs selected by the ALD may not be uniform over the length of the shot.
- We can conclude that the performance of the AFS, measured in terms of the number of selected KFs, is slightly better than that of the ALD as it captures the notion of activity in each shot in a better way. Therefore, the AFS makes more appropriate decisions in selecting each shot threshold and hence in determining the frequency of sampling for each individual shot. Those decisions have a direct impact on its effectiveness and applicability.

CONCLUSION

In this chapter, we have introduced two algorithms to effectively and efficiently select Key Frames (KFs) from segmented videos, an essential task in fully content analysis systems. At first, we present the basic form of the AFS algorithm followed by two enhancements that overcome its original shortcomings. The first one adjusts the value of the clip threshold based on its dimension. The second enhancement measures the amount of activity in each shot then adapts the threshold for each shot on the fly based on its activity level. The application of these enhancements gives us a two-level adaptation algorithm that can accurately perform the task of KFs selection and improve the results of the basic form by about 40% in most clips. Moreover, the two-level adaptation algorithm produces a near-optimal set of representative frames, compared to the ground truth, that can be used to summarize the video data or as access points.

The second algorithm, ALD, applies a similar first-level adaptation strategy to the one used by the AFS. The second-level adaptation policy in that algorithm uses a statistical criterion to adapt the threshold on a shot-by-shot basis. Performance comparisons of both algorithms were given, and we conclude the superiority of the AFS over the ALD algorithm. From our experimentation, the proposed algorithms proved insensitive to lighting changes, an evidence of robustness. Moreover, managing to select a near-optimal set of KFs from a clip with 6139 frames in less than 2 seconds while running on a SPARC Ultra 60 machine is evidence of the efficiency of the proposed algorithms.

It is worth noting at this point that the dependency upon the format of the input compressed video (MPEG videos in our case) is isolated into the shot-detection module. This means that the proposed KFs selection module can work with any type of compressed video formats provided that the same information is passed to it from the shot-detection stage.

Using the techniques developed in this chapter and the segmentation chapter, we managed to design and implement an effective and efficient system for analyzing MPEG compressed video data (Farag & Abdel-Wahab, 2002b).

REFERENCES

Avrithis, Y., Doulamis, A., Doulamis, N., & Kollias, S. (1999). A stochastic framework for optimal key frame extraction from MPEG video databases. *Journal of Computer Vision and Image Understanding, 75*(1), 3-24.

Bimbo, A. (1999). *Visual information retrieval*. San Francisco: Morgan Kaufmann Publishers.

Chen, J., Taskiran, C., Albiol, A., Delp, E., & Bouman, C. (1999). ViBE: A video indexing and browsing environment. *Proceedings of SPIE/IS&T Conf. Multimedia Storage and Archiving Systems IV, 3846*, (pp. 148-164).

Drew, M., & Au, J. (2000). Video key frame production by efficient clustering of compressed chromaticity signatures. *Proceedings of ACM International Conference on Multimedia*, (pp. 365-367).

Farag, W., & Abdel-Wahab, H. (2002a). Adaptive key frames selection algorithms for summarizing video data. *Proceedings of the 6th Joint Conference on Information Sciences*, (pp. 1017-1020).

Farag, W., & Abdel-Wahab, H. (2002b). A new paradigm for analysis of MPEG compressed videos. *Journal of Network and Computer Applications*, *25*(2), 109-127.

Girgensohn, A., & Boreczky, J. (1999). Time-constrained key frame selection technique. *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, (pp. 756-761).

Lew, M. (ed.) (2001). *Principles of visual information retrieval*. New York: Springer-Verlag.

Nagasaka, A., & Tanaka, Y. (1991). Automatic video indexing and full-video search for object appearance. *Proceedings of Visual Database Systems 2*, (pp. 113-127).

Rui, Y., Huang, T., & Mehrotra, S. (1998). Browsing and retrieving video content in a unified framework. *Proceedings of IEEE Workshop on Multimedia Signal Processing*, 9-14.

Singh, A. (1991). *Optical flow computation: A unified perspective*. New York: IEEE Computer Society Press.

Tonomura, Y., Akutsu, A., Otsuji, K., & Sadakata, T. (1993). VideoMap and VideoSpaceIcon: Tools for anatomizing video content. *Proceedings of ACM INTERCHI*, (pp. 131-141).

Wolf, W. (1996). Key frame selection by motion analysis. *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing, 2*, (pp. 1228-1231).

Yeo, B-L., & Liu, B. (1995a). On the extraction of DC sequence from MPEG compressed video. *Proceedings of IEEE International Conference on Image Processing, 2*, (pp. 260-263).

Yeo, B-L., & Liu, B. (1995b). Rapid scene analysis on compressed video. *IEEE Trans. Circuits and Systems for Video Technology*, *5*(6), 533-544.

Yeung, M., & Liu, B. (1995). Efficient matching and clustering of video shots. *Proceedings of IEEE International Conference on Image Processing*, *1*, (pp. 338-341).

Zhang, H-J., Kankanhalli, A., Smoliar, S., & Tan, S. (1993). Automatically partitioning of full-motion video. *Multimedia Systems*, *1*(1), 10-28.

Zhuang, Y., Rui, Y., Huang, T., & Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. *Proceedings of IEEE International Conference on Image Processing*, (pp. 866-870).

Chapter V

Very Low Bit-rate Video Coding

Manoranjan Paul, Monash University, Australia

Manzur Murshed, Monash University, Australia

Laurence S. Dooley, Monash University, Australia

ABSTRACT

This chapter presents a contemporary review of the various different strategies available to facilitate Very Low Bit-Rate (VLBR) coding for video communications over mobile and fixed transmission channels as well as the Internet. VLBR media is typically classified as having a bit rate between 8 and 64 Kbps. Techniques that are analyzed include Vector Quantization, various parametric model-based representations, the Discrete Wavelet and Cosine Transforms, and fixed and arbitrary shaped pattern-based coding. In addition to discussing the underlying theoretical principles and relevant features of each approach, the chapter also examines their benefits and disadvantages, together with some of the major challenges that remain to be solved. The chapter concludes by providing some judgments on the likely focus of future research in the VLBR coding field.

INTRODUCTION

Inexpensive and evermore powerful processors coupled with faster network access, the ever-burgeoning Internet, and a significant impetus in both research and standardization have all contributed to the infrastructure of modern video coding technology. This technology has supported and continues to enable a raft of multimedia applications as diverse as home video, video-on-demand, videoconferencing, cellular videophones, remote sensing, telemedicine, interactive multimedia databases, multimedia videotex, computer games, and multimedia annotation, communication aids for deaf people, video production, video surveillance and streaming Internet video. In many of these applications, the available bandwidth is sufficient, but with the increasing focus upon offering applications for wireless and mobile environments and new pervasive computing devices such as Personal Digital Assistants (PDA) and smart telephones, the demands in terms of bit rates are becoming evermore stringent, fostering considerable impetus towards very low bit-rate (VLBR) coding techniques, in particular below 64Kbps.

Over the last decade, a number of popular video compression standards have evolved. The first was developed by the International Organization for Standardization (ISO) Moving Picture Expert Group and was known as MPEG-1 (ISO/IEC, 1993). This was designated to provide video and audio compression for CD-ROM storage by operating at typical bit rates of 1.5 Mbps. It also targeted transmission over communication channels including integrated-services digital networks (ISDN) and local area networks (LAN). The next MPEG family member was MPEG-2 (ISO/IEC, 1995), which operated at typical bit rates of 10 Mbps and specifically focused upon compression of higher resolution video signals enhancing the scope of applications for high-quality digital television (DTV) and video, including Standard Definition TV (SDTV), Digital Versatile Disk (DVD) and High-Definition TV (HDTV). The most recent coding standard is MPEG-4 (ISO/IEC, 1998), which has the explicit aim of extending the capabilities of the earlier standards, particularly in low bit-rate video coding applications, tool-kit and content-based coding. One efficient strategy used by MPEG-4 involves *sprite* technology, which enables high-quality video distribution via the Internet and mobile networks. A *sprite* is a still image (usually much larger than the display) representing a background scene that exists for many frames; it is compressed and transmitted just once and then only the camera movement parameters and foreground video objects need to be coded and transmitted. *Sprite* coding typically requires up to 50% fewer bits to achieve the same subjective quality than conventional coding, though the coding performance is very dependent on the type of video sequence being processed and the ready identification or existence of a sprite.

While ISO MPEG video/audio standards focused upon generic coding applications, namely, the storage and asymmetric distribution of media, the International Telecommunication Union (ITU) series of video coding standards H.26X (including H.261, H.263 and H.264) targeted fully symmetric, real-time, point-to-point or multi-point communications. The first ITU Telecommunications Standardization Sector (ITU-T) standard, H.261 (ITU-T Recommendation, n.d.), focused upon ISDN videoconferencing applications, with a minimum bit rate of 64 Kbps, and then integer multiples thereof (*p*x64) Kbps. Improvements in computing performance, advances in video coding research coupled with the emergence of analogue modems and packet-based Internet Protocol (IP) networks as viable channels, led to the development of H.261's successor, H.263 (ITU-T Recommendation, 1996). Three variants of the H.263 standard (ITU-T Recommendation, 1996; ITU-T Recommendation, 1998; ITU-T Recommendation, 2000) have been proposed to accommodate significantly improved compression performance together with greater flexibility and error resilience as compared to H.261, at the inevitable expense of increased complexity.

Recently, the ITU-T Video Coding Expert Group (VCEG) and experts from MPEG have collaborated to develop the H.26L standard for low bit-rate visual communications. This standard known as Advanced Video Coding (AVC)/ISO MPEG-4 Part 10/H.26L/ ITU-T H.264 (ITU-T Rec., n.d.) is now embedded in the MPEG-4 video compression standard. H.264/AVC affords a number of advances in coding efficiency enhancement by employing an improved motion-prediction capability, a smaller block-sized integer-based transform, a special deblocking filter, and content-based entropy coding, which collectively provide a 50% bit-rate savings for equivalent perceptual quality relative to the performance of earlier standards (especially in higher-latency applications) (Wiegand, Sullivan, Bjontegaard, & Luthra, 2003).

So what exactly is VLBR coding? While very much a generic term, its origins date back to the early 1980s and attempts at compressing video for transmitting videophone signals over analog telephone lines. Today, VLBR broadly encompasses video coding that mandates a temporal frequency of 10 frames per second (fps) or less, and bit rates between 8 and 64Kbps to facilitate video communications over mobile and fixed telephone channel transmissions as well as the Internet. The challenge has always been that an uncompressed video sequence for VLBR applications requires a bit rate of up to 10 Mbps (Egger, Reusen, Ebrahimi, & Kunt, n.d.), though extensive research now means that the coding technology is now able to achieve the requisite compression ratio level necessary to meet this objective.

VLBR coding schemes can be broadly classified into two categories (Figure 1), *pixel* and *non-pixel* based techniques. The former is subdivided into *pel-recursive* and *optical flow equation* based techniques, while the latter is divided into five groups: Vector Quantization (VQ), Model-based coding (MBC), Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), and Pattern-based coding (PBC).

The basic video coding process is illustrated in Figure 2 and comprises motion estimation (ME), motion compensation (MC), transform, and entropy coding. ME involves identifying the translational displacement vectors (popularly called *motion vectors*) of objects based on the changes between two successive frames in a video

Figure 1. Block diagram of various VLBR coding approaches

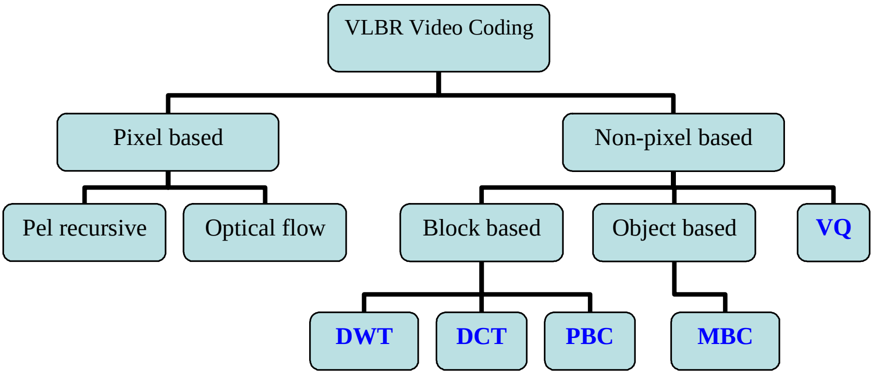
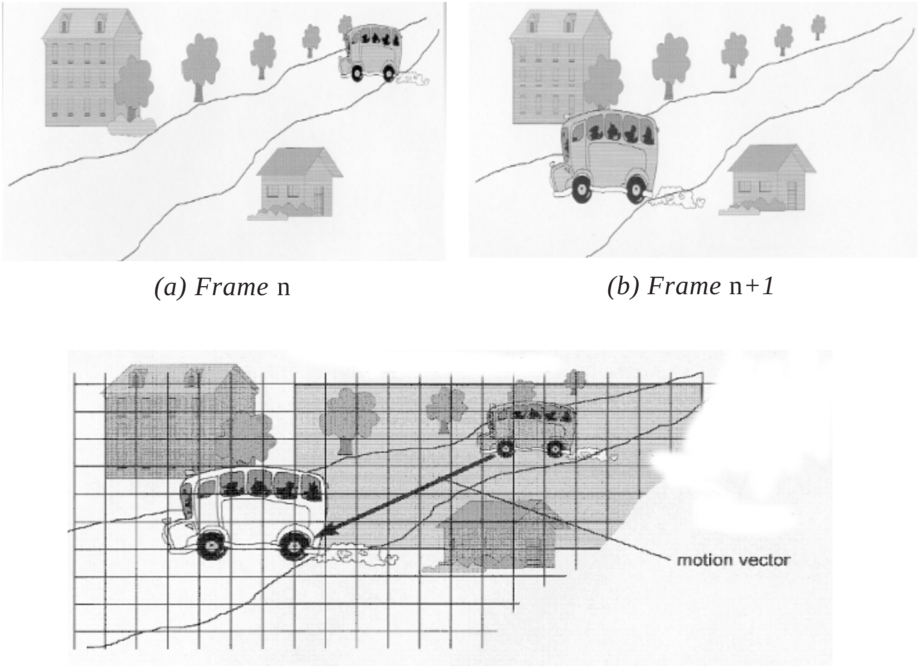


Figure 2. Motion vector calculation (Schmidt, 1999)



(a) Frame n

(b) Frame n+*1*

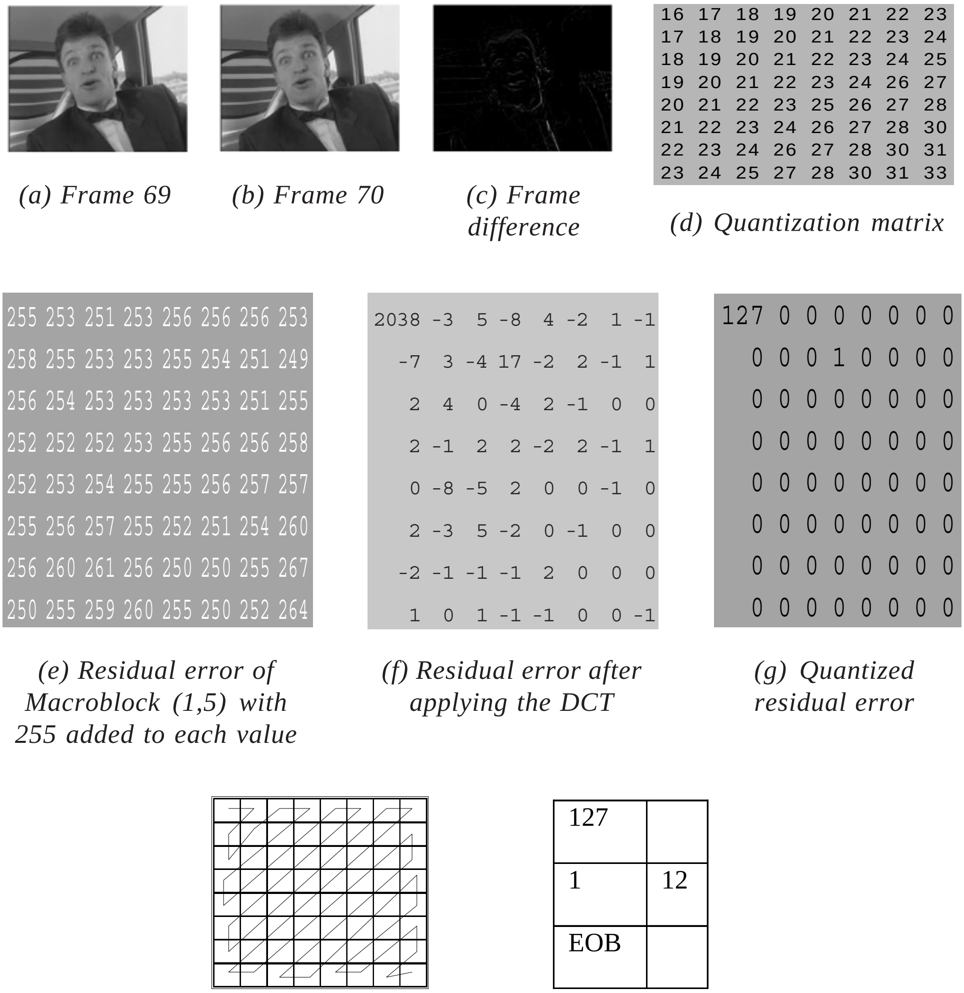
(c) Motion estimation

sequence, usually the current *n* and reference *n+1* frames. The motion vector in Figure 2(c) is calculated from the translational displacement between the macroblock in frame *n* of Figure 2(a) and its best matched block in frame *n+1* of Figure 2(b).

Having obtained the motion vector, MC involves calculating the differential signal (residual error) between the intensity value of the pixels in the moving areas and their counterparts in the reference frame, translated by the estimated motion vector. A practical hardware or software transform implementation, such as the DCT, then compresses the spatial image data, such that the total energy in the image becomes concentrated into a relatively small number of components; that is, the pixel data is decorrelated, so compression can be achieved.

An example illustrating the various steps involved in residual error processing is shown in Figure 3. In gray scale images, the range of possible pixel intensity values is from 0 to 255, so the minimum residual error is -255. Since the DCT requires positive input values, 255 is added to each error value before applying the DCT. For compression, the DCT coefficients are quantized by either a scalar or vector quantization matrix Figure 3(d). A vector quantization matrix typically has small values in top-left elements of the matrix and larger values in the bottom-right, so retaining low frequency information at the

Figure 3. Residual error processing for the Carphone *video sequence*



(a) Frame 69

(b) Frame 70

(c) Frame difference

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

(d) Quantization matrix

255	253	251	253	256	256	256	253
258	255	253	253	255	254	251	249
256	254	253	253	253	253	251	255
252	252	252	253	255	256	256	258
252	253	254	255	255	256	257	257
255	256	257	255	252	251	254	260
256	260	261	256	250	250	255	267
250	255	259	260	255	250	252	264

(e) Residual error of Macroblock (1,5) with 255 added to each value

2038	-3	5	-8	4	-2	1	-1
-7	3	-4	17	-2	2	-1	1
2	4	0	-4	2	-1	0	0
2	-1	2	2	-2	2	-1	1
0	-8	-5	2	0	0	-1	0
2	-3	5	-2	0	-1	0	0
-2	-1	-1	-1	2	0	0	0
1	0	1	-1	-1	0	0	-1

(f) Residual error after applying the DCT

127	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(g) Quantized residual error

(h) Zig zag ordering

127	
1	12
EOB	

(i) Zero RLC

expense of higher frequencies. Vector quantization however is very computationally expensive. H.26X conversely recommends scalar quantization for inter frame processing in order to reduce the computational time. The zig-zag ordering Figure 3(h) and Zero run length coding (RLC) Figure 3(i) is then applied, before entropy coding maps the resulting RLC symbols into a compressed data stream by exploiting latent redundancy by representing the most frequently occurring symbols by a small number of bits and vice versa.

Popular Video Formats and Implications on Bit Rate

In coding applications, video is often converted to one of a number of ‘intermediate formats’ prior to compression and transmission. The *Consultative Committee for International Radio communications* (CCIR) defined a unified standard for digital video and TV pictures known as Recommendation CCIR-601, which stipulates an uncompressed video data rate of 166Mbps for both NTSC and PAL TV signals. A number of variants of this standard have subsequently been defined for use in applications including DTV broadcasting, videoconferencing and video telephony. They all exploit the perceptually lower sensitivity of the human eye to color information by sub-sampling (also popularly known as *decimation*) both chrominance (colour) signals to a lower resolution relative to that of the luminance (gray-scale) signal Halsall, F. (2001) This leads to the 4:2:0 sampling nomenclature, which means that 2:1 sub-sampling is applied to both chrominance components in the horizontal and vertical directions. A set of popular picture resolutions is based upon the *common intermediate format* (CIF), where each frame has a spatial resolution of 352 × 288 pixels for the *luminance* component and 176 × 144 pixels for the *chrominance* components. The most recent video coding standard (MPEG-4 [ISO/IIEC, 1998], H.264 [ITU-T, 203]) supports various rectangular video formats based on CIF as shown in Table I from 16CIF down to Sub-QCIF. The choice of frame resolution depends on the application and available storage or transmission capacity. Among all the various video formats, Quarter-CIF (QCIF) and Sub-QCIF are primarily used for VLBR applications.

Besides sub-sampling the spatial resolution of an image, temporal sub-sampling can also be used to reduce the bit rate for transmitting video through limited bandwidth channels. Temporal sub-sampling means dropping certain intermediate frames in a video sequence so for VLBR applications, instead of 30 frames per second (fps), 15, 10, or 7.5fps may be used. For example, at 15fps alternative frames of the original 30fps sequence are dropped, while for 10fps, two consecutive frames are dropped after processing one frame. Reducing the temporal frequency by two however does not halve the bit rate, because temporal decimation introduces longer motion vectors and higher residual errors and as a consequence, the bit requirement increases.

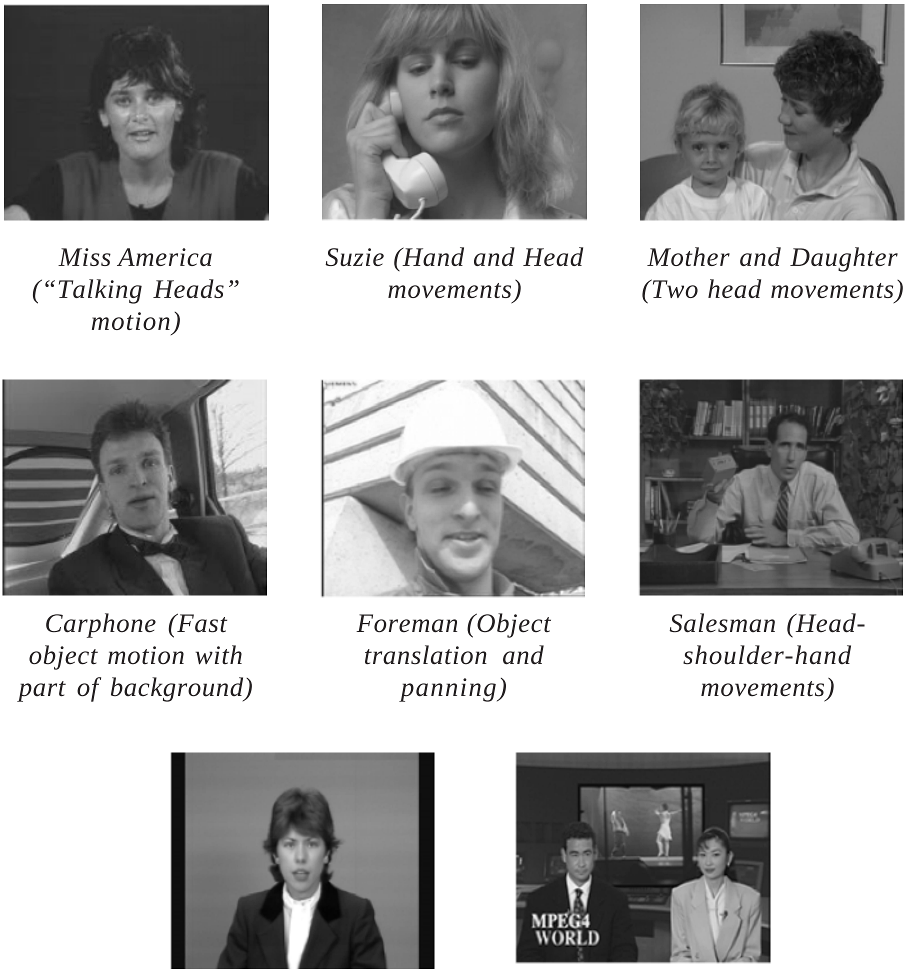
Standard Video Sequence

Due to the highly subjective nature of video, the broader coding research community has come to accept, almost be default, a series of standard test video sequences for evaluation and analysis of their results. For VLBR video coding, smooth motion video

Table 1. Video frame formats

Format	Luminance pixel resolution	Raw bit rate (Mbps) 4:2:0 @ 30 fps	Typical applications
Sub-QCIF	128 × 96	4.2	Mobile multimedia.
QCIF	176 × 144	8.7	Videoconference and mobile multimedia.
CIF	352 × 288	34.8	Videoconferencing.
4CIF	704 × 576	139	SDTV and DVD-video.
16CIF	1408 × 1152	560	HDTV and DVD-video.

Figure 4. Standard QCIF video sequences with their motion type



Miss America (“Talking Heads” motion)

Suzie (Hand and Head movements)

Mother and Daughter (Two head movements)

Carphone (Fast object motion with part of background)

Foreman (Object translation and panning)

Salesman (Head-shoulder-hand movements)

Claire (“Talking head” motion)

News (“Talking head” with moving background motion)

such as the *talking head* (head-shoulder) type sequences are used for performance comparison. Popular smooth motion sequences, which are usually in the QCIF format, include *Miss America*, *Suzie*, *Mother and Daughter*, *Carphone*, *Foreman*, *Salesman*, *Claire* and *News* (Figure 4). While they are all broadly classified as smooth motion sequences, some exhibit different types of motion, so for example *Carphone* possesses relatively high background object motion compared with either *Miss America* or *Claire*.

Frame Types

Each frame in a video sequence is encoded to produce coded frames of which there are three main types namely; Intra coded (I-frames), Predicted (P frames), and Bidirectional predicted (B-frames). I frames are encoded without any motion-compensation and are used as a reference for future predicted P- and B-type frames. I-frames however require a relatively large number of bits for encoding. P-frames are encoded using MC prediction from a reference frames which can be either an I or P frame. P frames are more efficient in terms of the number of bits required compared to I-frames, but still require more bits than B-frames. B-frames are encoded using MC prediction from two reference frames, the P- and/or I-frames before and after the current B-frame. They require the lowest number of bits compared to both I- and P-frames but incur both a computational and storage overhead. Thus, for VLBR applications in mobile and handheld devices for instance, for fast operations B frames are often not considered. All these frames are normally processed in a group, consisting of an I frame followed by a series of P- and B-frames to form the so-called Group of Pictures (GOP) arrangement used in MPEG-2. For general applications, the GOP length is usually relatively small (£12) though for VLBR applications this is large.

Video Coder

While most video encoders comprise the basic functionality of prediction, transformation, quantization, and entropy encoding, there still exists considerable variation in the structure of the Coder/Decoder (CODEC) arrangement. The basic encoder structure shown in Figure 5 (Richardson, 2004) includes two dataflow paths. The *forward* path (left to right), and a *reconstruction* path (right to left), while the corresponding dataflow path for the Decoder is shown in Figure 6 (Richardson, 2004). This illustrates the symbiosis that exists between the coding and decoding processes. An arbitrary input frame F_n is firstly sub-divided into *Macroblocks* (MB), which generally correspond to a group of 16×16 non-overlapping pixels in the original image. Each MB is then coded in either *intra* or *inter* mode. In both cases, a prediction MB *P* is formed based on a

Figure 5. Encoder structure

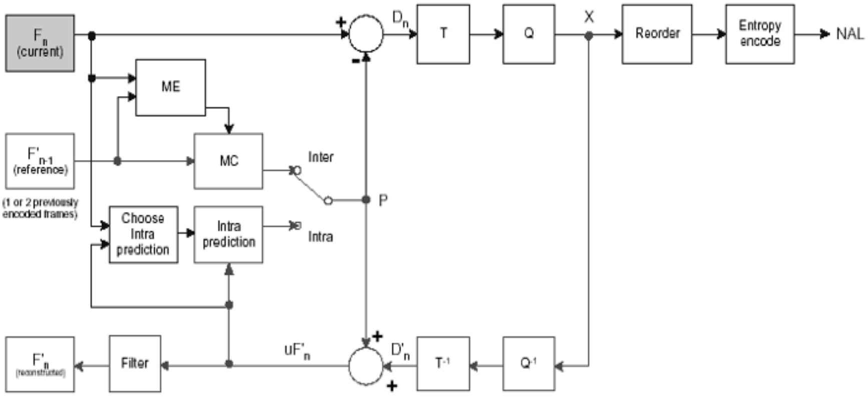
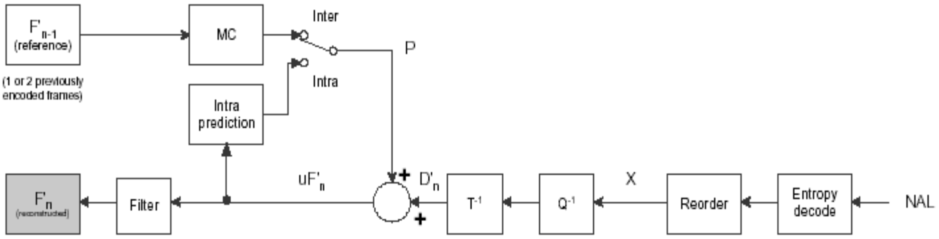


Figure 6. Decoder structure



reconstructed frame. In the intra-mode, *P* is formed from samples in the current frame *n* that have earlier been encoded, decoded and used to reconstruct the unfiltered sample uF'_n. In inter mode, *P* is formed by MC prediction from one or more reference frames F'_{n-1}; however, the prediction for each MB may be formed from one or two previous or forward frames (in time order) that have already been encoded and reconstructed. The prediction P is subtracted from the current MB to produce a residual error MB D_n which is then transformed using a block transform (T) and quantized (Q) to give a set of coefficients *X* which are re-ordered (Zigzag scanned) and entropy encoded, using any efficient variable length coding algorithm. The entropy-encoded coefficients, together with side information required to decode the MB (such as the MB prediction mode, motion vector and quantization step size) form the compressed bit stream, which is passed to the *Network Abstraction Layer* (NAL) for transmission or storage.

The decoder path uses the quantized MB coefficients *X* in order to reconstruct a frame for encoding further MBs. The coefficient X are re-scaled (Q^{-1}) and inverse transformed (T^{-1}) to produce a differential MB D'_n, which is a distorted version of D_n. At the decoder, the incoming compressed bit stream is disassembled and the data elements are entropy decoded and reordered to produce the quantized coefficients *X*. These are rescaled and inverse transformed to form D'_n. The decoder then creates a prediction MB *P*, identical to the original prediction *P* formed in the encoder. *P* is added to D'_n to produce uF'_n which is subsequently filtered to create the decoded MB F'_n.

This represents the general structure of a modern video CODEC and while some functional elements are modified or additional blocks included for special applications, the basic structure remains the same. Example of some of these components include a pre-processing filter — to reduce the noise introduced in capturing images from low-quality sources, or camera shake; and a post-processing filter — to reduce the blocking and/or ringing effects, and pattern prediction mode for pattern-based coding. These enhance performance in certain cases, though they also increase the hardware complexity.

PIXEL-BASED CODING

In pixel-based coding, motion vectors are calculated for each pixel in an image, with pel-recursive and optical flow being the two most popular methods. These will now be reviewed.

Pel-recursive Technique

Definition

In the pel-recursive method, the motion vector of a pixel is estimated by recursively minimizing a nonlinear function of the dissimilarity between two certain regions located in two consecutive frames, with the term region referring to either a single or group of pixels. Netravali and Robbins (1979) published the original pel-recursive algorithm to estimate motion vectors for motion-compensated inter-frame video coding. They defined the displacement frame difference (DFD) as:

$$DFD(x, y; d_x, d_y) = f_n(x, y) - f_{n-1}(x - d_x, y - d_y) \qquad \textbf{(1)}$$

where the subscripts *n* and *n*-1 indicate two successive frames upon which motion vectors are to be estimated; *x*, *y* are coordinates in image planes, and d_x, d_y are the two displacement vector components. Netravali and Robbins converted this nonlinear displacement estimation function into a minimization problem, which can be solved by various gradient descent techniques. The *steepest descent* method is actually used by Netravali and Robbins because of its simplicity and the existence of satisfactory analysis, and it is often used as the reference for comparing and evaluating new methods. The algorithm can be applied to a pixel either once or iteratively applied several times for displacement estimation. The current displacement vector estimation is used as the initial estimate for the next pixel, and the recursion can be performed horizontally, vertically, or temporally, which means the estimated vector is passed to the spatially bit-wise pixel in neighboring frames.

Features

The main features of the pel-recursive technique are that it is:

- very suitable for VLBR applications;
- better performing in video coding compared to block-based methods;
- noise sensitive;
- computationally very expensive; and
- prone to errors in the gradient estimation process due to the small noise

Evolution of Pel-recursive Technique

Bergmann (1982) modified the Netravali and Robin's (1979) algorithm by using the Newton-Raphson method of minimization that converges faster than the steepest descent method. Cafforio and Rocca (1983) introduced a new steepest descent method that was more effective for uniform regions, where the gradient is very small. Walker and Rao (1984) and Tekalp (1995) proposed an adaptive step size, using a small step size at edges or in non-homogenous regions and large step sizes in smoother areas. This variable step size approach substantially improved both the convergence rate and motion vector accuracy.

Biemond, Looijenga, Boekee, and Plompen (1987) proposed a Weiner-based motion estimation using a Taylor series, which had the advantages of: (1) generating a dense motion field that yielded a motion vector at every pixel; (2) a smooth motion field inside moving objects; and (3) sub-pixel accuracy. These benefits were counterbalanced, however, by the technique not being sufficiently robust to handle large motion vectors and providing slow convergence at the boundaries of moving objects and images. To overcome these problems, different scanning orders were used for ME (Csillag & Boroczky, 1999). A pel-recursive algorithm based upon recursive least-squares (LS) estimation, which minimized the mean square prediction error, was proposed by Gharavi and Reza-Alikhani (2001). This reduced the prediction error compared with the steepest descent method, particularly in regions where motion activity was relatively high. A fast ME algorithm that combined the advantages of both the block matching and pel-recursive methods using LS on a whole block was proposed by Deshpande and Hwang (1998). This exhibited superior speed performance but degraded picture quality. Usually scalar quantization is used on DFD; however, scalar quantization of DFD severely limits the rate-distortion performance of pel-recursive video coding systems. Shen and Chan (2001) used Vector Quantization to code the DFD. Although this method is suitable for VLBR applications, it has high computational complexity.

Summary

From a purely performance perspective, pel-recursive ME techniques are superior for video coding, but they are also computationally very expensive and noise sensitive, and, therefore, unsuitable for real time applications.

Optical Flow Equation Based Technique

Definition

This approach provides much greater accuracy in displacement estimation compared to both the pel-recursive and popular block-based ME techniques, because of the very high number of motion vectors (one vector per pixel) involved. Optical flow is the 2-D distribution of apparent velocities of movements of intensity patterns in an image plane (Horn & Schunk, 1981). It is important to highlight that, in certain cases, optical flow and 2-D motion are not the same. Horn and Schunk used the example of a uniform sphere rotating at a constant speed in a scene. Assuming the luminance and all other conditions remain the same, when the image was captured, there is no change in intensity pattern for this image sequence, though there is clearly a change in 2-D motion. Conversely, in a stationary scene, all objects in 3-D space are fixed, and if the luminance changes when the picture is captured in such a way that there is movement of intensity patterns (by the effect of different lighting imposing), although there is zero motion, the optical flow will clearly be is obviously non-zero.

Features

The main features of the optical flow equation based technique are that it is:

- not appropriate for high motion sequences;
- appropriate for computer vision;
- noise sensitive;
- known for aperture problems;
- not appropriate for VLBR due to being computationally very expensive and its high bit overhead;
- undesirable because of its effect of smoothness constraint; and
- prone to errors in gradient estimation.

Evolution of Optical Flow Based Techniques

The high bit overhead and computational complexity in optical flow ME prevents it from being of practical use in video coding. Shu, Shi, and Zhang (1997) proposed an efficient compression algorithm for VLBR coding applications using optical ME, DCT coding of the optical motion vectors, and region-adaptive thresholding. The pre-processing requirement, accurate ME due to the thresholds, and computational complexity militate against this method being applied to real-time VLBR coding applications. Liu, Chellappa, and Rosenfeld (2003) recently proposed an adaptive optical flow estimation algorithm by combining the 3-D structure tensor with a parametric flow model to improve the ME accuracy. Ku, Chiu, Chen, and Lee (1996) modified the existing optical flow algorithm by incorporating an edge- preserving constraint and pyramid approach to generate a more accurate motion field and reduce the probability of becoming trapped in local minimum.

Summary

The main drawback of optical flow ME is that, while it is a very effective technique in the field of computer vision, the additional side information that must be encoded and transmitted, coupled with the noise sensitivity, aperture problems, and computational expense, mean that it is not appropriate for real-time VLBR coding.

NON-PIXEL BASED CODING

Non-pixel based coding generically refers to those techniques where motion vectors (MV) are calculated based on a grouping of pixels rather than just a single pixel. Using a MV for each pixel is much more accurate but incurs a substantial encoding overhead, which is why it is unsuitable for real-time VLBR applications. There are a myriad of different ways that the grouping can be defined. In vector quantization (VQ), for example, it is defined as a predefined template, while in object-based coding, an arbitrary shape that covers an entire object is defined. In block-based techniques, pixels are collected into rectangularly shaped, block combinations (usually referred to as *macroblocks*) such as 16 × 16, 8 × 8 and 4 × 4 pixels. In this section, we will review the various features, advantages, and disadvantages of these approaches.

Vector Quantization

Definition

Vector Quantization (VQ) is a non-standard video coding technique but is very effective for data compression because, unlike scalar quantization, it seeks to exploit the correlation between components within a vector. Optimum coding efficiency can be achievable if the vector dimension is infinite, so the correlation between all components is exploited. The design of an optimal VQ from empirical data was originally proposed and investigated by Linde, Buzo, and Gray (1980), in their LBG clustering algorithm, while a good overview of VQ-related research was presented by Nasrabadi and King (1988).

A vector quantizer is defined as a mapping Q of K-dimensional Euclidean space R^k into a finite subset Y of R^K. Thus

$$Q : R^K \rightarrow Y \qquad \textbf{(2)}$$

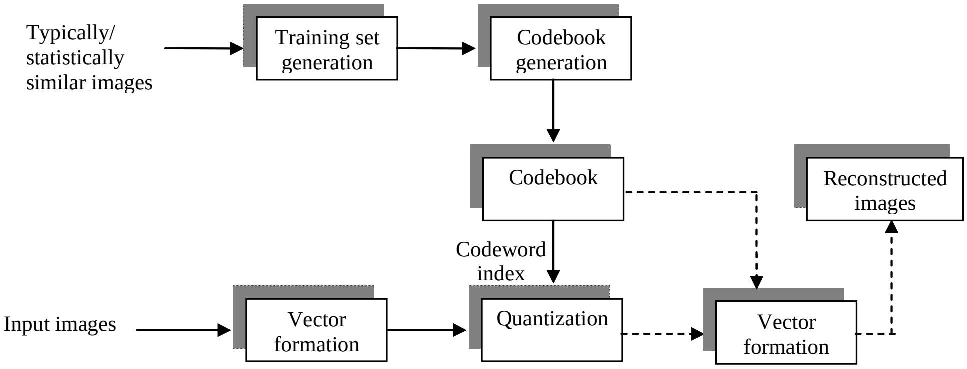
where $Y = (\hat{x}_i ; i = 1,2,\cdots,N)$ is the set of reproduction vectors from input vector x and N the number of vectors in Y. The set Y is called the *Codebook* and each element $\hat{x}_i$ is called a *codeword*.

Features

The main features of VQ are that it:

- can be used for VLBR coding of image sequences;
- exploits the correlation between components within the vector;
- decodes the image sequences simply by using a look-up table;
- increases the complexity exponentially with dimension that limits real-time applications;

Figure 7. Vector Quantization CODEC



- is difficult to select an appropriate training set for codebook generation;
- is a very challenging task to generate a universal codebook that is image dependent; and
- is intractable to scale the codebook for better coding efficiency.

Basic Procedure

Figure 7 shows the functional block diagram of a basic VQ codec. It comprises (1) vector formation; (2) training set generation; (3) codebook generation, and (4) quantization. Each of these constituent blocks is now briefly examined:

1. *Vector formation* is the decomposition of images into a set of 2-D vectors. These vectors are usually not formed directly from the spatial image data, but rather from the transform domain (transform coefficients). This ensures that the vector size is compressed with a corresponding reduction in the computational complexity of VQ coding.
2. Achieving optimal VQ is highly dependent on choosing the best *training set*, which is selected from either the image or statistically similar images.
3. *Codebook generation* is the most important process in VQ, since coding efficiency will be optimal when the interrelations between the codewords in a codebook are minimized and, concomitantly, the intra-relations between codewords in separate codebooks is maximized. In Gersho (1982), for example, the mean square error (MSE) criterion is applied such that the input vector source is classified into a predefined number of regions by the minimum distance rule between intra codewords and the maximum distance rule between inter codewords.
4. *Quantization* selects the most appropriate codeword in the codebook for each input vector using some prescribed metric such as MSE or absolute error. An exhaustive search process over the entire codebook provides the optimal result but is time consuming. There are alternative search algorithms such as tree-search, which though sub-optimal, are much faster.

Evolution of VQ

Research into trying to improve the potential of VQ for VLBR applications includes Baker and Gray’s (1983) design of a mean/shape vector quantizer (M/SVQ) where the vector mean is scalar quantized while the resulting error vector, obtained by subtracting the sample mean from the input vector, is vector quantized. Baker (1983) extended their work to mean/residual VQ (RVQ) and subsequently mean/reflected residual VQ (Barnes, Rizvi, & Nasrabadi, 1996). An advantage of the latter approach is the relatively small number of bits required due to the reduced codebook size. A small codebook, however, does not reproduce edges very well, particularly in VLBR coding applications because of the diversity of edge types that results in perceptible edge degradation in many cases. Ramamurthi and Gersho (1986) proposed a *Classified* VQ (CVQ) algorithm that organized each image vector into several classes to address this edge degradation problem. Vectors with distinct perceptual features, such as edges, are generated from different subsources, belonging to different classes. The classifier then determines the class for each vector so each is appropriately coded with a quantizer specifically designed for that particular class, thus providing an overall better perceptual quality than classical VQ. Chang and Chen (1996) used an edge-based, site-matched finite-state CVQ that only transmitted moving blocks (vector) in an image, while Yong and Tseng (2001) proposed a smooth side-match CVQ that combines CVQ and a variable block-size segmentation, which outperforms CVQ when fixed size blocks are used.

Codebook replenishment and inter-frame VQ (Goldberg & Sun,1989; Karayiannis & Li, 2002) reduce the number of bits because of the adaptability in transmitting a small codebook that matches the local statistics of the image being coded. *Transform* VQ (TVQ) converts statistically dependent (correlated) pixels into uncorrelated coefficients by transforming data from the spatial to the frequency domain. Due to the computational complexity, the image is usually divided into manageable sub-images and the transform performed on each of these. The transform coefficients are then non-uniformly quantized using a scalar quantizer. The quantization levels are defined by a bit assignment matrix in which certain high-frequency coefficients are discarded. This matrix is used at both the coding and decoding. Exploitation of the correlation of adjacent picture elements and significant codebook size reduction are the main advantages of TVQ, though this is counterbalanced by blocking artifacts — which can be removed though the classified Lapped orthogonal transform (Venkatraman, Nam, & Rao, 1995).

Future Trends of VQ

Vector quantization is a non-standard video coding strategy; however, many researchers are engaged in trying to exploit its potential (Kwon, Venkatraman, & Nasrabadi, 1997; Man, Queiroz, & Smith, 2002; Pan, Lu, & Sun, 2003; Shen, Zeng, & Liou, 2003; Terada, Takeuchi, Kobayashi, & Tamaru, 1998), though real-time VQ is limited by the high computational overhead that often necessitates a parallel implementation. A real-time multi-stage hierarchical VQ has been proposed by Terada et al. (1998) using a functional memory- type parallel, *Single Instruction, Multiple Data* (SIMD) processor. Wireless or mobile video communications through a noisy or noiseless environment is feasible with VQ using three-dimensional sub-band coding (Man et al., 2002). Using the sum and variance characteristics, an efficient encoding VQ algorithm is proposed by Pan et al. (2003), where each vector is separated into two sub-vectors: one composed of the first half of vector components, and the second consisting of the remainder. Three inequalities based on the sum and variance of a vector and its two sub-vectors components are then used to reject those codewords that are not possible to be the nearest codeword. This results in considerable computational time savings, and also does not introduce any additional distortion compared to the conventional full-search algorithm. Shen et al. (2003) described a sophisticated codebook-updating scheme where the operational codebook is updated using not only the current input vector but also the codewords at all positions within a selected neighborhood (defined as locality), while the operational codebook is organized in a cache manner. This algorithm memories information concerning previously coded vectors in quantizing the current input vector while updating the operational codebook.

Summary

To achieve high video data compression, quantization is the pivotal element in the CODEC process. VQ performs better than scalar quantization as it exploits the intrinsic spatial and temporal correlation of adjacent pixels in the video data. An idealistic VQ approach based on a combination of variable vector size, classified lapped transform, multi-stage, dynamic codebook updating using locality, parallel computing structures, together with a small codebook size could theoretically be a very strong competitor to any contemporary digital video coding standard. Pragmatically, however, it infeasible to incorporate all the aforementioned properties, because many have individual trade-offs. One major problem with VQ is that it does not reconstruct edge vectors efficiently as the codebook is unable to reproduce all possible patterns. VQ with a dynamically updated codebook based upon locality provides a good approximation of sub-image but often requires a large number of bits due to the high codebook transmission frequency to the decoder. Generally, a VQ coding system requires pre-processing for vector and codebook formation, as well as the codebook transmission overhead. The codebook searching time also takes a significant amount of time, and these limitations ultimately restrict the range of applications of VQ.

Model-based Coding

Object-based coding for VLBR is a fundamental component of the MPEG-4 video-coding standard, although the concept is not exactly brand new. Model-Based Coding (MBC), for example, was first introduced in 1981 by Wallis, Pratt, and Plotkin (1981) and represents a special kind of object-based coding. Applications of MBC, however, have tended to be been restricted to video telephony and conferencing, where only one or two objects are considered and some *a priori* knowledge about a scene's content exists. In contrast, MPEG-4 is able to handle any number of arbitrary-shaped objects (though practical limitations usual constrain the number) without *a priori* information being required about the scene contents.

Definition

In contrast to the conventional digital video coding standards, which are based on eliminating spatial and temporal redundancies in a sequence by wave-transforming, MBC treats the images as two-dimensional (2-D) projections of a 3-D world with *a priori* knowledge concerning the scene contents (Li & Chen, 1998). One or more moving objects in a video sequence is analyzed using computer vision techniques to create a parametric model incorporating key information concerning the size, location, and motion of these objects. At the decoder, the model synthesizes each object by using computer-graphical methods, with automatic tracking techniques enabling the model to mimic the respective object's movements. The parameters needed to animate the model are then coded and transmitted to the receiver, which reconstructs the model (Pearson, 1995). For low-quality images, the animation data are sufficient to give a good approximation to the original image sequence, but for higher quality an additional residual pixel signal is required that typically comprises the coded frame differences between the original video sequence and the animated model. As with parametric models in other media forms, such as Linear Predictive Coding (LPC) in speech synthesis, the bit-rate performance of MBC is very good because only the model parameters are transmitted, and this has attracted attention as it provides high-quality images at very low bit rates. As a consequence, the MBC has been viewed as a potential competitor for MPEG-4 and H.263, though major practical problems remain to be solved, namely, the difficulty in modelling unknown objects and the inevitable presence of analysis errors.

Features

The main features of Model-Based Coding include:

- the encoder and decoder use exactly the same parametric model for each object in a scene;
- *a priori* knowledge of a scene's contents is required before coding;
- automatic computation of the key model parameters is an intractable problem; and
- it is only suitable for VLBR real-time video coding applications such as video telephony and conferencing.

Basic Procedure

A typical MBC video coding functional diagram is shown in Figure 8. A 2-D or 3-D parametric model representation for the video sequence is first constructed. Each input image is analyzed and a set of model parameters calculated and transmitted to the decoder. The reconstructed image is then synthesized using these parameters using an identical model to that at the encoder. The image modelling, analysis, and synthesis processes are the three kernel elements of MBC. In image modelling, a geometric wireframe model is used for image structure description (Figure 9). The generality and flexibility of the wireframe model may become disadvantageous, for example, if an object implies hidden or implicit structure or motion constraints (Buck & Diehl, 1993). The geometric models are classified into a surface-based description and volume-based description. The surface description is easily converted into a surface representation that can be encoded and transmitted. In these models, the surface shape is represented by a set of points that represent the vertices of triangle meshes. Any surface variation can be represented by adjusting the size of the patches; that is, for more complicated areas, more triangle meshes are used to approximate the surface, while for smooth areas, the mesh can be expanded and fewer vertices used to represent the surface. The volume-based description is a usual

Figure 8. Basic schematic diagram of model-based image coding

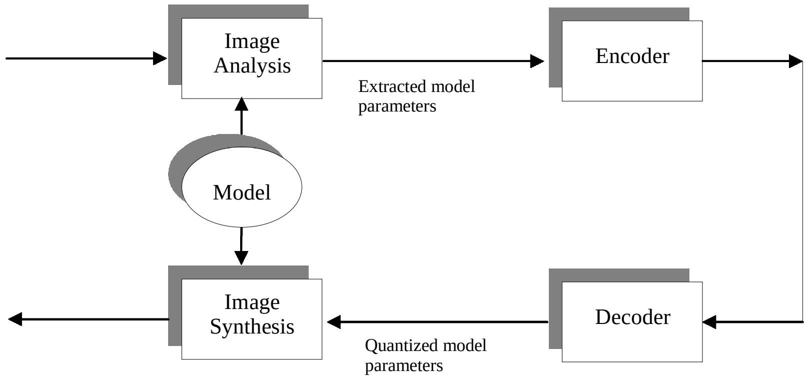
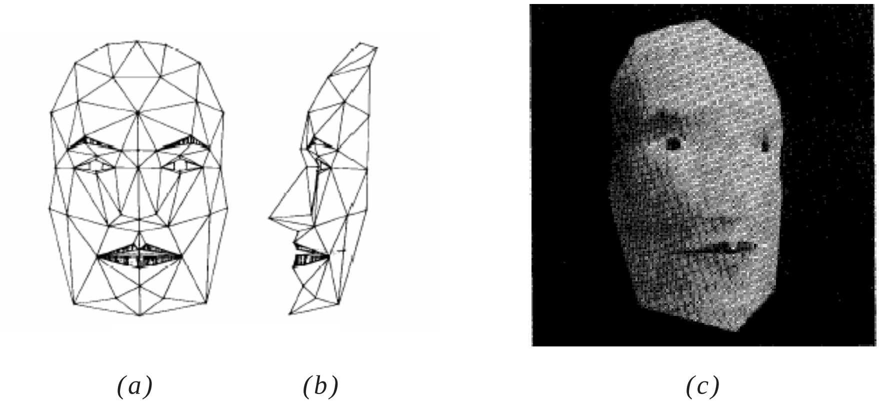


Figure 9. Wire-frame model of the face: (a) front view, (b) profile, and synthesized face image assuming simple surface and one light source (Platt & Badler, 1981; Parke, 1982)



approach for modelling most solid-world objects (Shi & Shun, 1999) as it is insensitive to scene and object complexity.

Image analysis techniques are employed to automatically extract the key model parameters at the encoder, such as the position of the facial features, motion information, and depth. The position of the facial features is the most important step in being able to automatically fit a generic wireframe to a specific face, which in turn impacts on the motion. Image synthesis reconstructs a natural-looking facial image using the facial-image model and encoded parameters.

Image analysis extracts vital parameters required by the coding, such as the position of the facial features, motion parameters, and depth. The motion of the head and changes of facial expression are used to extract the parameters, which are transmitted for a number of feature points. The position of the facial features is the most important step for automatically fitting a generic wireframe to a specific face. ME is also dependent on this. Image synthesis involves reconstructing a natural-looking facial image using an image model that incorporates the transmitted parameters.

Evolution of MBC

MBC analysis-synthesis image coding (Pearson, 1989) has been successfully used in low bit-rate video telephone (Welsh, 1988) applications. At the encoder, input images are processed by a 3-D analysis model and represented by a set of extracted parameters, which are then transmitted to the decoder, which, using exactly the same 3-D model as the encoder, synthesizes the image.

The following steps comprise MBC:

- **Human face modelling:** A predefined coarse 3-D wireframe model is manually adjusted to the input image and the deformed wireframe face model representing the shape of the transmitted human face is registered.

- **Encoder parameter extraction:** The motion of the head and changes to facial expression is used in extracting the parameters, which are transmitted for a number of feature points.
- **Receiver image synthesis:** The decoder modifies the 3-D wireframe face model using the transmitted parameters and texture maps an original 2-D face image onto the model to create the decoded image.

The ME method of model-based coding can be categorized in two stages. The first identifies the features located on the object and tracks these between frames to form a 2-D correspondence between consecutive frames, and the 3-D motion parameters are estimated between corresponding pairs. Welsh (1991) established the correspondence for the moving feature points by template matching from a generated codebook, which were then used to estimate the best fitting values of global motion parameters. Fukuhara, Asai, and Murakami (1993) used template matching to also establish these correspondences, but applied a neural network to estimate the head motion parameters.

Nakaya, Chuah, and Harashima (1991) proposed a model-based/MC-DCT hybrid coding system by combining the advantages of MBC and waveform MC-DCT to overcome the limitations of MBC. It incorporates waveform coding into the MBC, which can code those objects not covered by the basic model and cancel analysis errors so when the first fails, the latter is still able to improve coding efficiency. In this hybrid model, the face region of the images is approximately coded by a MBC using the facial model, while the shoulder region is assumed to be a rigid object, and thus motion is compensated as a single large block before being processed by the MC-DCT coder. The background is coded using only MC-DCT. Overall, this hybrid approach is especially notable at very low transmission rates, that is, at 16Kbps, though its performance depends on the efficiency of the image analysis and synthesis technologies used. Better performance can be achieved by establishing an image assessment strategy so that model and waveform-based coding are kept from interfering with each other.

Future Trends of MBC

The effective modelling of objects is the key issue in MBC. The performance and complexity of image analysis/synthesis depends on the model adopted. In a conventional MBC scheme, predefined static models are generally used, though these can neither be adapted to fit the real features of the object nor updated dynamically in the applications. Moreover, a very specific model is necessary for a particular object, so a generic model cannot be used to generate models for similar objects, thereby saving memory space (Sui, Chan, & Siu, 2001). Certain methods generate the object model with stereo graphics (Chen & Lin, 1997) and laser scanning. Though laser scanning can provide a very accurate model of an object, the size and cost of the equipment are high. Conversely, stereo graphics are used in specific views. A gradually generated and dynamically modified model using scanned video frames is an alternative for real applications (Sui et al., 2001). It has some advantages over conventional methods, such as no specific view with known orientation of the object is required, extracted information is used to update the generic model, and the feature extraction processes for different views can be performed in parallel. Object analysis largely depends upon extracting key features including eyes, eyebrows, ears, mouth, and nostrils. Among these, eyes, eyebrows, and mouth are non-rigid objects, while the ears and nostrils can become occluded. The effects of photometric changes, occlusions, and shear effects make the selection of points for tracking ill-defined for all but a few points (Bozdagi, Tekalp, & Onural, 1994). In block tracking for 3-D ME solution, some of blocks may encompass distinct features, which are easily tracked and contribute favorably, whereas others introduce errors. The removal of the large motion vectors that exhibit disagreement between the 2-D translational field and a projection of the estimated 3-D rotation can improve the ME (Woods, 2001).

Summary

Existing block-based coding standards, such as H.263, do not consider the arbitrary shape of moving objects and, as a result, their prediction efficiency is compromised. To reduce the prediction error for arbitrary-shaped objects, switching between a waveform coder and MBC sometimes can exhibit superior performance at low bit-rates. The general working principle of a *switching coder* is to change on the basis of the required number of bits and image quality for enhancing the overall coding performance. Although MBC opens the possibility of very low bit-rate image transmission, the following problems inhibit its acceptance:

1. **Generality:** The main assumption that the input images always consists of a moving face and shoulder is not appropriate for practical use. To enhance its application in general cases, precise 3-D structure of the scene from 2-D images is necessary but using existing technology this is very difficult.
2. **Analysis errors:** This can sometimes affect the decoded images seriously, such as a laughing face being transmitted as a crying face or an upside-down face at the decoder. The face animation has been adopted for the MPEG-4 visual coding. The body animation is under consideration for version 2 of the MPEG-4 visual coding.

Wavelet-Based Video Coding

Definition

In the context of low bit-rate video coding, wavelet theory has demonstrated an ability to not only provide high coding efficiency, but also spatial and quality scalability features. Grossman and Morlet (1984) first introduced the wavelet transform in 1984 by mapping a time or spatial function into a two-dimensional function as a and t , where a corresponds to the amount of time scaling or dilation and t represents the time shift. The wavelet transform is defined as follows. Let $f(t)$ be any square integrable function, i.e., it satisfies

$$\int_{-\infty}^{+\infty} \left| f(t) \right|^2 dt < \infty \, . \qquad \textbf{(3)}$$

The continuous-time wavelet transform of with respect to a wavelet is defined (Shi & Shun, 1999) as:

$$W(a,\tau) = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{|a|}}\psi^*\left(\frac{t-\tau}{a}\right)dt \tag{4}$$

where a and τ are real variables and * denotes complex conjugation. The wavelet function is then expressed as:

$$\psi_{a\tau}(t) = |a|^{-1/2}\psi\left(\frac{t-\tau}{a}\right) \tag{5}$$

where $\psi(t)$ is known as the *mother wavelet,* which is stretched when a>1 and contracted or dilated when 0<a<1. This function integrates to zero (must have zero mean) and be square integrable (possess finite energy). The following Haar wavelet is an example of a simple wavelet function.

$$\psi(t) = \begin{cases} 1 & 0 \le t \le 1/2 \\ -1 & 1/2 \le t \le 1 \\ 0 & otherwise \end{cases} \tag{6}$$

For digital image compression, $f(t)$ is represented as a discrete superposition rather than an integral, so dilation and translational parameters take the dyadic values of $a = 2^k$ and $\tau = 2^k l$, where both k and l are integers. $\psi_{a\tau}(t)$ then becomes

$$\psi_{kl}(t) = 2^{-k/2}\psi\left(2^{-k}t - l\right) \tag{7}$$

and the corresponding DWT can be written as:

$$W(k,l) = \sum_{-\infty}^{+\infty} f(t)\psi_{kl}^*(t)dt \tag{8}$$

with the inverse DWT being

$$f(t) = \sum_{l=-\infty}^{+\infty}\sum_{l=-\infty}^{+\infty} d(k,l)2^{\frac{-k}{2}}\psi\left(2^{-k}t - l\right). \tag{9}$$

The wavelet transform at the discrete values of a and τ are represented by

$$d(k,l) = W(k,l)/C \tag{10}$$

where C is an integral constant. The $d(k,l)$ coefficients are referred to as the DWT of the function $f(t)$ (Daubechies, 1992; Vetterli & Kovacevic, 1995). The wavelet transform of $f(t)$ still has the discretization in its two parameters, but $d(k,l)$ is a continuous time function. To make it discrete, let $t=mT$, where m is an integer and T is the sampling interval. Note, to ensure there is no information loss, the value of T must be chosen according to the Nyquist sampling theorem.

Features

The main features of the DWT in a low bit-rate video coding context are as follows:

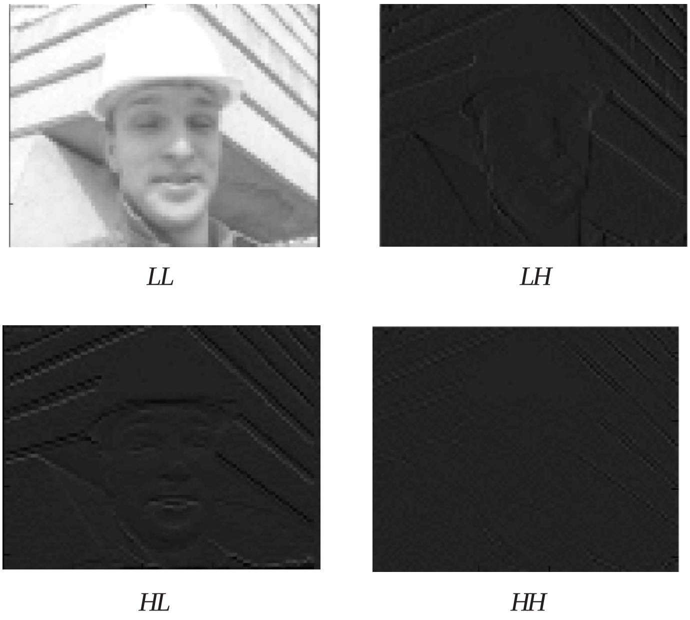
- DWT has high decorrelation and energy compaction efficiency.
- The wavelet basis functions match well with the Human Visual System (HVS) characteristics.
- Blocking artifacts and perceptual distortion are far less visible in wavelet filters due to the spatially global decomposition, resulting in subjectively better reconstructed images.
- The DWT allows multiple resolution analysis (MRA) that supports high scalability since wavelet coefficient data structures are spatially self-similar across subbands.
- The number of image pixels and DWT coefficients are the same, so there is no information is lost.
- DWT requires more memory and processing time because global decomposition requires the whole image to be considered as a large size block.
- Computational complexity is relatively high compared to DCT.
- Due to the large block size, efficient coding is often impossible, especially in VLBR, because it cannot differentiate active from static regions.

Basic Procedure

The main part of video coding is transforming spatial image data into a frequency representation. The DWT, unlike the DCT, decomposes a complete image or a large rectangular region of the image, in contrast to small block sizes (8×8, 4×4) used for DCT implementations. A single-stage DWT consists of a filtering operation that decomposes an image into four frequency bands as shown in Figure 10: “LL” — is the original image, low-pass filtered and sub-sampled by a factor of 2 in the horizontal and vertical directions. This sub-sampling may be applied repetitively; “HL”– is high-pass filtered in the vertical direction and contains residual vertical frequencies (i.e., the vertical component of the difference between the sub-sampled “LL” image and the original image); “LH”— is high-pass filtered in the horizontal direction and contains residual horizontal frequencies; “HH”— is high-pass filtered in both horizontal and vertical directions and contains residual diagonal frequencies. This DWT decomposition has a number of important properties:

- The number of DWT coefficients is the same as the number of pixels in the original image so nothing is added or removed.

Figure 10. First frame of Miss America after one level decomposition by DWT



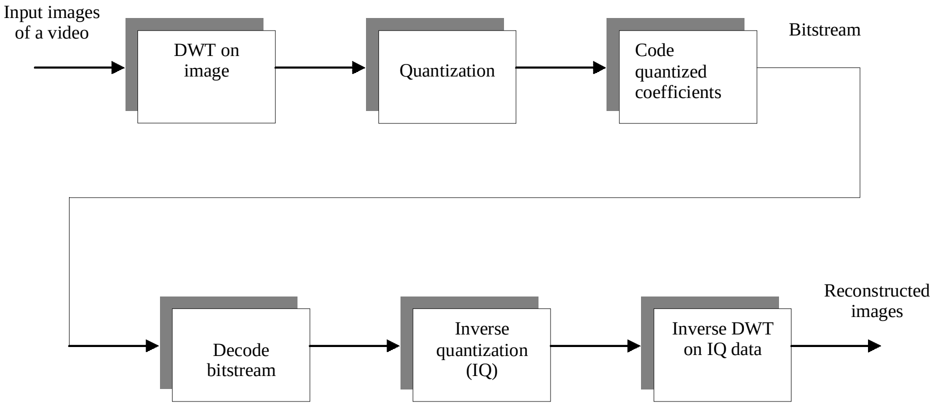
- Many of the high-frequency coefficients ("HH," "LH," and "LH") at each stage are insignificant. This reflects the fact that low frequencies carry important image information, which implies that discarding the more insignificant higher-frequency coefficients while preserving the significant ones, compresses the image efficiently.
- The DWT coefficient data structure is spatially self-similar across sub-bands (Figure 10) which leads to the important characteristic of spatial scalability.

A block diagram of the basic wavelet-based video codec is shown in Figure 11. The input images of a video sequence are used as the DWT input and the same number of wavelet coefficients is generated. To select the most significant coefficients, quantization is applied and insignificant information removed. Several algorithms have been developed to code this information in efficient way so that by sending minimum information, better image quality will be achieved at the decoder.

Evolution of DWT

DWT multilevel decomposition of an image has the property that the lowest levels correspond to the highest-frequency sub-band and finest spatial resolution, and the highest levels correspond to the lowest-frequency sub-band and the coarsest spatial resolution. Arranging the frequency sub-bands from lower- to higher-normal expectation is the energy reduction. Moreover, if the transform coefficients at a particular level have a lower energy, then coefficients at the lower levels or high-frequency sub-bands, which

Figure 11. Block diagram of video coder and decoder using wavelet

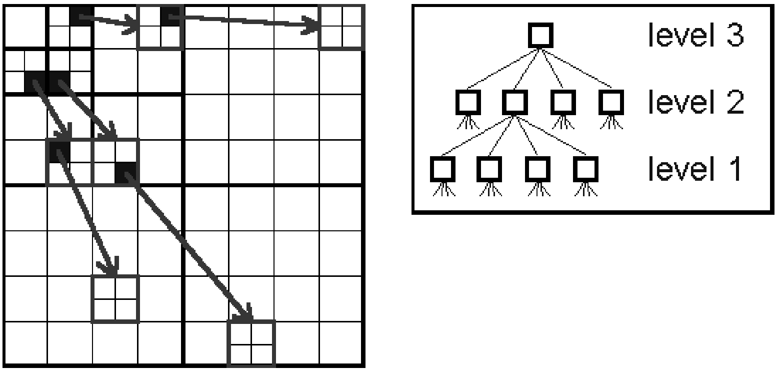


correspond to the same spatial location, would have lower energy. This self-similarity across sub-bands of wavelet coefficient data structure is a vital property in the efficient embedded-bitstream coding point of view. The advantage of embedded coding schemes is that they allow the encoding process to terminate at any point so that a target bit rate or distortion metric can be met exactly. Due to the some constraints in embedded coding, non-embedded coding for a given target bit rate or distortion requirement is sometimes preferable. The Embedded Zero-tree Wavelet (EZW) (Shapiro, 1993) and Set Partitioning in Hierarchical Trees (SPIHT) (Said & Pearlman, 1996) are two examples of embedded coding algorithms.

To exploit the properties of the wavelet transformation, a data structure called a *Zerotree* is used in embedded coding system. A zerotree is a quad-tree of which all nodes are equal to or smaller than the root. A coefficient in a low sub-band can be assumed as having four descendants in the next higher sub-band and the four descendants each also have four descendants in the next higher sub-band, i.e., every root has four leaves (Figure 12). The tree is coded with a single symbol and reconstructed by the decoder as a quad-tree filled with zeroes. It is assumed that the root has to be smaller than the threshold against which the wavelet coefficients are currently being measured. Thus, the whole tree can be coded with a single zerotree symbol. If the image is now scanned in a predefined order, from a high to a low scale, implicitly many positions are coded through the use of zerotree symbols. Of course, sometimes the zerotree rule is violated, but, in practice, the probability of this is very low. The price is the addition of the zerotree symbol to the code alphabet (Valens, 1999).

Said and Pearlman (1996) offered an alternative interpretation to the EZW algorithm to better highlight the reasons for its excellent performance. They identified three key concepts in EZW: (1) partial ordering by magnitude of the transformed coefficients with a set partitioning sorting algorithm, (2) ordered bit-plane transmission of refinement bits, and (3) exploitation of the self-similarity of the wavelet transform across different scales of an image. In addition, they presented a new and more effective implementation of the

Figure 12. Relations between wavelet coefficients in different subbands as quad-trees (Valens, 1999)



modified EZW algorithm based on Set Partitioning in Hierarchical Trees (SPIHT). They also proposed a scheme for progressive transmission of the coefficient values that incorporate the concepts of ordering the coefficients by magnitude and transmitting the most significant bits first. A uniform scalar quantizer is used, with the claim that the ordering information made this simple quantization method more efficient than expected. An efficient way to code the ordering information is also proposed that results in the SPIHT coding, in most cases, surpassing the performance obtained by the EZW algorithm (Saha, 2000).

VLBR uses embedded coding in another application, namely image browsing (Shapiro, 1993). A user is typically connected to an image database through a low bit-rate channel and finds a potentially interesting picture, but must wait to determine if the desired features are present. With a fully embedded bit stream, the same code can accommodate both the intelligibility objective required for a rejection decision and the higher-quality objective for an acceptance decision. A MC wavelet transform is generally believed to be a better approach than a non-motion compensated approach for VLBR video coding. Usually the MC is block-based using a full search, which is simple and easy to implement in hardware but is also computationally complex and produces noticeable blocking artifacts. It also has artificial discontinuities due to the noisy motion causing undesirable high-frequency sub-band coefficients, as well as an increase of side information to transmit the entropy-coded motion vectors. An adaptive multi-grid block matching with different resolutions, rather than a multi-resolution pyramid and a mesh refinement technique for variable block size to reduce the blocking artefacts is described by Ebrahimi and Dufaux (1993). It measures the true motion more accurately and provides near optimal solutions in minimizing the energy of the prediction error. It also decreases the side information while keeping the same accuracy for the motion fields by adopting a variable block size. The conventional multi-layer approach of embedded bitstream can only supply a discrete number of bitstream layers. To improve the performance, the number of layers and the quality/bit-rate level associated with each layer has to be determined at encoding time. The varying bandwidth limitations for video streaming cannot be satisfied in this manner. SNR scalability coding technique has been standardized in MPEG-4 for providing fine granularity scalability (FGS), though this added functionality is achieved with a substantial loss in compression performance (Hsiang & Woods, 2001). An embedded coding system based on 3-D sub-band filter banks combining *set partitioning* and *context modelling* is described by Hsiang and Woods (2001) that is error resilient, better than MPEG-2, and free from DCT-blocking artefacts.

The main drawback of DWT-based coding is its computational overhead. For real-time applications, there have been various attempts at improving the computational time, including an integer-based DWT (Lin, Zhang, & Zheng, 2000; Zeng & Cumming, 1998). A *leaky bucket*-approach for real-time applications has also been proposed by Balster and Zheng (2001) to control the bit rate through unpredictable transmission channels using a 3-D DWT compression system. This maintains maximum and equal frame rate for both the server and receiver, which is a fundamental requirement for real-time systems.

To exploit the intra- and inter-band correlation in the wavelet domain, EZW, SPIHT used zerotrees to implicitly classify wavelet coefficient. Zerotrees are a subset of a spatial-frequency tree consisting entirely of insignificant wavelet coefficients, where the spatial-frequency tree naturally comes from the spatial and frequency relationship in the hierarchical pyramid, as shown in Figure 12. With the identification of most insignificant coefficients using zerotree coding, the uncertainty of significant coefficients is reduced and coding efficiency is improved. Naturally, significant pixels forming from the edge of the part of the image form irregular clusters. Consequently, the insignificant parts within sub-bands are also irregular. Both the irregular significant and insignificant parts cannot be effectively exploited by the regular structured zerotree, which limits the performance of zerotree methods.

Conversely, a morphological representation of wavelet data (Servetto, Ramchandran, & Orchard, 1999), Significant-Linked connected component analysis (Chai, Vass, & Zhung, 1999), tries to exploit the intra-band correlation by forming the within-sub band irregular-shaped clusters based on the known significant coefficient. The coding result proves that irregular clusters can effectively capture the significant pixels within sub-bands. The *Embedded block coding with optimized truncation* (EBCOT) algorithm (Tauman, 2000) exploits only the intra-band correlation, being based on the independent coding and optimization of code-blocks within sub-bands. The outstanding performance of EBCOT demonstrates that strong intra-band correlation provides high-coding efficiency and is effectively achieved by explicitly classifying coefficients based on their sign and magnitude redundancy (Peng & Kieffer, 2002).

Shaped adaptive DWT (SA-DWT) is a technique for efficient coding of arbitrary-shaped visual objects that is important in object-oriented multimedia applications. The main features of SA-DWT are: (1) the number of coefficients after SA-DWT is identical to the number of pixels in the original object, (2) the spatial correlation, locality properties of wavelet transforms, and self-similarity across sub-bands are well preserved, and (3) for a rectangular region, SA-DWT is identical to the conventional DWT (Li & Li, 2000).

Future Trends

As all images are different in wavelet-based coding, the wavelet filter should be chosen adaptively, depending on the statistical nature of the image being coded. Wavelet filters are highly image-dependent with no single wavelet filter consistently performing better than others on all images. Similar results have also been observed in the context of lossless compression using various integer-to-integer wavelet transforms (Saha & Vemuri, 1999). This adaptive filter selection is important because, when the performance of the wavelet filter is initially poor, use of sophisticated quantization and context modelling of transform coefficients may not always provide significant compensation for this. Dynamically determining the most appropriate wavelet filter based on the type and statistical nature of the input image to be coded may well prove to be a fruitful research topic.

Summary

The main advantages of DWT- over DCT-based video coding are the absence of blocking artefacts and greater compatibility with the human visual system. However, it suffers from relatively high computational complexity and poor VLBR video coding efficiency in the bits distribution for active and non-active regions. 3-D wavelet coding may well be an alternative to DCT-based coding.

DCT Manipulation-Based Coding

Definition

The DCT plays an extremely important role in most image and video coding standards. It was established by Ahmed, Nararajan, and Rao (1974) as a variant of the Discrete Fourier Transform (DFT). The basic idea behind these transformations is that any periodic function can be decomposed into a finite number of sine and cosine functions with frequencies being multiples of the data window length *N*. The DCT of a discrete function *f*(*x*,*y*), *x*, *y* = 0,1,...,*N*–1 is defined as

$$F(u,v) =$$

$$\left(\frac{2}{N}\right)k_u k_v \sum\sum f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\times\cos\left(\frac{(2x+1)v\pi}{2N}\right) \quad u,v = 0,1,\ldots,N-1 \tag{11}$$

where

$$k_u k_v = \frac{1}{\sqrt{2}} \qquad \text{for } u, v=0$$

$$= 1 \qquad \text{for } u, v=1,2\ldots N\text{-}1.$$

The DCT transforms the discrete block (usually 8×8) of spatial image to produce a block of transform coefficients. The performance of a block-based transform for image compression depends on how well it decorrelates the information in each block. In the example in Figure 13, the original 8´8 block has its energy distributed across all 256 samples, while the DCT representation is much more compacted (closely correlated). In

Figure 13. Original block of image (a) (i.e., Frame #1 block 5,5 of Miss America) and (b) corresponding DCT

41	39	54	100	126	122	125	131
44	41	54	97	124	122	128	134
46	42	53	94	121	123	131	139
43	41	52	91	119	124	134	140
43	42	53	90	118	124	134	139
56	54	59	90	114	123	134	138
79	72	68	89	110	120	134	139
98	87	76	89	106	117	134	141

(a)

768	-263	-19	33	39	0	-13	0
-31	-39	-43	0	14	0	0	0
19	33	12	0	0	0	0	0
-11	-11	-13	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	-1
0	0	0	0	1	0	0	0

(b)

the DCT coefficient block, the energy is concentrated into a few significant coefficients (top left of the array), which are the low-frequency components, representing the gradual changes of brightness (luminance) in the original block (Figure 13). The bottom-right portion of the array comprises high-frequency components, and these represent rapid changes in brightness. The coefficients are, therefore, decorrelated, so smaller-valued coefficients may be discarded (for example, by quantization and thresholding) without significantly affecting the quality of the reconstructed image block at the decoder. This makes the DCT a powerful tool for image and video compression.

Features of DCT

The main features of the DCT are as follows:

- DCT has a high decorrelation capability and energy-compaction efficiency.
- DCT is relatively simple for hardware implementation and less computationally expensive than DWT.
- DCT can be applied to any block size.
- This transformation is very well established and is the basis for many of today's digital video coding standards including MPEG-X and H.26X.
- With the exception of the optimal *Karhunen Loève Transform* (KLT) that is data-dependent, the DCT is superior to other transforms including the *Discrete Hadamard*

Transform (DHT), DWT, and DFT in terms of mean-square error, energy compaction and bit rate vs. distortion performance in the reconstructed image.

Evolution of DCT

For VLBR coding, especially videophone applications, it can be common for the coefficients of a block to be zero after motion estimation/compensation, DCT, and quantization. A method that detects all-zero DCT coefficient blocks before DCT and quantization would greatly improve the coding speed (Xuan, Zhenghua, & Songyu, 1998). In H.263, N= 8, so (11) gives

$$\left|F(u,v)\right| < \frac{1}{4}\sum_{x=0}^{7}\sum_{y=0}^{7} abs(f(x,y)) \tag{12}$$

The condition for all-zero DCT coefficients is

$$\left|F(u,v)\right| < 2Q \tag{13}$$

where u,v=0,1,…, 7, and Q denotes the quantization level. Thus

$$\sum_{x=0}^{7}\sum_{y=0}^{7} abs(f(x,y)) < 8Q \tag{14}$$

This inequality provides the conditions under which all DCT coefficients are zero. Using (14), ~40% of blocks in the *Miss America* and *Claire* sequences can be so classified and, since the technique uses a Sum of Absolute Difference (SAD) as a criterion, no additional computation is involved. Moreover, the threshold can be adapted with the quantization level. Simulation results also reveal that 16Q provides a good trade-off between image distortion and computational overhead.

The main problem of this technique is that some all-zero blocks may not be detected. This is common when using this technique in H.263 coding of head-shoulder sequences (Jun & Yu, 2001), though it does not affect the video quality, just the coding speed. A modification was proposed by Jun and Yu (2001). $\left|F(u,v)\right| < 2Q$ is a sufficient and necessary condition for the early detection of all-zero DCT blocks, and is equivalent to $\max\left(\left|F(u,v)\right|\right)$. An alternative detection was also proposed by Jun and Yu as follows:

$$\sum_{x=0}^{7}\sum_{y=0}^{7}\left|f(x,y)\right| < \left\lceil NQ\sec^{2}\left(\frac{\pi}{2N}\right)\right\rceil . \tag{15}$$

This inequality may not limit the efficiency either, but with the efficiency guideline, it improved the efficiency of the approach of Xuan et al. (1998). This paper also proposed another detection criterion, which does not impact on video quality:

$$\max\left(\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left|f_p(x,y)\right|,\left|\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f_q(x,y)\right|\right)<\left\lceil\frac{2Q}{\alpha}\right\rceil \quad \textbf{(16)}$$

where

$$f_p(x,y)\in\{f(x,y)>0,\ f_p(x,y)=f(x,y);\ f(x,y)<0,\ f_p(x,y)=0\}$$

$$f_q(x,y)\in\{f(x,y)<0,\ f_q(x,y)=f(x,y);\ f(x,y)>0,\ f_q(x,y)=0\}$$

$$\alpha=\frac{2}{N}\cos^2\left(\frac{\pi}{2N}\right)$$

Experimental results showed that the ratio of correct detection (Budge & Baker, 1985; Cafforio & Rocca, 1983) increased by ~1-2% and ~10-20% respectively. Using this technique, ~60% of blocks in the *Miss America* and *Claire* sequences are correctly determined as all-zero DCT coefficient blocks, which improves the coding speed significantly in a VLBR coding context.

The Position-Dependent Encoding (PDE) approaches (Apostolopoulos & Lim, 1992; Apostolopoulos, Pfajfer, Jung, & Lim, 1994; Pfajfer, 1994; Reed & Lim, 1998) for coding DCT coefficients can be applied in any transform/sub-band filtering scheme for image or video compression. In the conventional approach, the amplitude values and run-lengths can be coded with separate codebooks or they can be jointly coded with a single codebook. With joint encoding, each (run-length, amplitude) pair is coded as a single event. In this way the coder exploits the correlation that exists between short run-lengths and small amplitudes. In separate codebooks, the coding approach exploits the different phenomena with different statistics. Traditionally, DCT coefficients are ordered in some manner (typically through zigzag scanning), but this does not exploit how the statistics of the run-length depend upon the starting position of the run. For example, runs beginning in the low-frequency region are typically shorter than runs beginning in the mid- to high-frequency regions. Moreover, the traditional way does not exploit how the statistics of the coefficient amplitude depend on the particular coefficients to be encoded. For example, in an intra-block, the low-frequency coefficients typically have larger amplitude than mid- or high-frequency coefficients. On the other hand, for inter-blocks, the statistics are usually more uniformly spread.

This idea was further extended by Apostolopouls et al. (1994) by using multiple codebooks for inter- and intra-encoded regions of a video signal. The PDE approach exploited the difference in statistics and range of the statistics by using different Huffman codebooks depending on the starting position of the run-lengths and statistics of the amplitudes. A total of 94 codebooks were used for run-length coding: 31 for intra-luminance components (Y) run-lengths, 15 for intra-chrominance components (UV) run-lengths, 32 for inter-Y run-lengths, and 16 for inter-UV run-lengths. A total of 14 codebooks were used for amplitudes coding: 3 for each intra-Y and intra-UV amplitude coding, 3 for each intra-Y and intra-UV amplitude, and 4 for each inter-Y and inter-UV amplitude. The Huffman codebooks were trained based on a weighted sum of the collected data.

Figure 14. Visual quantization matrix for QCIF image

1.000	0.9962	0.9850	0.9668	0.9423	0.9122	0.8775	0.8393
0.9944	0.9906	0.9795	0.9615	0.9372	0.9073	0.8730	0.8351
0.9778	0.9741	0.9633	0.9458	0.9221	0.8931	0.8596	0.8227
0.9511	0.9476	0.9373	0.9206	0.8979	0.8701	0.8381	0.8026
0.9158	0.9125	0.9028	0.8871	0.8658	0.8396	0.8094	0.7759
0.8734	0.8704	0.8615	0.8469	0.8272	0.8029	0.7748	0.7437
0.8260	0.8232	0.8151	0.8018	0.7837	0.7615	0.7358	0.7072
0.7752	0.7727	0.7654	0.7534	0.7371	0.7171	0.6937	0.6678

The results revealed an average reduction of 6.1% in the required bit rate for PDE compared to the single-codebook approach. In comparing PDE with MPEG's joint encoding, the tests showed that PDE performed approximately 5.8% better; however, an accurate comparison between PDE and joint encoding cannot be made for two reasons: (1) the PDE scheme was optimized for encoding certain sequences, while the joint encoding scheme was not, and (2) escape codes were not implemented in the PDE scheme, while they were part of the joint scheme. Escape codes are important because, aside from reducing the implementation complexity, they also have the beneficial effect of increasing the robustness of an entropy-coding system. In addition, the joint codebooks were designed by MPEG for I, P, and B frames, while this approach performed tests only with I and P frames.

Reed and Lim (1998) overcame these limitations by using Joint PDE. They observed that the range of possible run-lengths varies with the starting position of the run. More clearly, the range of the run-lengths decreases as the starting position of a non-zero quantized DCT coefficients increases within a block. Since MPEG-2 uses 8×8 block sizes, the run- length range starting at position (0, 0) within a block (DC) is 0 to 63, so 6 bits are needed to represent all run-length possibilities. However, traversing the block starting at DC and following some specified scanning pattern, the number of coefficients remaining in the block decreases, thus decreasing the run-length range. Therefore, using 6 bits to represent the run- lengths for non-zero mid- and high-frequency coefficients is unnecessary. Reed and Lim compared results with MPEG-2 by considering I, P, and B frames, as well as escape codes. The Joint PDE decreases bits by 7.6% on average compared with MPEG-2 by considering 31 2-D Huffman codebooks. Its performance is also better than the PDE, which only achieved a bit-reduction of about 5.8%. In VLBR coding, computing the motion vectors occupies a large percentage of the bit rate and, therefore, the overall performance of the PDE and Joint PDE will diminish.

Any block-based video coding standard such as MPEG-X, H.26X suffers from blocking effects when operating at very low bit rates. To improve the subjective quality of the coded images, the DCT coefficients are weighted using a visual quantization matrix (Figure 14) before quantization which increases the PSNR and more importantly reduces annoying blocking effects in the perceived visual quality (Ngan, Chai, & Millin, 1996).

Both visual and normal quantization is applied in DCT coefficients to reduce the bit rate of video transmission or storage. Another way to reduce the bit rate is DCT-Zone coding (Ngamwitthayanon, Ratanasanya, & Amornraksa, 2002). Normally, a zigzag scan order is used to code the DCT coefficients from top-left to bottom-right because the human eye is least sensitive to the changes in mid-high frequency regions. Some DCT coefficients within these regions will be omitted for the entropy-coding schemes for VLBR coding using visual and normal quantization. However, the absence of high-frequency DCT coefficients directly affects the sharpness of the resultant images. An investigation by Ngamwitthayanon et al. showed that omitting mid frequencies results in significant improvements in term of output bit rate while only slightly degrading the quality of resultant image, which is far less perceptually noticeable by the human visual system.

Blocking artefacts are the main concern in block-based video coding and to reduce such artefacts, post filtering processes are recommended in the new H.264 standard (ITU-T, 2003). There are some simple algorithms (Chou, Crouse, & Ramchandran, 1998; Meier, Ngan, & Crebbin, 1998), while Luo and Ward (2003) used original pixel level, correlation between neighboring blocks, and an edge-preserving, smoothing filter to reduce the artefacts.

Traditional MC techniques exploit only two frames of temporal redundancies in a video sequence, but 3-D DCT-based coding offers greater potential (Chan & Siu, 1997; Lai & Guan, 2002; Tai, Wu, & Lin, 2000). Variable temporal length 3-D DCT coding is particularly suitable for complex video scenes.

Future Trends and Summary

The DCT is presently the most widely used transform in image and video compression algorithms based on MPEG-X and H.26X standards. Its popularity depends on its data-compaction performance close to the optimal KLT and its relatively low computational complexity. In addition, unlike the KLT, the DCT is not data dependent and its transform matrix exhibits symmetries that can be exploited to obtain very efficient hardware and software implementations (Rao & Yip, 1990). Computing the DCT and the subsequent quantization of the transform coefficients are the most demanding steps of any DCT-based video encoder after ME. Docef. Kossentini, Nguuyen-Phi, and Ismaeil (2002) reduced the computational time significantly by embedding the quantization in DCT and introducing a multiplier-less integer implementation of the quantized DCT with early termination at zero values. A multiplier-less integer DCT is used in the new H.264 standard, which also enables variable-sized DCT blocks such as 4×4 instead of the usual 8×8. Blocking artefacts remain a major concern of DCT-based coding. Much research has been done in an attempt to reduce or eliminate its effects, but there is still scope for further research.

Pattern-Based Coding (PBC)

Definition

Reducing the transmission bit-rate while concomitantly retaining image quality continues to be the major challenge for efficient very low bit-rate video compression standards, such as H.26X, MPEG-X. However, these standards are still unable to encode moving objects within a 16×16 pixel *macroblock* (MB) during ME, resulting in all 256 residual error values being transmitted for MC, regardless of whether there are moving objects or not. Pattern-based coding (PBC) has the ability to code a video sequence in very low bit rate. Let $Ck(x,y)$ and $Rk(x,y)$ denote the kth MB of the current and reference frames, each of size W pixels $\times$ H lines, respectively of a video sequence, where $0 \leq x,y \leq 15$ and $0 \leq k < W/16 \times H/16$. The moving region $M_k(x,y)$ of the kth MB in the current frame is obtained as follows:

$$M_k(x,y) = T(|C_k(x,y) \bullet B - R_k(x,y) \bullet B|) \qquad \textbf{(17)}$$

where B is a 3×3 unit matrix for the morphological closing operation $\bullet$ (Gonzalez & Woods, 1992), which is applied to reduce noise, and the thresholding function $T(v) = 1$ if $v > 2$ and 0 otherwise. On the basis of total number '1' in M_k , all MBs are classified into several MB classes. An MB containing a moderate number of '1' is a partial-motion MB. A number of templates (i.e., predefined or extracted patterns) are used to code partial-motion MBs and the full-motion MBs and no-or-little- motion MBs are treated as in the H.26X standard. The partial-motion MB can then be coded using the m pixels (typically $\mu = \{64, 128, 192\}$) of that pattern with the remaining 256-μ pixels being skipped as *static background*. Therefore, successful pattern matching can theoretically have a maximum compression ratio of 4:1 for any MB. The actual achievable compression ratio will be lower, due to the computing overheads for handling an additional MB type, the pattern-identification numbering, and pattern-matching errors.

Features of PBC

The main features of pattern-based video coding are as follows:

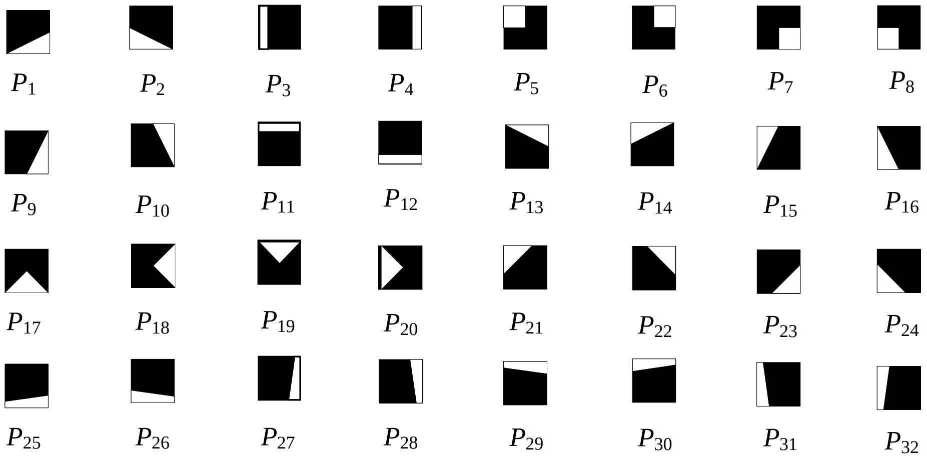
- PBC uses predefined or extracted patterns for the coding of partial-motion MBs.
- Pattern size is less than the size of an MB, so the ME computational time requirement is less than that of H.26X.
- Due to the better management of a motion area, the coding performance of PBC in VLBR is better than H.26X.
- PBC is a simplified form of object-based coding.
- PBC can code up to 20% less bit rate in same image quality and improve the image quality by up to 2dB for compatible bit rates when compared to H.263 standard.
- PBC is suitable for smooth but not high-motion video sequences.

Evolution of PBC

Fukuhara et al. (1997) first introduced the pattern concept in ME and MC by using four MB-partitioning patterns, each comprising 128-pixels. ME and MC were applied to all eight possible 128-pixel partitions of an MB, and the pattern with the smallest prediction error selected. Having only four patterns, however, meant it was insufficient to represent moving objects. With a sufficient number of blocks, the shape of a moving object can be more accurately represented, but this carries a higher processing expenditure (Wong, Lam, & Siu, 2001).

The MPEG-4 (ISO/IEC, 1998) video standard introduced the concept of content-based coding by dividing video frames into separate segments comprising a background and one or more moving objects. To address the limitations of the above approach (Fukuhara et al., 1997), Wong et al. (2001), in their *Fixed-8* algorithm, exploited the idea of partitioning the MBs via a simplified segmentation process that again avoided handling the exact shape of moving objects, so that popular MB-based motion estimation techniques could be applied. Wong et al. classified each MB into three distinct categories: (1) *Static MB* (SMB): MBs that contain little or no motion; (2) *Active MB* (AMB): MBs that contain moving object(s) with little static background; and (3) *Active-Region MB* (RMB): MBs that contain both static background and part(s) of moving object(s). SMBs and AMBs are treated in exactly the same way as in H.263. For RMB coding, Wong et al. assumed that the moving parts of an object may be represented by one of the eight predefined patterns, P_1–P_8 in Figure 15. An MB is classified as RMB if, by using some *similarity* measure, the part of a moving object of an MB is well covered by a particular pattern. The RMB can then be coded using the 64 pixels of that pattern with the remaining 192 pixels being skipped as *static background*. Therefore, successful pattern matching (e.g., pattern size 64) can theoretically have a maximum compression ratio of 4:1 for any MB. The actual achievable compression ratio will be lower due to the

Figure 15. Pattern codebook of 32 regular-shaped, 64-pixel patterns, defined in 16×16 blocks, where the white region represents 1 (motion) and black region represents 0 (no motion)



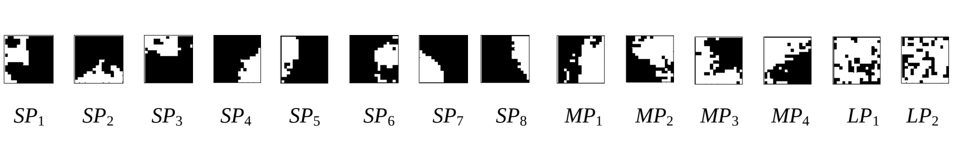
computing overheads for handling an additional MB type, the pattern-identification numbering, and pattern-matching errors. Overall, this affords superior prediction and compression efficiency and reduces the encoding time compared to H.263 by between 8% and 53% for smooth-motion sequences.

In implementing the above categories, a MB is considered as a *candidate RMB* (CRMB) if at least one of the four 8×8 quadrants does not possess moving pixels (Wong et al., 2001). This classification may, in certain instances, reduce the number of RMBs by misclassifying a possible CRMB as an AMB, where only a few moving pixels exist in another quadrant. Conversely, it may also increase the computational complexity by misclassifying an AMB as a CRMB where all but one quadrant has many moving pixels. Ultimately, a CRMB is classified as an RMB depending on a similarity measure with the patterns in the codebook. Paul, Murshed and Dooley (2002) introduced a new efficient parametric ($\delta \in \{64,96,128\}$) MB classification definition, where δ is the total number of moving pixels in a MB, without regard to any 8x8 quadrant. The pattern-based coding exhibits the best performance when $\delta = 128$ using half-pel accuracy.

Using only eight patterns for all video types results in potentially many RMBs being neglected, as moving regions vary widely between sequences. If the codebook size is extended, the number of RMBs will increase so that for a Fixed-λ algorithm, the image quality improves as the residual error is reduced, though there will be a corresponding increase in the number of pattern-identification bits for each RMB. The rationale for extending the PC size and then selecting a subset of patterns based on video content is that in the algorithm (Wong et al., 2001), it was observed that the coding efficiency with eight patterns (Fixed-8 algorithm) was superior to using only the first four patterns. Paul et al. (2002) observed a similar trend, but with diminishing returns when the PC is extended to 32 patterns (Figure 15).

To counter the diminishing improvement in coding efficiency caused by the increased number of bits to identify each of the 32-patterns, only the λ (< 32) best-matched patterns are used. The pattern-set selection process is, however, not straightforward, since in locating the best λ pattern set, it is not sufficient to simply select the patterns with the highest frequencies. All the RMBs that were initially matched against a pattern and subsequently discarded need to be considered again for matching. Some of these patterns may no longer be classified as RMBs, and the frequencies of the patterns may also change. In certain cases, this change will lead to a different ranking in the best pattern set. In Paul et al. (2002a), a new *variable pattern selection* (VPS) algorithm was developed to select the λ best-matched pattern set using a *preferential selection* approach

Figure 16. 64-pixel 8 small patterns (SP), 128-pixel 4 medium patterns (MP), and 192-pixel 2 large patterns (LP) are extracted using ASPS-3 algorithm from Miss America, *where the white region represents 1 (motion) and black region represents 0 (no motion)*



analogous to the Australian Preferential Voting System (Australian Government, 2000) where the pattern with the lowest frequency is eliminated and its preferences redistributed to the remainder. A second preferential selection algorithm called *extended VPS* (EVPS) (Paul et al., 2002b) was presented that afforded significant processing speed improvements, reducing the computational complexity by a factor as great as 9, while maintaining similar quality and compression efficiency to VPS. An optimal subset selection from a large number of patterns is a NP-complete problem (Paul et al., 2003b). A near-optimal pattern set selection algorithm was proposed in Paul et al. (2003b) that achieved the optimum result with a probability of 0.99. All these various algorithms performed better than H.263 but required pre-processing steps to select the best subset of patterns.

Measuring the similarity between a CRMB and all the patterns in the codebook on a piecewise-pixel basis is very computationally expensive, especially when the codebook size is large, which is always desirable for better coding efficiency. However, it can easily be observed that not all patterns are *relevant* for consideration when using the similarity measure (Paul et al., 2003a). Paul et al. (2003a) proposed a *Real-Time Pattern Selection* (RTPS) algorithm without any pre-processing steps. The RTPS is a proximity-based *pattern relevance* measure to dynamically create a smaller-sized *customized pattern codebook* (CPC) for each CRMB, by eliminating irrelevant patterns from the original codebook by considering gravitational centre. The RTPS algorithm uses a novel mechanism to control the size of the CPC within predefined bounds to adapt the computational complexity of the pattern selection process, so ensuring real-time operation. RTPS is thus able to process arbitrary-sized codebooks while this real-time constraint is upheld. Furthermore, the computational overhead of the similarity metric is reduced significantly by performing the processing on a quadrant-by-quadrant basis with the option to terminate whenever the measure exceeds a predefined threshold value.

All these various pattern-matching algorithms (Fukuhara et al., 1997; Paul et al., 2002a, 2002b, 2002c, 2003a, 2003b; Wong et al., 2001) assumed the actual moving regions of a RMB were similar to one or more of the patterns in the codebook (Figure 15). It is thus an approximation of the actual shape of an object. Another problem of these algorithms is that they failed to capture a large number of *active-region macroblocks* (RMB), especially when the object-moving region is relatively larger in a video sequence. The other problem of using predefined patterns is deciding exactly how many patterns are

Table 2. Percentage of different MB types generated by the RTPS(4) algorithm (Paul et al., 2003a)

Video sequences	SMB(%)	RMB(%)	AMB(%)
Miss America	63	20	18
Suzie	30	24	47
Mother & Daughter	46	24	30
Carphone	21	28	51
Foreman	13	26	61
Claire	78	14	9

Table 3. Percentage of bit-rate reduction for same PSNR values using the ASPS-3, EASPS, ASPS, and Fixed-8 compare to H.263 for six standard sequences

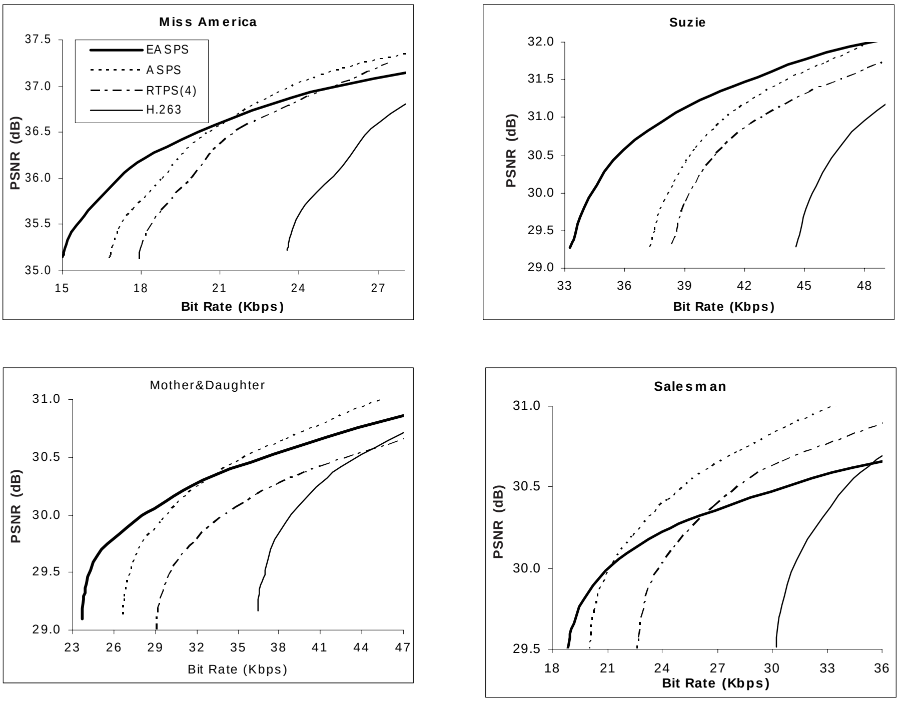
Video sequences	PSNR (dB)	Bit Rate (Kbps)			
		ASPS-3 (88)	EASPS (89)	ASPS (87)	Fixed-8 (129)
Miss America	35.5	20.8	18.8	15.2	6.3
Suzie	29.5	18.9	14.1	7.3	1.2
Mother & Daughter	29.0	21.8	18.2	14.4	5.6
Car phone	29.5	12.6	9.4	7.8	1.2
Foreman	26.0	16.5	13.3	6.6	2.2
Claire	33.5	14.3	13.4	10.4	3.0

optimum for coding. A large number of codebook patterns can better approximate the moving region of RMB, but requires more bits for pattern identification, so there is an inevitable compromise when choosing the codebook size. Intuitively, a small codebook size with lower inter-pattern relations is a desirable choice for better coding efficiency. However an arbitrarily shaped region-based MC algorithm may be a better choice to address these problems. Yokoyama, Miyamoto, and Ohta (2001) applied image-segmentation techniques to acquire arbitrarily shaped regions, though its performance was not guaranteed because it used a large number of thresholds.

The first issue can be solved if the coding system uses the pattern set extracted from the content of a video sequence. Table 1 shows the percentage of MB types generated by the RTPS(4) algorithm. While SMBs contribute nothing to the bit rate, RMBs contribute nearly 25% and AMBs 75% of the overall bit rate, so any attempt to reduce the number of AMBs leads to better compression. So a larger size pattern set, together with conventional size, will solve the second issue. The third problem may be solved if inter-relations between extracted patterns are exploited. Three algorithms based on *Arbitrary-Shaped Pattern* (Paul et al., 2003c, 2003d, 2003e) were proposed that solved the above problems by first extracting dynamically different-size pattern sets from the actual video content without assuming any pre-defined shape. Subsequently, these extracted pattern sets are used to represent the different types of RMB using a similarity measure as in all other pattern-matching algorithms. In Figure 16, different-size extracted pattern sets are shown. They are almost similar to the predefined pattern set. Paul et al. (2004a, 2004b) also proposed a new similarity metric for pattern-based video coding that improved the computational coding efficiency up to 58% when compared to the H.263 digital video coding standard.

Table 2 shows the bit-rate reduction of four PBC algorithms as compared with H.263 for comparable image quality. A maximum 22% bit reduction is achievable when the best PBC algorithm is used for VLBR video coding. The rate-distortion curves for the standard video test sequences are shown in Figure 17. From the curves, it is clear that PBC is best suited for relatively low bit-rate applications, since at higher bit rates, the performance of the H.263 standard is consistently better.

Figure 17. Coding performance comparisons for standard test video sequences



H.264 Standard and PBC

The advanced video coding standard H.264, an integrated part of MPEG-4, was finalized in June 2003, and represents a new and exciting technical solution appropriate for a broad range of applications (Richardson, 2002), including broadcast over cable, satellite, cable modem, Digital Subscriber Lines (DSL), terrestrial, interactive or storage, conversational services over ISDN, Ethernet, LAN, wireless and mobile networks, video-on-demand, and multimedia messaging. A summary of the main features of H.264 are:

- Variable block-size ME and MC with 4×4 size DCT.
- Quarter-pel accuracy motion estimation is available.
- Multiple reference frames can be used.
- Integer transform approximation of the DCT.
- Two special *switching* frame types for supporting multiple bitstreams.
- Directional spatial prediction (16-mode) for intra-coding.
- In-the-loop deblocking filter.
- Arithmetic and context-adaptive entropy coding.
- Flexible MB ordering and arbitrary slice ordering.
- Data partitioning.

Table 4. Comparison of different H.26X digital video coding standards

Features	H.261	H.263	H.264
1) Bit-rate	*p*×64Kbps where *p*=1…30. Typically 64 to 384Kbps. No lower than 64 Kbps.	Support from 20 Kbps to 2 Mbps.	Support ≤ 20Kbps.
2) Block size	16×16 (DCT 8×8). One motion vector per MB.	16×16(DCT 8×8). Up to 4 motion vectors per MB.	Various sizes including 16×16, 16×8, 8×16, 16×4, 4×4 and. (DCT 4×4).
3) Loop filter	Yes	Yes	Yes, format is different
4) Frame size	QCIF, CIF	Sub-QCIF, QCIF, CIF, 4CIF, 16CIF.	Sub-QCIF, QCIF, CIF, 4CIF, 16CIF.
5)Performance ranking	2	4	5
6)Motion compensation	Integer pel	Half & integer pel	Quarter, half & integer pel
7) Coding mode	No flexibility	19 mode	more
8) Pictures	I,P	I,P,B	I, B, P and extra SP and SI mode for switching between encoded streams.
9) Coding method	VLC from fixed look up table	Arithmetic	CABAC (Context-based adaptive arithmetic coding)

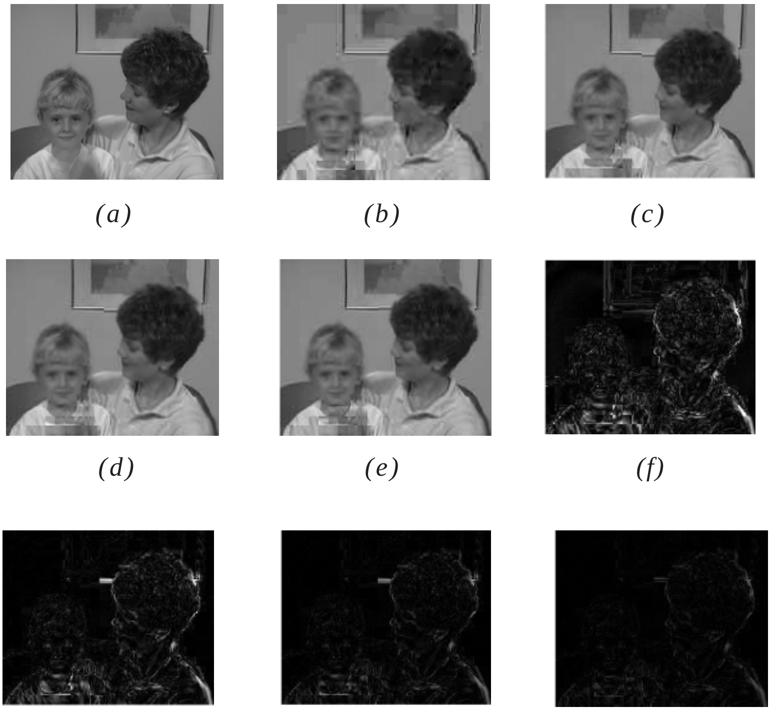
So far, PBC has not been considered as part of the H.264 standard, though variable block sizes are available for motion estimation, such as 16 × 16, 16 × 8, 8 × 8, 4 × 8 and 4 × 4 (total 8-mode). Depending upon the level of motion involved, one of these eight modes is selected, however, sometimes, this mode selection is not accurate and can be very time consuming, so PBC affords the facility to provide an additional important mode to improve the approximate motion. In Table 3, a comparison of H.261, H.263, and H.264 features helps in understanding the differences between them.

To compare the subjective performance, the original frame #13, reconstructed frames, and frame differences are presented in Figure 18 for the *Mother & Daughter* sequences. The sequence was coded at 28Kbits/frame. The gray-scale intensity of all the absolute frame- difference images has been magnified by a factor of three to provide a clearer visual comparison. The results for the reconstructed frames using ASPS can be readily perceived as better than those of the RTPS(4), the algorithm in Wong et al. (2001), and the H.263 standard.

Future Trends of PBC and Summary

PBC is the most appropriate scheme for VLBR video-coding applications. While VPS, EVPS, and other sub-optimal algorithms require some pre-defined steps in selecting the best-pattern set from a large codebook, RTPS is the only suitable candidate for real-time pattern-based video coding. However, a general conclusion in evaluating the performance of all PBC algorithms is that a strategy based on using arbitrary-shaped patterns is undoubtedly the best approach, though this is counterbalanced by the fact that the complexity currently involved in off-line pattern generation limits its application.

Figure 18. (a) Mother & Daughter frame #13; (b) to (e) Reconstructed frames using the H.263, Fixed-8, RTPS(4), and ASPS algorithms respectively; (f) to (i) Frame differences (×2) of (b), (c), (d), and (e) respectively with respect to (a)



(a) *(b)* *(c)*

(d) *(e)* *(f)*

(g) *(h)* *(i)*

Certainly, research needs to be undertaken to explore a real-time arbitrary-shaped PBC approach. Some other interesting open research questions concerning PBC are:

- Performance analysis of "hybrid" PBC by using both pre-defined regular-shaped patterns and content-based generated patterns.
- Performance analysis of different size of pre-defined regular-shaped patterns, from 2 to 192-pixels patterns, in variable block size motion estimation.
- Adaptive similarity threshold in PBC for controlling the rate-distortion ratio.
- Scalable coding by PBC using various pattern codebooks.

CONCLUSIONS

This chapter has explored a series of approaches to support effective VLBR video coding, ranging from vector quantization through to Model, Wavelet, DCT and Pattern-based coding. Efficient representation and exploitation of motion information will become more important for VLBR applications, and how we use advanced motion estimation techniques for the human visual system to estimate and encode dense motion fields for coding purposes will be the source of much on-going research. The DCT is still the most popular, mature, and widely accepted transform for most video compression standards, from very low through to high bit-rate applications. Vector Quantization will continue to play an increasingly important role in VLBR by developing a larger vector dimension and employing signal adaptive codebooks. Motion-compensated 3-D DWT will inevitably become a direct competitor of DCT — provided the computational time issue can be improved — especially as it offers the attractive features of spatial and temporal scalability. Model-based coding could also become more widely accepted for VLBR videoconferencing and videophonic applications, provided the issues of being able to automatically fit the wireframe to a speaker's face without *a priori* knowledge can be solved, along with other perceptual challenges such as how to effectively model a person wearing glasses. Object-based video coding represents a very promising option for VLBR coding, though the problems of object identification and segmentation need to be addressed by further research. Pattern-based coding is a simplified object-segmentation process that is computationally much less expensive, though a real-time, content-dependent pattern generation approach will certainly improve its acceptance for VLBR coding.

Finally, despite the large number of challenges that remain to be solved, if current video-coding research continues the rapid and inexorable rate of innovation that has been witnessed over the past decade, then the focus could well shift in the not too distant future from VLBR, to *ultra* VLBR (<8 Kbps) video-coding applications.

REFERENCES

Ahmed, N., Nararajan, T., & Rao, K. R. (1974). Discrete cosine transform. *IEEE Trans. Comput.*, 90-93.

Aizawa, K., Harashima, H., & Saito, T. (1989). Model-based analysis synthesis image coding for a person's face. *Image communication, 1*(2), 139-152.

Aizawa, K., & Huang, T. S. (1995). Model-based image coding: Advanced video coding techniques for very low bit rate applications. *IEEE proceedings*, *83*(2), 259-271.

Apostolopoulos, J., & Lim, J.S. (1992). *Position-dependent run-length encoding*. Moving Pictures Expert Group.

Apostolopoulos, J., Pfajfer, A., Jung, H.M., & Lim, J.S. (1994). Position-dependent encoding. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, (ICASSP'94), V/573 -V/576.

Australian Government Web site. (2000). *Preferential Voting System in Australia.* Retrieved July 14, 2003: *http://www.australianpolitics.com/voting/systems/preferential.shtml*

Baker, R. L. (1983). Vector quantization of digital image. PhD dissertation, Stanford, CA: Stanford University.

Baker, R.L., & Gray, R.M. (1983). Differential vector quantization of achromatic imagery. *Proceedings of the International Picture Coding Symposium*, 105-106.

Balster, E. J., & Zheng, Y. F. (2001, Aug). Real-time video rate control algorithm for a wavelet based compression scheme. *Proceedings of 44th IEEE Midwest Symposium on Circuits and Systems, 1*, 492-496.

Barnes, C. F., Rizvi, S. A., & Nasrabadi, M. (1996). Advances in residual vector quantization: A review. *IEEE Trans. on Image Processing, 5*(2), 226-262.

Bergmann, H.C. (1982). Displacement estimation based on the correlation of image segments. *IEEE Proceedings of International Conference on Electronics Image Processing*, 215-219, UK.

Biemond, J., Looijenga, L., Boekee, D.E., & Plompen, R.H. (1987). A pel recursive Weiner-based displacement estimation algorithm. *Signal Processing, 13*, 399-412.

Bozdagi, G., Tekalp, M.A., & Onural, L. (1994). 3-D motion estimation and wireframe adaptation including photometric effects for model-based coding of facial image sequences. *IEEE Trans. on Circuits and Systems for Video Technology, 4*(3), 246-256.

Buck, M., & Diehl, N. (1993). Model-based coding. In I. Sezan & R. I. Lagendijk (Eds.), *Motion analysis and image sequence processing.* Boston, MA: Kluwer.

Budge, S.E., & Baker, R.L. (1985). Compression of color digital images using vector quantization in product codes. *Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing,* 129-132.

Cafforio, C., & Rocca, F. (1983). The differential method for image motion estimation. In T.S. Huang (Ed.), *Image Sequence Processing and Dynamic Scene Analysis*, pp. 104-124. Germany.

Carlsson, S. (1988). Sketch-based coding of gray level images. *Signal Processing, 15*, 57-83.

Chai, B., Vass, J., & Zhung, X. (1999). Significance-linked connected component analysis for wavelet image coding. *IEEE Trans. Image Processing, 8*(6), 774-784.

Chan, Y.L., & Siu, W.C. (1997). Variable temporal-length 3-D discrete cosine transform coding. *IEEE Trans. on Image Processing, 6*(5), 758-763.

Chang, R.F., & Chen, W. M. (1996). Inter-frame difference quadtree edge-based side-match finite-state classified vector quantization for image sequence coding. *IEEE Trans. Circuits and Systems for Video Technology, 6*(1), 32-41.

Chen, L.H, & Lin, W.C. (1997). Visual surface segmentation from stereo. *Image and Vis. Comput., 15*, 95-106.

Chou, J., Crouse, M., & Ramchandran (1998). A simple algorithm for removing blocking artifacts in block transform coded images. *IEEE Signal Processing.*

Chowdhury, M.F., Clark, A.F., Downton, A.C., Morimastu, E., & Pearson, D.E. (1994). A switched model-based coder for video signals. *IEEE Trans. on Circuits and Systems for Video Technology, 4*(3), 216-226.

Csillag, P., & Boroczky, L. (1999). New methods to improve the accuracy of the pel-recursive Wiener-based motion estimation algorithm. *Proceedings of IEEE International Conference on Image Processing,* (ICIP '99), 3, 714-716.

Daubechies, I. (1992). *Ten lectures on wavelets*. CBMS Series. Philadelphia, PA: SIAM.

Deshpande, S.G., & Hwang, J.N. (1998). A new fast motion estimation method based on total least squares for video encoding. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, (ICASSP '98), 5, 2797-2800.

Diehl, N. (1991). Object-orientated motion estimation and segmentation in image sequences. *Signal Processing: Image Communication, 3*, 23-56.

Docef, A., Kossentini, F., Nguuyen-Phi, K., & Ismaeil, I.R. (2002). The quantized DCT and its application to DCT-based video coding. *IEEE Trans. on Image Processing, 11*(3), 177-187.

Ebrahimi, T., & Dufaux, F. (1993). Efficient hybrid coding of video for low bit rate applications. *IEEE International Conference on Communications, 1*, 522-526.

Egger, O., Reusens, E., Ebrahimi, T., & Kunt, M. (n.d.). Very low bit rate coding of visual information - A review. *Signal processing laboratory, Swiss Federal Institute of Technology at Lausanne*, Switzerland.

Forchheimer, R., & Kronander, T. (1989). Image coding: From waveforms to animation. *IEEE Trans. Acoustics, Speech, Signal Processing, 37*, 2008-2023.

Fukuhara, T., Asai, K., & Murakami, T. (1997). Very low bit-rate video coding with block partitioning and adaptive selection of two time-differential frame memories. *IEEE Trans. Circuits Syst. Video Technol., 7*, 212-220.

Fukuhara, T., & Murakami, T. (1993). 3-D motion estimation of human head for model-based image coding. *Communications, Speech and Vision, IEEE Proceedings I, 140*(1), 26-35.

Gersho, A. (1982). On the structure of vector quantizer. *IEEE Trans. Inf. Theory*, IT-28, 157-166.

Gharavi, H., & Reza-Alikhani, H. (2001). Pel-recursive motion estimation algorithm. *IEE Electronics Letters, 37*(21), 1285-1286.

Gilge, M., Englehardt, T., & Mehlan, R. (1989). Coding of arbitrarily shaped segments based on a generalized orthogonal transform. *Image communication, 1*(2), 153-180.

Goldberg, M., & Sun, H. F. (1985). Image sequence coding using vector quantization. *IEEE Trans. on Commun., 34*(7), 703-710.

Gonzalez, R.C., & Woods, R.E. (1992). *Digital image processing*. Reading, MA: Addison-Wesley.

Grossman, A., & Morlet, J. (1984). Decompositions of hard functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal, 15*(4), 723-736.

Halsall, F. (2001). *Multimedia communications: Applications, networks, protocols and standards*. Reading, MA: Addison-Wesley.

Horn, B.K.P., & Schunk, B.G. (1981). Determining optical flow. *Artificial Intelligent, 17*, 185-203.

Hou, W., & Mo, Y. (2000). Very low bit rate video coding using motion-compensated 3-D wavelet transform. *Proceedings of International Conference on Communication Technology, 2*, 1169-1172.

Hsiang, S.T., & Woods, J. W. (2001). Embedded video coding using invertible motion compensated 3-D subband/ wavelet filter bank. *Signal Processing: Image Communication, 16*(8), 705-724.

ISO/IEC 11172. MPEG-1. (1993). Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s.

ISO/IEC 13818. MPEG-2. (1995). Information technology: Generic coding of moving pictures and associated audio information: Video.

ISO/IEC 14496. MPEG-4. (1998). Information technology: Coding of audio-visual objects.

ITU-T Rec.H.261 (1993). Video CODEC for audiovisual services at *p*´ 64kbits/s.

ITU-T Rec. H.263. (1996). Video coding for low bit-rate communication.

ITU-T Rec. H.263. (1998). Video coding for low bit-rate communication. Version 2.

ITU-T Rec. H.263. (2000). Video coding for low bit-rate communication. Version 3.

ITU-T Rec. H.264/ISO/IEC 14496-10 AVC. (2003) Joint Video Team (JVT) of ISO MPEG and ITU-T VCEG, JVT-G050.

Jasinschi, R.S., & Moura, J.M.F. (1995). Content-based video sequence representation. *Proceedings of International Conference on Image Processing (ICIP'95)*, 2, 229-232.

Jun, S., & Yu, S. (2001). Efficient method for early detection of all-zero DCT coefficients. *Electronics Letters*, *37*(3), 160-161.

Kampmann, M. (2002). Automatic 3-D face model adaptation for model-based coding of video sequences. *IEEE Trans. on Circuits and Systems for Video Technology*, *12*(3), 172-182.

Karayiannis, N.B., & Li, Y. (2001). A replenishment technique for low bit-rate video compression based on wavelets and vector quantization. *IEEE Trans. on Circuits and Systems for Video Technology*, *11*(5), 658-663.

Kishino, F., & Yamashita, K. (1989). Communication with realistic sensation applied to a teleconferencing system, *IEICE Technology Rep*.

Ku, C.W., Chiu,Y.M., Chen, L.G., & Lee, Y.P. (1996). Building a pseudo object-oriented very low bit-rate video coding system from a modified optical flow motion estimation algorithm. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, (ICASSP-96), 4, 2064-2067.

Kunt, M., Ikonomopoulos, A., & Kocher, M. (1985). Second generation image coding techniques. *IEEE Proceedings*, *73*, 795-812.

Kwon, H., Venkatraman, M., & Nasrabadi, N. M. (1997). Very low bit rate video coding using variable block-size entropy-constrained residual vector quantizers. *IEEE Journal on Selected Areas in Communication*, *15*(9), 1714-1725.

Lai, T.H., & Guan, L. (2002). Video coding algorithm using 3-D DCT and vector quantization. *Proceedings of International Conference on Image Processing*, 1, 741-744.

Li, H., & Forchheimer, R. (1994). Image sequence coding at very low bitrates: A review. *IEEE Trans. on image processing*, *3*(5), 589-609.

Li, H., & Forchheimer, R. (1994). Two-view facial movement estimation. *IEEE Trans. on Circuits and Systems for Video Technology*, 4(3), 276-286.

Li, H., Roivainen, P., & Forchheimer, R. (1993). 3-D motion estimation in model-based facial image coding. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, *15*(6), 545-555.

Li, S., & Li, W. (2000). Shape-adaptive discrete wavelet transforms for arbitrary shaped visual object coding. *IEEE Trans. Circuits and Systems for Video Technology*, *10*(5), 725-743.

Li, Y.C., & Chen, Y.C. (1998). A hybrid model-based image coding system for very low bit-rate coding. *IEEE Journal on Selected Areas in Communications*, *16*(1), 28-41.

Lin, C., Zhang, B., & Zheng, Y. (2000, Dec). Packet integer wavelet transform constructed by lifting scheme. *IEEE Trans. Circuits and Systems for Video Technology*, *10*(8), 1496-1501.

Linde, A., Buzo, A., & Gray, R.M. (1980). An algorithm for vector quantizer design. *IEEE Trans. Commun.*, COM-28, 84-95.

Liu, H., Chellappa, R., & Rosenfeld, A. (2003). Accurate dense optical flow estimation using adaptive structure tensors and a parametric model. *IEEE Trans. on Image Processing*, *12*, 1170-1180.

Luo, Y., & Ward, R. K. (2003). Removing the blocking artifacts of block-based DCT compressed images. *IEEE Trans. on Image Processing*, *12*(7), 838-842.

Man, H., Queiroz, R.L., & Smith, M.J.T. (2002). Three-dimensional subband coding techniques for wireless video communications. *IEEE Trans. Circuits and Systems for Video Technology*, *12*(6), 386-397.

Maragos, P. (1987). Tutorial on advances in morphological image processing and analysis. *Opt. Eng.*, *26*(7), 623-632.

Meier, T., Ngan, K., & Crebbin, G. (1998). Reduction of coding artifacts at low bit rates. *Proceedings of SPIE Visual Communications Image Processing.*

Musmann, H.G., Hotter, M., & Ostermann, J. (1989). Object-oriented analysis-synthesis coding of moving images. *Signal processing: Image communication*, *1*(2), 117-138.

Nakaya, Y., Chuah, Y.C., & Harashima, H. (1991). Model-based/waveform hybrid coding for videotelephone images. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing* (ICASSP-91), *4*, 2741-2744.

Nakaya, Y., & Harashima, H. (1994). Motion compensation based on spatial transformation. *IEEE Trans. on Circuits and Systems for Video Technology*, *4*(3), 339-356.

Nasrabadi, N.M., & King R.A. (1988). Image coding using vector quantization: A review. *IEEE Trans. on Communications*, *36*(8), 957-971.

Netravali, A.N., & Robbins, J.D. (1979). Motion-compensated television coding. *Bell System Technology Journal*, Part I, *58*(3), 631-670.

Ngamwitthayanon, N., Ratanasanya, S., & Amornraksa, T. (2002). Zone coding of DCT coefficients for very low bit rate video coding. *Proceedings of IEEE International Conference on Industrial Technology*, 2, 769-773.

Ngan, K.N., Chai, D., & Millin, A. (1996). Very low bit rate video coding using H.263 coder. *IEEE Trans. on Circuits and Systems for Video Technology*, *6*(3), 308-312.

Ong, E.P., Tye, B.J., Lin, W.S., & Etoh, M. (2002). An efficient video object segmentation scheme. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, (ICASSP'02), 4, 3361-3364, May.

Pan, J.S., Lu, Z.M., & Sun, S. H. (2003). An efficient encoding algorithm for vector quantization based subvector technique. *IEEE Trans. on Image Processing*, *12*(3), 265-270.

Parke, F. I. (1982). Parameterised models for facial expressions. *IEEE Computer Graphics*, *12*.

Paul, M., Murshed, M., & Dooley, L. (2002a). A Low Bit-Rate Video-Coding Algorithm Based Upon Variable Pattern Selection. *Proceedings of 6th IEEE International Conference on Signal Processing* (ICSP-02), Beijing, 2, 933-936.

Paul, M., Murshed, M., & Dooley, L. (2002b). A variable pattern selection algorithm with improved pattern selection technique for low bit-rate video-coding focusing on moving objects. *Proceedings of International Workshop on Knowledge Management Technique* (IKOMAT-02), Crema, Italy, 1560-1564.

Paul, M., Murshed, M., & Dooley, L. (2002c). Impact of macroblock classification on low bit rate video coding focusing on moving region. *Proceedings of International*

Conference on Computer and Information Technology (ICCIT-2002), Dhaka, Bangladesh, 465-470.

Paul, M., Murshed, M., & Dooley, L. (2003a). A new real-time pattern selection algorithm for very low bit-rate video coding focusing on moving regions. *Proceedings of IEEE International Conference of Acoustics, Speech, and Signal Processing* (ICASSP-03), Hong Kong, 3, 397-400.

Paul, M., Murshed, M., & Dooley, L. (2003b). A real time generic variable pattern selection algorithm for very low bit-rate video coding. *IEEE International Conference on Image Processing* (ICIP-03), 3, 857-860, Barcelona, Spain.

Paul, M., Murshed, M., & Dooley, L. (2003c). An arbitrary shaped pattern selection algorithm for very low bit-rate video coding focusing on moving regions. *Proceedings of 4th IEEE Pacific-Rim International Conference on Multimedia* (PCM-03), Singapore.

Paul, M., Murshed, M., & Dooley, L. (2003d). Very low bit rate video coding focusing on moving region using three-tier arbitrary shaped pattern selection algorithm. *Proceedings of International Conference on Computer and Information Technology* (ICCIT-2003), Dhaka, Bangladesh, 2, 580-585.

Paul, M., Murshed, M., & Dooley, L. (2003e). Very low bit rate video coding using an extended arbitrary shaped-pattern selection algorithm. *Proceedings of Fifth International Conference on Advances in Pattern Recognition (ICAPR-03)*, 1, 308-311. India.

Paul, M., Murshed, M., & Dooley, L. (2004a). A new efficient similarity metric generic computation strategy for pattern-based very low bit-rate video coding. *Proceedings of IEEE International Conference of Acoustics, Speech, and Signal Processing* (ICASSP-04), Canada.

Paul, M., Murshed, M., & Dooley, L. (2004b). A real-time pattern selection algorithm for very low bit-rate video coding using relevance and similarity metrics. Accepted to *IEEE Trans. on Circuits and Systems on Video Technology Journal.*

Pearson, D. (1989). Model-based image coding. In *Proceedings of the Global Telecommunications Conference* (GLOBECOM-89), 1, 554 -558.

Pearson, D. E. (1995). Developments in model-based video coding. *IEEE Proceedings, 83*(6), 892-906.

Peng, K., & Kieffer, J. (2002). Embedded image compression based on wavelet pixel classification and sorting. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP-02, 4, 3253-3256.

Pfajfer, A. (1994). Position-dependent encoding. Master's thesis, MIT, Cambridge, MA.

Platt, S. M., & Badler, N. I. (1981). Animating facial expression. *IEEE Computer Graphics. 12*, 245-252.

Ramamurthi, B., & Gersho, A. (1986). Classified vector quantization of images. *IEEE Trans. on Communication, 34*(11), 1105-1115.

Rao, K.P., & Yip, P. (1990). *Discrete cosine transformation: Algorithms, advantages, applications*. New York: Academic Press.

Reed, E.C., & Lim, J.S. (1998). Efficient coding of DCT coefficients by joint position-dependent encoding. *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP'98), May, 5, 2817-2820.

Richardson, I.E.G. (2002). *Video codec design*. New York: John Wiley & Sons.

Richardson, I.E.G. (2003). *H.264 and MPEG-4 video compression*. New York: Wiley Press.

Richardson, I.E.G. (2004). H.264/MPEG-4 Part 10: Overview. Retrieved January 14, 2004: *http://www.vcodex.fsnet.co.uk/h264.html*

Saha, S. (2000). Image compression - from DCT to wavelets: A review. Retrieved February 20, 2004: *http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html*

Saha, S., & Vemuri, R. (1999). Adaptive wavelet coding of multimedia images. *Proceedings of ACM Multimedia Conference*, November 1999.

Said, A., & Pearlman, W.A. (1996). A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.*, 243-250.

Schmidt, G. (1999). *A compatible modulation strategy for embedded digital data streams*. PhD dissertation, University of Glamorgan, Wales.

Servetto, S., Ramchandran, K., & Orchard, M. (1999). Image coding based on a morphological representation of wavelet data. *IEEE Trans. Image Processing, 8*(9), 1161-1173.

Shapiro, J. (1993). Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Process,* 3445-3462.

Shapiro, J. M. (1993). Application of the embedded wavelet hierarchical image coder to very low bit rate image coding. *Proc. of IEEE Int. Conference of Acoustics, Speech, and Signal Processing* (ICASSP-93), 5, 558-561.

Shen, G., Zeng, B., & Liou, M.L. (2003). Adaptive vector quantization with codebook updating based on locality and history. *IEEE Trans. on Image Processing, 12*(3), 283-293.

Shen, J., & Chan, W. Y. (2001). A novel code excited pel-recursive motion compensation algorithm. *IEEE Signal Processing Letters, 8*(4), 100-102.

Shi, Y.Q., & Sun, H. (1999). *Image and video compression for multimedia engineering fundamentals, algorithms, and standards*. Boca Raton, FL: CRC Press.

Shu, L., Shi , Y.Q., & Zhang, Y.Q. (1997). An optical flow based motion compensation algorithm for very low bit-rate video coding. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP-97, 4, 2869-2872.

Sui, M., Chan, Y.H., & Siu, W.C. (2001). A robust model generation technique for model based video coding. *IEEE Trans. on Circuits and Systems for Video Technology, 11*(11), 1188-1192.

Tai, S.C., Wu, Y.G., & Lin, C.W. (2000). An adaptive 3-D discrete cosine transform coder for medical image Compression. *IEEE Trans. on Information Technology in Biomedicine, 4*(3), 259-263.

Tauman, D. (2000). High performance scalable image compression with EBCOT. *IEEE Trans. Image Processing, 9*(7), 1158-1170.

Tekalp, A.M. (1995). *Digital video processing*. Englewood Cliffs, NJ: Prentice-Hall.

Terada, K., Takeuchi, M., Kobayashi, K., & Tamaru, K. (1998). Real-time low bit- rate video coding algorithm using multi-stage hierarchical vector quantization. *Proceedings of IEEE International Conference of Acoustics, Speech, and Signal Processing* (ICASSP-98), 3, 397-400.

Valens, C. (1999). EZW encoding. Retrieved January 21, 2004: *http://perso.wanadoo.fr/polyvalens/clemens/ezw/ezw.html*

Venkatraman, S., Nam, J. Y., & Rao, K. R. (1995). Image coding based on classified lapped orthogonal transform-vector quantization. *IEEE Trans. Circuits and Systems for Video Technology, 5*(4), 352-355.

Vetterli, M., & Kovacevic, J. (1995). *Wavelet and subband coding*. Englewood Cliffs, NJ: Prentice-Hall.

Walker, D.R., & Rao, K.R. (1984). Improved pel-recursive motion compensation. *IEEE Trans.Commun.*, COM-32, 1128-1134.

Wallis, R.K., Pratt, W.K., & Plotkin, M. (1981). Video conferencing at 9600 bps. *Proceedings of Picture Coding Symposium*, 104-105.

Wang, D., & Wang, L. (1997). Global motion parameters estimation using a fast and robust algorithm. *IEEE Trans. on Circuits and Systems for Video Technology, 7*(5), 823-826.

Welsh, W.J. (1987). *Model-based coding of moving images at very low bit rate.* Picture Coding Symposium.

Welsh, W.J. (1988). Model-based coding of videotelephone images using an analysis method. *Proc. Picture Coding Symp*. (PCS-88), 4-5.

Welsh, W.J. (1991). Model-based coding of videotelephone images. *Electronics & Communication Engineering Journal, 3*(1), 29-36.

Wiegand, T., Sullivan, G.J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol, 13*(7), 560-576.

Wong, K.-W., Lam, K.-M., & Siu, W.-C. (2001). An efficient low bit-rate video-coding algorithm focusing on moving regions. *IEEE Trans. Circuits and Systems for Video Technol., 11*(10), 1128-1134.

Woods, J.C. (2001). Outlier removal algorithm for model-based coded video. *Proceedings of International Conference on Image Processing,* 3, 110-113.

Xu, G., Agawa, H., Nagashima, Y., Kishino, F., & Kobayashi, Y. (1990). Three-dimensional face modelling for virtual space teleconferencing systems. *IEICE Trans., 73*(10).

Xuan, Z., Zhenghua, Y., & Songyu, Y. (1998). Method for detecting all-zero DCT coefficients ahead of discrete cosine transformation and quantisation. *Electronics Letters, 34*(19), 1839 -1840.

Yang, S.B., & Tseng, L.Y. (2001). Smooth side-match classified vector quantizer with variable block size. *IEEE Trans. on Image Processing, 10*(5), 677-685.

Yokoyama, Y., Miyamoto, Y., & Ohta, M. (1995). Very low bit-rate video coding using arbitrarily shaped region-based motion compensation. *IEEE Trans. Circuits and Systems for Video Technology, 5*(6), 500-507.

Zeng, Z., & Cumming, I. (1998). SAR image compression based on the discrete wavelet transform. *Fourth International Conference on Signal Processing.*

Section III

Video Data Security and Video Data Synchronization and Timeliness

Chapter VI

Video Biometrics

Mayank Vatsa, Indian Institute of Technology, Kanpur, India

Richa Singh, Indian Institute of Technology, Kanpur, India

P. Gupta, Indian Institute of Technology, Kanpur, India

ABSTRACT

Biometrics is a fast, user-friendly personal identification with a high level of accuracy technology. This chapter is highlighting the biometrics technologies that are based on video sequences viz. face, eye (iris/retina), and gait. The basics behind the three video-based biometrics technologies are discussed, along with a brief survey.

INTRODUCTION

"Biometrics" means "life measurement," but the term is usually associated with the use of unique physiological characteristics to identify an individual. The application that most people associate with biometrics is security. However, biometric identification has a much broader and growing relevance as computer interface becomes more natural. Knowing the person with whom you are conversing is an important part of human interaction, and you can expect computers of the future to have the same capabilities. As this trend progresses, biometrics becomes increasingly closer to everyday life.

Biometric technology, along with greater vigilance and more effective procedures, is now being touted as the vehicle to create a deterrent to those who seek to terrorize society by exploiting the weaknesses that, since the level of threat was previously considered low, have prevailed in traditional identification methods. Biometrics traits are now being used to prevent this. A number of biometric traits have been developed and used to authenticate a person's identity. Video biometrics is the most secure in today's world, as it can be used for surveillance purposes and maintaining security.

By special characteristics, we mean features such as the face, iris, fingerprint, signature, etc. This method of identification is preferred over traditional passwords and PIN-based methods for various reasons, such as:

- The person to be identified is required to be physically present at the time-of-identification.
- Identification based on biometric techniques obviates the need to remember a password or carry a token.

A biometric system is essentially a pattern-recognition system that makes a personal identification by determining the authenticity of a specific physiological or behavioral characteristic possessed by the user. Biometric technologies are thus defined as the "automated methods of identifying or authenticating the identity of a living person based on a physiological or behavioral characteristic." A biometric system can be either an identification system or a verification (authentication) system, both of which are defined below.

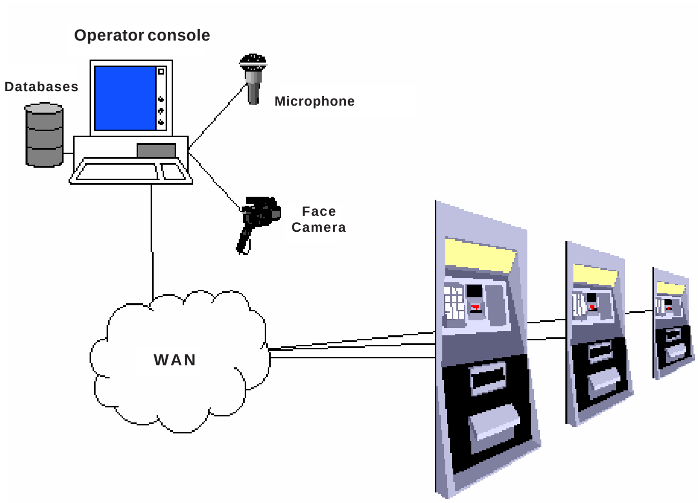
- *Identification - One to Many*: Biometrics can be used to determine a person's identity even without his or her knowledge or consent. For example, scanning a crowd with a camera and using face-recognition technology, one can determine matches against a known database.
- *Verification - One to One*: Biometrics can also be used to verify a person's identity. For example, one can grant physical access to a secure area in a building by using finger scans or can grant access to a bank account at an ATM by using retinal scan.

Biometric authentication requires the comparison of a registered or enrolled biometric sample (biometric template or identifier) against a newly captured biometric sample (for example, the one captured during a login). This is a three-step process (*Capture*, *Process*, *Enroll*) followed by a *Verification* or *Identification* process.

During the *Capture* process, a raw biometric is captured by a sensing device such as a fingerprint scanner or video camera. The second phase of processing is to extract the distinguishing characteristics from the raw biometric sample and convert them into a processed biometric identifier record (sometimes called biometric sample or biometric template). The next phase does the process of enrollment. Here the processed sample (a mathematical representation of the biometric — not the original biometric sample) is stored/registered in a storage medium for future comparison during an authentication. In many commercial applications, there is a need to store only the processed biometric sample. The original biometric sample cannot be reconstructed from this identifier.

There are many biometric characteristics that may be captured in the first phase of processing. However, automated capturing and automated comparison with previously stored data require the following properties of biometric characteristics:

Figure 1. An example of biometric system



- *Universal*. Everyone must have the attribute. The attribute must be one that is universal and seldom lost to accident or disease.
- *Invariance of properties*. They should be constant over a long period of time. The attribute should not be subject to significant differences based on age or either episodic or chronic disease.
- *Measurability*. The properties should be suitable to capture without waiting time and it must be easy to gather the attribute data passively.
- *Singularity*. Each expression of the attribute must be unique to the individual. The characteristics should have sufficient unique properties to distinguish one person from any other. Height, weight, hair and eye color are all attributes that are unique, assuming a particularly precise measure, but do not offer enough points of differentiation to be useful for more than categorizing.
- *Acceptance*. The capturing should be possible in a way acceptable to a large percentage of the population. Excluded are particularly invasive technologies, i.e., technologies that require a part of the human body to be taken or that (apparently) impair the human body.
- *Reducibility*. The captured data should be capable of being reduced to a file that is easy to handle.
- *Reliability and Tamper-resistance*. The attribute should be impractical to mask or manipulate. The process should ensure high reliability and reproducibility.
- *Privacy*. The process should not violate the privacy of the person.
- *Comparable*. It should be able to reduce the attribute to a state that makes it digitally comparable to others. The less probabilistic the matching involved, the more authoritative the identification.

- *Inimitable*. The attribute must be irreproducible by other means. The less reproducible the attribute, the more likely it will be authoritative.

Among the various biometric technologies being considered are fingerprint, facial features, hand geometry, voice recognition, iris scan, retina scan, vein patterns, palmprint, DNA, keystroke dynamics, ear shape, odor, signature verification, and gait.

Biometrics authentication is done both in still and video images. In still images, biometric features are captured, and the algorithm is applied for matching and comparison of the features, but for video images, the task is somewhat complicated. In the case of dynamic feature extraction, we locate or track a moving object in a video sequence. From a video sequence, the frame needs to be extracted and from each frame the motion estimation is done for tracking the head, face, or any desired feature. After tracking, the features are then extracted and the comparison is done. The various biometric traits that can be implemented in video for surveillance applications are face, eye (iris and retina) and gait.

People recognize one another by looking at each other's face, so it is no surprise that biometric technology can do the same. Face recognition is a noninvasive process where a portion of the subject's face is photographed and the resulting image is reduced to a digital code. Facial recognition records the spatial geometry of distinguishing features of the face. There exist several algorithms for facial recognition. For recognizing face in a video sequence, the body is to be tracked from which the face is detected and then recognition is done.

Eye scanning measures the iris pattern in the colored part of the eye. The varying blood vessel patterns in eye are used for recognition. These patterns develop from birth and are unique for every individual, even a person's left and the right eyes have different patterns. The patterns are extracted and stored in the form of digitized templates of some bytes (96 bytes for retina and 256 bytes for iris). For video images, the eye is detected from the image and from the eye the region of interest is extracted and the processing is done.

Gait recognition is another kind of biometric technology that can be used to monitor people without their cooperation. Walking speed and style make the composite signature that characterizes the overall feel of someone's walk. Some researchers are working on visually based systems that use video cameras to analyze the movements of each body part — the knee, the foot, the shoulder, and so on. It could even be used to spot people who are moving around in suspicious ways, which may include repetitive walking patterns (suggesting they're "casing out" a target) or movements that do not appear natural given their physicality. For recognizing a person using gait, the body is tracked from video sequences and the features are obtained for comparison.

This chapter thus includes a step-by-step description of the various issues and some well- known algorithms involved in the above-mentioned biometric traits. It explains how to develop any biometric system for security applications at places where video sequences are provided as input to the system.

FACE

As the old saying goes, "Seeing is believing." Vision plays a very important role in our daily life. The underlying mechanism of human vision is not clear; people can see objects and recognize them with little effort. The ability of human vision for face recognition is very robust, as people can recognize thousands of faces learned through their lifetime and identify familiar faces in a crowd even after years of separation. Also, this skill is not affected by pose, age, expression, lighting conditions, or make-up.

Machine-based face recognition is an intriguing and challenging problem for a number of reasons. Many researchers are interested in recognizing the human face and are encouraged by the wealth of applications for an automatic face-recognition system. Thus far, most systems have treated face recognition as one more biometric, suitable for security applications, for physical access control and computer logon, or for database lookup in databases of face images, such as those used by the police and passport or driving license authorities.

Face recognition in video has become widely accepted as one of the most valuable sources of information. In the context of human-oriented applications, such as security surveillance, immersive and collaborative environments, multi-media games, computer-human interactions, video-conferencing, video annotation and coding, etc., video information is examined for the presence of information about faces. The most important tasks underlying this “Face in Video” scenario are: Tracking, Face Detection, and Face Recognition.

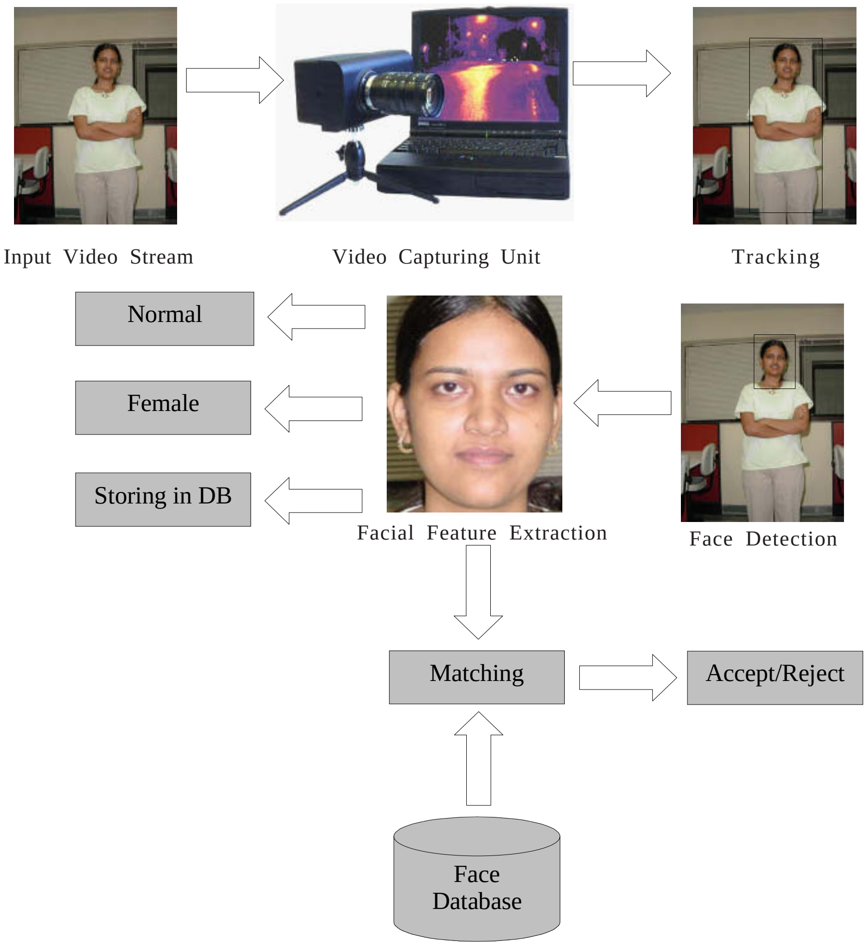
Significant research has been conducted on still face recognition (Chellappa, Wilson, & Sirohey, 1995; Zhao, Chellappa, Rosenfeld, & Phillips, 2000) where the probe is the still image. An abstract representation of an image after a suitable geometric and photometric registration is formed using the various algorithms, and then recognition is preformed based on the new representation. However, the research efforts using video as the probe are relatively fewer because of low-quality video sequences, less illumination and pose variation, uncertainty in selection of good frames, and the small size of face images that makes the characteristics of human face difficult to extract. But video-based face recognition offers several advantages over still image-based face recognition:

- Video provides abundant image data; we can select good frames on which to perform classification.
- Video provides temporal continuity; this allows reuse of classification information obtained from high-quality frames in processing low-quality frames.
- Video allows tracking of face images; hence, phenomena such as facial expressions and pose changes can be compensated for, resulting in improved recognition.

Figure 2 shows the system architecture of a generic face-recognition system for video sequences. The input is a video sequence that is captured using a video camera attached to the workstation with the help of frame grabber. This input is first sent to the tracking module to track the human body. The output of this module is then used by the face-detection module where the face is localized, and the output of this module is the face-detected image. In the next phase, the features are extracted and, based on those features, a matching algorithm can give the facial expression, gender, and authentication results according to the maintained database.

Thus, in a face-recognition system for video sequences, there are three main tasks: Tracking, Face Detection, and Recognition (Matching). There are several algorithms available for these three tasks. The following subsections discuss these tasks individu-

Figure 2. Generic face recognition system for video sequences



ally, and finally, some of the algorithms for face recognition in video sequences are explained.

Tracking

"Looking at people" — generally called tracking — covers the whole body tracking in video sequences. This is the first step towards designing a face-recognition system for video environment. There are various algorithms available for solving the tracking problem. These algorithms can mainly be classified into three categories:

- Algorithms based on a 2-D environment without explicit shape models;
- Algorithms based on a 2-D environment with explicit shape models; and
- Algorithms based on a 3-D environment.

However, these categories can overlap each other in some cases. The most general algorithm for tracking can be implemented by completely bypassing a pose-recovery step and by describing human movement in terms of simple, low-level, 2-D features from region of interest. Algorithms based on 2-D information without explicit shape models are generally based on background image subtraction, skin-color detection followed by morphological operations, template matching, and algorithms based on shape and motion information. Some algorithms are based on detection of a statistical shape model and track the contour of persons. Such types of algorithms are based on snakes (Kass, Witkin, & Terzopoulos, 1987) and active shape models (Cootes & Taylor, 1992).

There are some algorithms based on motion segmentation. Velocity information is also in use that can track persons quite accurately. Skin color segmentation and feature tracking are also used to track the person in videos (Yang, Kriegman, & Ahuja, 2002). There are some works that explicitly use *a priori* knowledge of how the human body appears in 2-D, taking essentially a model-and view-based algorithms to segment, track, and label body parts. The models or knowledge used are typically stick figures and/or wrapped around with blobs. Some algorithms use hierarchical and articulated curve-fitting algorithms to perform the task, whereas some others use segmentation over time, region-based algorithms, shape-color models, and intensity-edge- depth- motion cues to detect people in video.

The algorithms based on 3-D information take the advantage of largely available *a priori* knowledge about the kinematics and shape properties of the human body to make the problem tractable. Tracking in 3-D can also handle events such as occlusion and collision. There are four main components in 3-D tracking algorithms: prediction, synthesis, image analysis and state estimation. The prediction component predicts the behavior for next step and incorporates semantic knowledge into the tracking process. The synthesis component acts as a translator in between the prediction and image levels, which allows the image analysis component to selectively focus on a subset of regions and look for a subset of features. The state-estimation component computes the new state using the segmented image. This is the most general framework that can be applied for both 2-D and 3-D tracking. In places where motion is also one of the main concerns, 3-D tracking is used. A detailed survey of the tracking algorithms can be found in Gavrila (1999).

Face Detection

Face detection in an arbitrary image is used to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face. Face detection is the first step in the face-recognition system used for security purpose. The face recognition could not be done properly if a face is not detected correctly. Thus, a reliable face- recognition algorithm is necessary for face recognition. Face detection from a single image or from image sequence is a very challenging task, because it involves locating faces with no prior knowledge about the scales, locations, and orientations (upright, rotated) with or without occlusions, with different poses (frontal, profile), etc. Facial expressions and lighting conditions can also change the overall appearance of faces, thereby making it difficult to detect them. The appearance of human faces in an image depends on the poses of humans and the viewpoints of the acquisition devices. The challenges associated with face detection can be attributed to the following factors:

- *Pose.* The images of a face vary due to the relative camera-face pose (frontal, 45 degree, profile, upside down), and some facial features such as an eye or the nose may become partially or wholly occluded.
- *Presence or absence of structural components.* Facial features such as beards, mustaches, and glasses may or may not be present, and there is a great deal of variability among these components, including shape, color, and size.
- *Facial expression.* The appearance of faces is directly affected by a person's facial expression.
- *Occlusion.* Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.
- *Image orientation.* Face images vary for different rotations about the camera's optical axis.
- *Imaging conditions.* When the image is formed, factors such as lighting and camera characteristics affect the appearance of a face.

There are various problems related to face detection. *Face localization* aims to determine the image position of a single face; this is a simplified detection problem with the assumption that an input image contains only one face (Moghaddam & Pentland, 1997; Yam & Lam, 1998). The goal of *facial feature detection* is to detect the presence and location of features, such as eyes, nose, nostrils, eyebrows, mouth, lips, ears, etc., with the assumption that there is only one face in an image (Craw, Tock, & Bennett, 1992; GrafChen, Petajan, & Cosatto, 1995).

Here the various existing algorithms for face detection are discussed. Algorithms to be used also depend upon the images available. For color images, the detection algorithm based on skin color is applied, while for real time video type of applications, algorithms involving the motion are considered that is a moving face another type of algorithm that is used is for unconstrained scenes such as a given black and white still image. For this the image features are extracted using various algorithms like genetic algorithms, neural networks, templates, shape, support vector machines, feature based. There are even several disadvantages with different algorithms. One disadvantage with the algorithm based on color images is that it does not work on all kinds of skin colors, and it is not very robust under varying lighting conditions. Face detection by motion generally works, based on the algorithm that in reality a face is almost always moving, so the manipulations can be done by calculating the moving area. However, a problem arises if there are other objects in the image background that are moving. Some new algorithms can be implemented by merging two algorithms; a better result can be obtained by combining several of them. Surveys discussing various algorithms to face detection are available in Samal and Iyengar (1992) and Yang, Kriegman, and Ahuja (2002).

Face Recognition

Humans can easily recognize a known face in various conditions and representations. This remarkable ability of humans to recognize faces with large intra-subject variations has inspired vision researchers to develop automated systems for face recognition. However, the current state-of-the-art machine vision systems can recognize faces only in a constrained environment. Note that there are two types of face comparison scenarios, (1) face *verification* (*or authentication*) and (2) face *identification* (*or recognition*). Face verification involves a one-to-one match that compares a query face image against a template face image whose identity is being claimed; face identification involves one-to-many matches that compare a query face image against all the template images in a face database to determine the identity of the query face. The main challenge in vision-based face recognition is the presence of a high degree of variability in human face images. There can be potentially very large intra-subject variations (due to 3D head pose, lighting, facial expression, facial hair, and aging) and rather small inter-subject variations (due to the similarity of individual appearances). Currently available vision-based recognition techniques can mainly be categorized into two groups based on the face representation that they use: (1) appearance-based, which uses holistic texture features, and (2) geometry-based, which uses geometrical features of the face.

It can be stated that external and internal facial components and distinctiveness, configuration, and local texture of facial components all contribute to the process of face recognition. Humans can *seamlessly blend* and *independently perform* appearance-based and geometry-based recognition approaches efficiently. Therefore, we believe that merging the holistic texture features and the geometrical features (especially at a semantic level) is a promising method to represent faces for recognition. While we focus on the 3D variations in faces, we should also take the temporal (aging) factor into consideration while designing face recognition systems. In addition to large intra-subject variations, another difficulty in recognizing faces lies in the small inter-subject variations. Different persons may have very similar appearances. Identifying people with very similar appearances remains a challenging task in automatic face recognition. For example, identifying twins is very difficult with respect to face recognition. The algorithms that are used for designing a face-recognition system are (Zhao, Chellappa, Rosenfeld, & Phillips, 2000): Feature-based, Template-based, Principal-Component Analysis, Auto-Associative Neural Networks, Dynamically Stable-Associative Learning, Elastic Graph Matching, Flexible Appearance Models, etc. Though many face-recognition algorithms have been proposed and have demonstrated significant promise, the task of robust face recognition is still difficult. The recent FERET test revealed that there are at least two major challenges: the Illumination variation problem and the Pose variation problem. Either of the problems may cause serious performance degradation for most existing systems. For an example, changes in illumination conditions can change the 2-D appearance of a 3-D face object dramatically, and hence can seriously affect system performance. These two problems have been documented in many evaluations of face-recognition systems and in the divided opinion of the psychology community. Unfortunately, they are unavoidable when face images are acquired in an uncontrolled environment as in surveillance video clips.

As stated earlier, there are fewer algorithms that deal with face recognition in video. This section discusses some of the well-known algorithms that have shown promise in "Face Recognition in Video."

Venkatesh, Palanivel, and Yegnanarayana (2002) describes a system for face detection and recognition in an image sequence. Motion information is used to find the moving regions, and probable eye region blobs are extracted by thresholding the image. These blobs reduce the search space for face verification, which is done by template matching. Eigen analysis of edginess representation of face is used for face recognition.

One-dimensional processing is used to extract the edginess image of the face. Experimental results for face detection show good performance, even across orientation and pose variation to a certain extent. Face recognition is carried out by cumulatively summing up the Euclidean distance between the test face images and the stored database, which shows good discrimination for true and false subjects.

Kruger, Gross and Baker (2002) presents an appearance-based, 3-D face recognition approach that is able to recognize faces in video sequences, independent from face pose. For this, the authors (Kruger, Gross, & Baker, 2002) have combined eigen light-fields with probabilistic propagation over time for evidence integration. Eigen light-fields allow the building of an appearance-based 3-D model of an object; probabilistic approaches for evidence integration are attractive in this context as they allow a systematic handling of uncertainty and an elegant way for fusing temporal information. Experiments demonstrate the effectiveness of our approach. This approach has been successfully tested on more than 20 testing sequences, with 74 different individuals.

Torres and Vila (2002) have addressed the following problem. Given a set of known images and given a video sequence to be indexed, the problem is to find where the corresponding persons appear in the sequence. Conventional face-detection schemes are not well suited for this application, and more efficient schemes have to be developed. In this paper, the authors have modified the original generic eigenface-based recognition scheme by introducing the concept of selfeigenfaces. The resulting scheme is very efficient for finding specific face images and coping with the different face conditions present in a video sequence. Their main objective was to develop a tool to be used in the MPEG-7 standardization effort to help video-indexing activities. Good results have been obtained using the video test sequences used in the MPEG-7 evaluation group.

Nakajima, Pontil, Heisele, and Poggio (2000) describes a system that learns from examples to recognize people in images taken indoors. Images of people are represented by color-based and shape-based features. Recognition is carried out through combinations of Support Vector Machine classifiers (SVMs). Different types of multi-class strategies based on SVMs are explored and compared to k-Nearest Neighbors classifiers (kNNs). The system works in real time and shows high performance rates for people recognition throughout one day.

Senior (1999) describes the application of a face-recognition system to video indexing, along with labeling faces in the video and identifying speakers. The face-recognition system can be used to supplement acoustic speaker identification, when the speaker’s face is shown, to allow indexing of the speakers, as well as the selection of the correct speaker-dependent model for speech transcription. The author has described the feature detection and recognition methods used by the system, as well as a new method of aggregating multiple Gabor jet representations for a whole sequence. Several approaches to using such aggregate representation for recognition of faces in image sequences are compared. The author has also shown that there is significant improvement in recognition rates when the whole sequence is used instead of a single image of the face.

Nastar (1998) presents an experimental setup for real-time face identification in a cluttered scene. To design this system, color images of people have been recorded with a static camera. After that, a face has been detected from these video sequences, and the resulting images are stored in a database. At a future time, a person standing in front of the camera (although against a different background) is identified if his or her image is present in the database. In this experiment, the main variation of the faces is wide-pose variation (out-of-image plane rotation of the head); some scale variation is also present. For real-time ability, a combination of simple image features through a voting procedure for performing face recognition has been carried out.

In an interesting work by Howell and Buxton (1996), the authors have presented experiments for unconstrained face recognition using Radial Basis Function (RBF) networks in low-resolution video information. RBF algorithms have shown excellent levels of performance where the view varies, and the authors also discussed how to relax constraints on data capture and improve preprocessing to obtain an effective scheme for real-time, unconstrained face recognition. Howell and Buxton have also concluded that the locally tuned, linear Radial Basis Function networks can be used for excellent performance in the simpler face-recognition task in video sequences when there is any training set of image sequences.

In Chellappa, Zhou, and Li (2002), a time series state space model has been introduced in a Bayesian approach to accommodate the video. In this model, the goal reduces to estimate the posterior distribution of the state vector given the observations up to the present, and the *Sequential Importance Sampling* (SIS) algorithm is invoked to generate a numerical solution to this model. However, the ultimate goal of this system is to estimate the posterior distribution of the identity of humans for recognition purposes. In this work, two methods have been presented to approximate the distribution under different experimental scenarios.

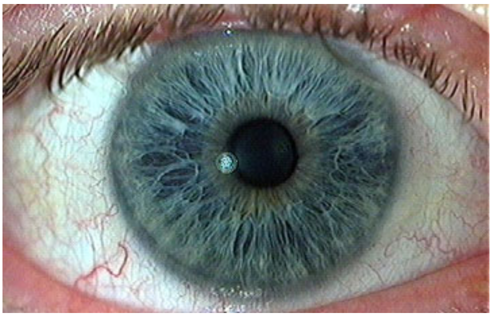
In Kruger and Zhou (2002), a new exemplar-based probabilistic algorithm for face recognition in video sequences is presented. The algorithm has two stages in which the exemplars (selective representatives from the raw video) are first automatically extracted. These exemplars are used to summarize the gallery video information. In the second part, the exemplars are used as centers for the probabilistic mixture distributions for the tracking and recognition process. Experimental results on more than 100 training and testing sequences of 25 different individuals have demonstrated the effectiveness of the algorithm.

These are some of the algorithms for face recognition in video sequences. There are possibilities for developing the algorithms for recognizing human faces in video sequences from the algorithms that have been used in still images. Research is still going on to overcome the limitations of the video-based recognitions.

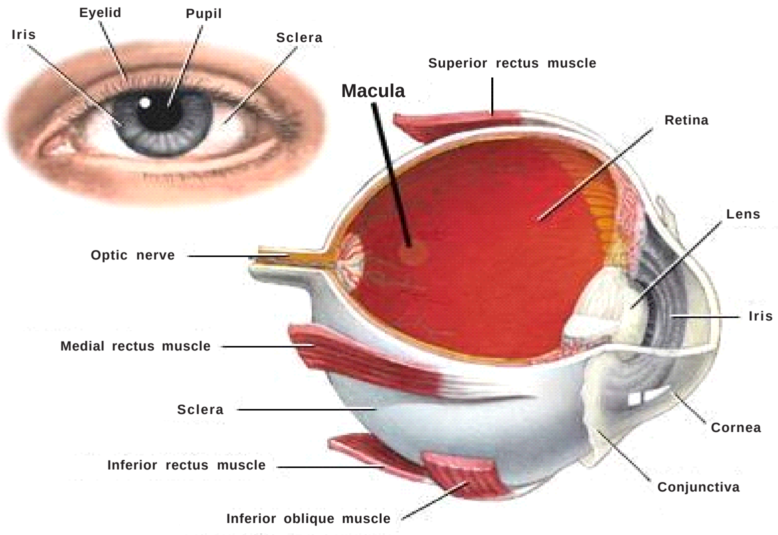
EYE

The eye is a complex organ that serves as the core of our most treasured sense — sight. When light enters the eye, it passes through the cornea where two-thirds of focus is achieved. The light then passes through the pupil where the iris adjusts the amount of light that is allowed to enter. The focused light finally reaches the retina, seven layers of alternating cells and processes that convert a light signal into a neural signal ("signal transduction"). The actual photoreceptors are the rods and cones, but the cells that transmit to the brain are the ganglion cells. The axons of these ganglion cells make up the optic nerve, the single route by which information leaves the eye. Once the image reaches the brain you have sight!

Figure 3. Eye (from www.whyfiles.org/163amd_eye)



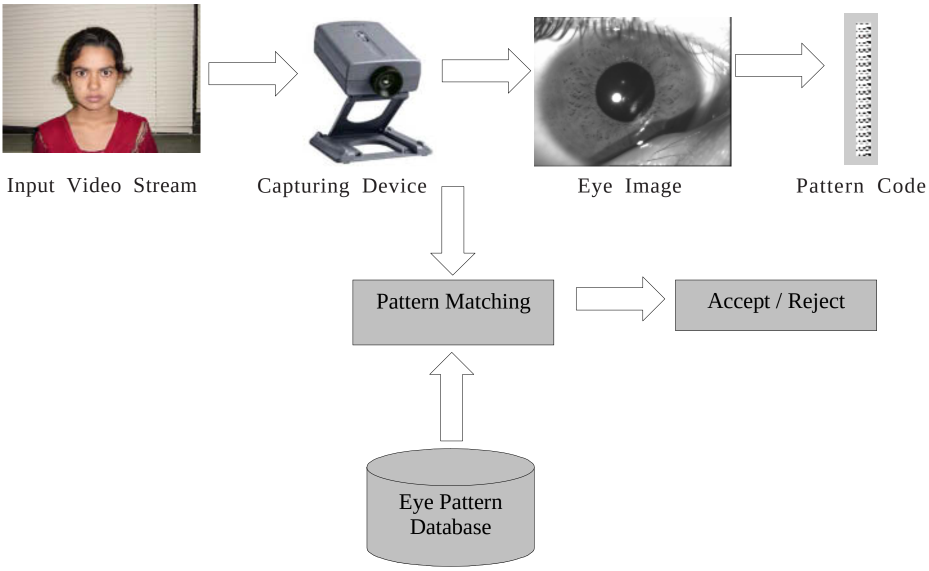
(a)



(b)

For authenticating any person using the eye as the biometric feature, the process given in the diagram above is followed. An input video stream is taken using an infrared sensitive CCD camera and the frame grabber. For taking the video stream infrared light is used for taking the video stream because the blood vessels absorb infrared light more quickly than the surroundings. From this video stream, the eye is captured using some capturing algorithm. This eye image is localized using various image-processing algorithms. The area of interest (retina/iris) is then detected from the eye and the features (300 to 700 data points) are extracted. These features are encoded into a pattern using some algorithm like wavelet theory. For enrollment, the templates/patterns are stored in the database whereas for authentication, the encoded patterns are matched with those stored in the database using some pattern-matching algorithm.

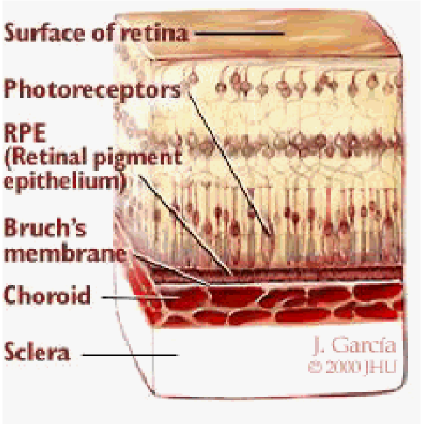
Figure 4. Person authentication using eye (retina/iris)



Retinal Scan

The retinal scan is actually one of the oldest biometrics. Two ophthalmologists discovered that each eye possess a unique pattern of blood vessels in the 1930s. Later, in the 1950s, another physician studying maternal twins found that their retinal patterns were also unique (Jain, Bolle, & Pankanti, 1999). They even discovered that barring disease and severe injury, the retina's vascular patterns are stable throughout one's lifespan. The retinal biometrics were put to use when the first retinal scan was made by

Figure 5. Anatomy of the retina and posterior eye (from www.whyfiles.org/163amd_eye)



EyeDentify. The Eyedentification 7.5 is the first retinal scan developed for commercial use.

A retinal scan is a well-established biometric trait and involves the electronic scanning of the retina — the innermost layer of the eyeball. Scanning is done by emitting a beam of incandescent light that bounces off the person's retina and returns to the scanner. This retinal scanning system analyzes the pattern of the blood vessels at the back of eye and records it into an easily retrievable digitized database, or measures the intensity of IR light reflected from an annular region of the retina. The reason for the annular scan is the reduction of light noise arising from corneal reflections. The width and diameter of a retinal scanner's annular region of measurement is chosen to allow for the collection of an adequate set of data (even if the pupil is very small). These patterns/ data are unique to every individual and cannot be replicated.

Scanning involves using a low-intensity light source and an optical coupler and can read the patterns at a great level of accuracy. It does require the user to remove glasses, place his or her eye close to the device, and focus on a certain point for several seconds, generally 15-20. The eye's natural reflective and absorption properties are used to map a specific portion of the retinal vascular structure. Three hundred to 400 points of reference are captured and stored in a 96-byte field. To enroll any person into the database, a minimum of five scans is required which takes about 45 seconds. Thus the template size for matching is very small and the operational speed is also good. It gives very high accuracy in comparison to most other biometrics.

The drawback is that the user must look directly into the retinal reader. This is inconvenient for eyeglass wearers and in public applications; there may also be concerns about the spread of germs because of the need for physical contact with the retinal scanner. Trauma to the eye and certain diseases can even change the retinal vascular structure. Orientation problems are minimized because the eye naturally aligns itself as it focuses on an illuminated target.

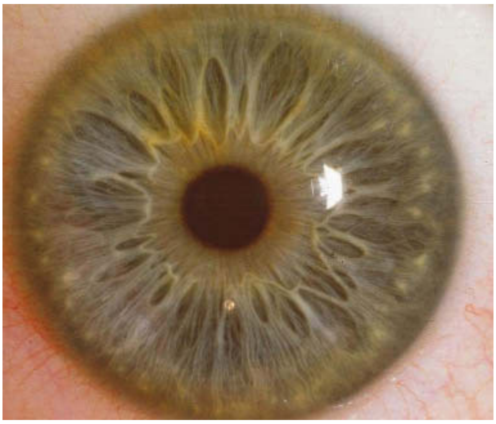
The application of retinal scanning is limited to high-end security applications like controlling physical access to sensitive areas or rooms in military installations and power plants, etc., due to the comparatively high cost of the retinal scanning systems. Because of the restricted usage of retinal scanning technology, the false acceptance and the false rejection rate are tolerated. The false acceptance rate (where an unauthorized user is accepted) is less than 0.0001%, and the false rejection rate (where an authorized user is rejected) is 10%.

Iris

Iris recognition uses the iris — the colored circle (as shown in Figure 6) that surrounds the pupil — as the physical characteristic to be measured. The iris contains many randomly distributed immutable structures, making each iris distinct from another. Like the retina, the iris does not change with time.

Since the 1990s, many researchers have worked on this problem. In this section, the various algorithms for iris recognition will be discussed. The input video is captured and the eye is extracted from the input frame. From this eye, the iris is detected and further processing is done for authentication using the iris. The human iris-identification process is basically divided into four steps:

Figure 6. Iris image (from http://www.cl.cam.ac.uk/users/jgd1000/)



- *Localization.* The inner and the outer boundaries of the iris are calculated.
- *Normalization.* The irises of different people may be captured in different sizes; size may also vary for the same person because of the variation in illumination and other factors.
- *Feature extraction.* The iris provides abundant texture information. A feature vector is formed that consists of the ordered sequence of features extracted from the various representation of the iris images.

Figure 7. Iris image in infra-red light (from http://www.cl.cam.ac.uk/users/jgd1000)

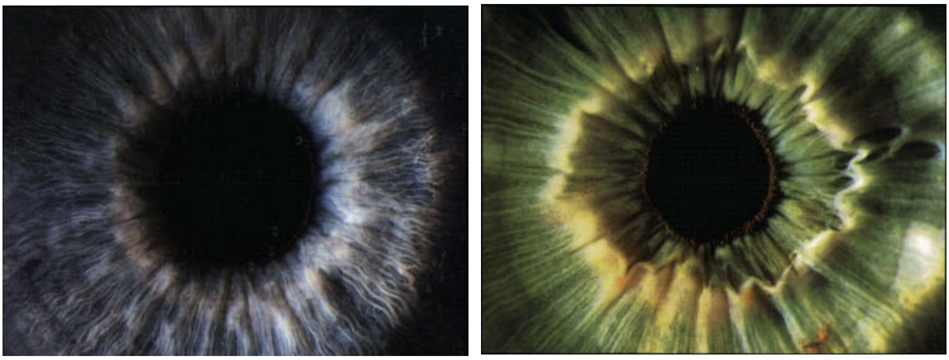
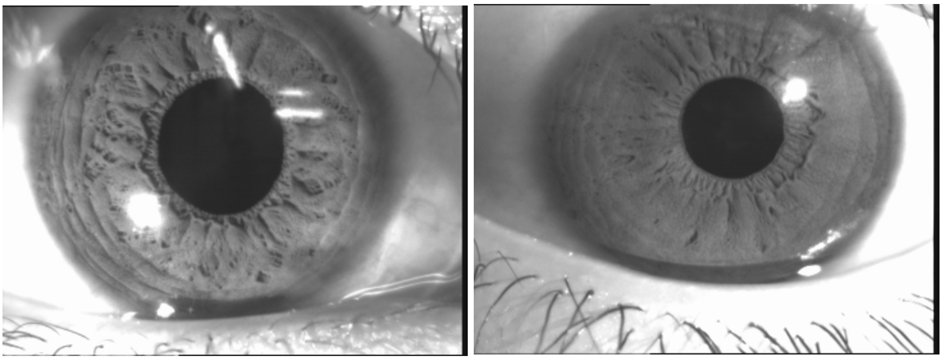


Figure 8. Iris image in visible light



- *Matching.* The feature vectors are classified through different thresholding algorithms like Hamming Distance, weight vector and winner selection, dissimilarity function, etc.

Daugman (1993, 2001) was the first to provide an algorithm for iris recognition. He has designed the algorithm based on Iris Codes. In the preprocessing step, for example, inner and outer boundaries of the iris are located. Integro-differential operators are then used to detect the centre and diameter of the iris, then the pupil is also detected using the differential operators. For conversion from Cartesian to polar transform, rectangular representation of the required area is made. The feature-extraction algorithm is designed based on modified complex-valued 2-D Gabor wavelets (Daugman, 1993, 2001). For matching, Hamming Distance has been calculated by the use of simple Boolean Exclusive – OR operator and for the perfect match, the hamming distance equal to zero is obtained. The algorithm has an accuracy of more than 99.9%. It is found that the time required for iris identification is less than one second.

Wildes (1999) has made use of an isotropic band-pass decomposition derived from the application of Laplacian of Gaussian filters to the image data. Like Daugman (1993, 2001), Wildes has also used the first derivative of image intensity to find the location of edges corresponding to the borders of the iris. The Wildes system explicitly models the upper and lower eyelids with parabolic arcs, whereas Daugman excluded the upper and the lower portions of the image. The results of this system are found good enough to recognize the individuals in minimum time period.

Boashash and Boles (1998) have presented a new algorithm based on zero-crossings (Mallat, 1991). They have first localized and normalized the iris by using edge detection and other well-known computer-vision algorithms. The zero-crossings of the wavelet transform are then calculated at various resolution levels over concentric circles on the iris. The resulting one dimensional (1-D) signals are then compared to the model features using different dissimilarity function. This system can handle noisy conditions as well as variations in illumination. This algorithm is also translation, rotation, and scale invariant. A similar type of system has been presented by Sanchez-Avila, Sanchez-Reillo, and deMartin-Roche (2001) that is based on zero-crossing discrete dyadic wavelet transform representation and has shown a high level of accuracy.

In Noh, Kwanghuk, Lee, and Kim (2002), a new algorithm has been proposed to extract the features of iris signals by Multi-resolution Independent Component Identification (M-ICA). M-ICA provides good properties to represent signals with time frequency. This extracts the iris features that are used for matching using conventional algorithms. The accuracy obtained is low because the M-ICA does not give good performance on class-separability.

There are some other researchers who have used different algorithms for feature extraction. Dargham, Chekima, Liau, and Lye (2002) used thresholding to detect the iris from the pupil and the surroundings. The detected iris is then reconstructed into a rectangular format. Self- organizing map networks are then used for recognizing the iris patterns. The accuracy obtained by the network is around 83%. In another algorithm by Li, Tan, and Wang (2002), circular symmetry filters are used to capture local texture information of the iris, which are then used to construct a fixed-length feature vector. The nearest-feature line algorithm is used for iris matching. The results obtained were 0.01% for false match and 2.17% for false non-match rate. Chen and Yuan (2003) have developed an algorithm for extracting the iris features based on fractal dimension. The iris zone is partitioned into small blocks in which the local fractal dimension features are computed as the iris code. Finally, the patterns are matched using the k-means and neural networks. The results obtained are 91.8% acceptance for authentic person and 100% rejection rate for fakers. Yong, Tieniu, and Wang (2000) used Gabor filters and 2-D wavelet transforms for feature extraction. For identification, weighted Euclidean distance classification has been used. This algorithm is invariant to translation and rotation and tolerant to illumination. The classification rate on using Gabor is 98.3%, and the accuracy with wavelets 82.51%. Tisse, Torres, and Robert (2002) have proposed an algorithm for localization and extraction of iris. For localization, a combination of the integro-differential operators with a Hough Transform is used, and for feature extraction, the concept of instantaneous phase or emergent frequency is used. Iris code is generated by thresholding both the models of emergent frequency and the real and imaginary parts of the instantaneous phase. Finally, the matching is performed using Hamming distance. In this algorithm, the false rejection rate obtained is 11%. Lim, Lee, Byeon, and Kim (2001) have used the Haar Wavelet transform to extract features from iris images. By applying the transform four times on image of size 450X60 and combining the features, 87-bit feature vector has been obtained. This feature vector is the compact representation of the iris image. Finally, for classification of feature vectors, weight-vector initialization and winner-selection strategy has been used. The recognition rate obtained is around 98.4%. In Machala (2001), two new algorithms of the statistical and computer evaluations of the iris structure of a human eye for personal identification have been proposed that are based partly on the correlation analysis and partly on the median binary code of commensurable regions of digitized iris image. Similarly, an algorithm of eye-iris structure characterization using statistical and spectral analysis of color iris images is considered in Gurianov, Zimnyakov, and Galanzha (2001). They used Wiener spectra for characterization of iris patterns. In Kois (2001) and DellaVecchia, Chmielewski, Camus, Salganioff, and Negin (1998), human iris structure is explained and classified using coherent Fourier spectra of the optical transmission.

In Ales (2001), an efficient biometric security algorithm for iris recognition system with high performance and high confidence has been described. The system is based on an empirical analysis of the iris image, and it is split into several steps using local image properties. The various steps are: capturing iris patterns; determining the location of iris boundaries; converting the iris boundary to the stretched polar coordinate system; extracting the iris code based on texture analysis using wavelet transforms; and classification of the iris code. The proposed system use the wavelet transforms for texture analysis, and it depends heavily on the knowledge of general structure of human iris. The system has been implemented and tested using a dataset of 240 samples of iris data with different contrast quality. Because the iris (the colored part of the eye) is visible from a distance of about one foot, direct contact with the scanner is not required nor is it necessary to remove eyeglasses. The algorithm does not rely on the iris color, and this is important because of the popularity of colored contact lenses. Scanning overcomes most of the problems of retinal scanners.

Disadvantages of iris recognition include problems of user acceptance and the relative expense of the system as compared to other biometric technologies.

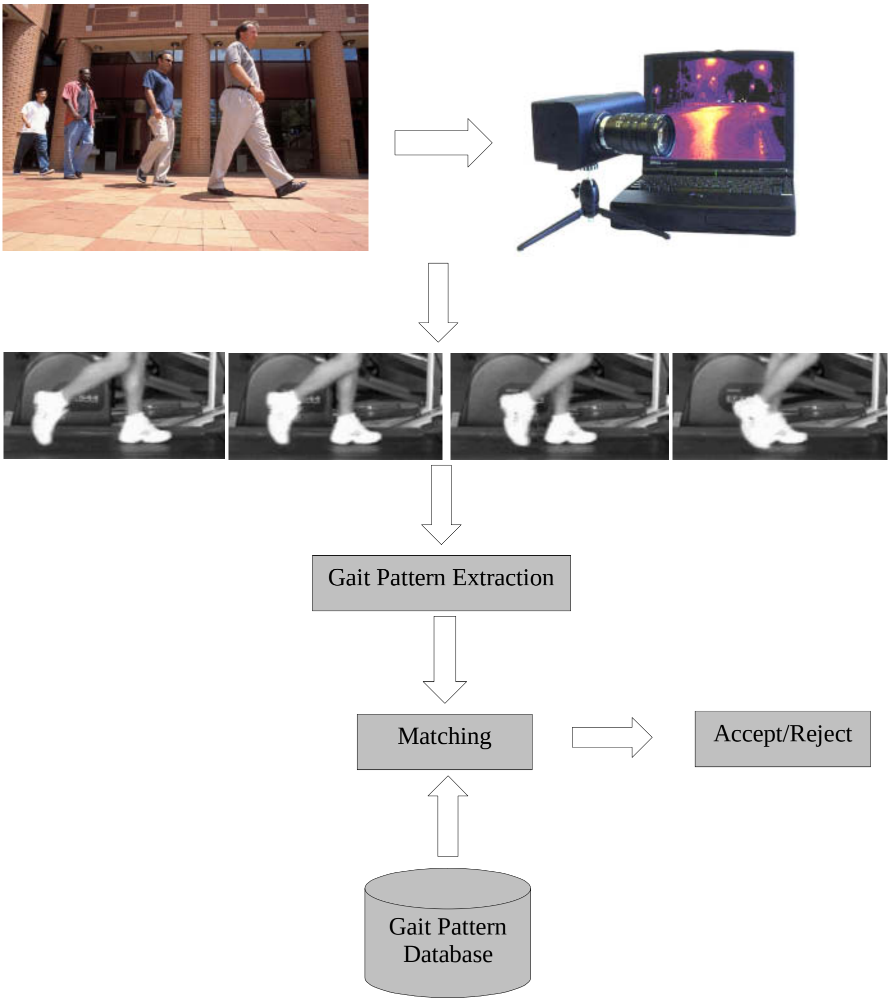
GAIT

Recognizing people by their gait is an emergent biometric. The potential of gait as a biometric has also been encouraged by the considerable amount of evidence available in medicine and literature. The main advantages of using this biometric are that it can be applied from a distance and that too the individual under observation need not to be aware of it. Every individual's gait is unique (Nixon, 2000), dependent on the musculoskeletal structure. The dynamic nature of gait implies that it is more variable than other types of biometrics and may be influenced by psychological factors (such as mood and stress) and physical factors as well (e.g., footwear). This is very useful for surveillance applications.

Figure 9 explains the procedure for gait recognition. A video sequence is captured (without concerning the user that his/her photograph is being captured). From this sequence, the structure of the person is obtained by tracking some specific points or by using motion information from a series of poses. Once the structure of the person is obtained, the information about gait is extracted and the matching is done for comparison with the database.

We can divide the work in automatic gait recognition from visual measurements into two types of algorithms — model-free and model-based algorithms. In model-free algorithms, there is no underlying representation of the three-dimensional structure of walking, but they have an implicit model of walking built into their algorithms of extracting features. These algorithms analyze the motion or shape of the subjects as they walk, and the features extracted from the motion or shape are used for identification. Model-based algorithms either model the person or explicitly model the style of the walking of the person. In person models, a body model is fit to the person in every frame of the walking sequence, and parameters are measured on the body model as the model deforms over the walking sequence. In walking models, a model of how the person moves is created, and the parameters of the model are learned for every person. Most of the algorithms have two major deficiencies: the lack of generality of viewing condition, and that most researchers do not consider whether the confusion that arises is caused by vision yielding noisy measurements or by having chosen features with low discrimination power.

Figure 9. Steps for gait recognition



Perhaps the first paper in the area of gait recognition comes from the Psychology field. Cutting and Kozlowski (1977) and Cutting, Proffitt, and Kozlowski (1978) have determined that people could identify other people base solely on gait information. Stevenage, Nixon, and Vince (1999) have extended that work by exploring the limits of human ability to identify other humans by gait under various viewing conditions.

The algorithms for gait recognition include modeling gait as a spatio-temporal sequence and as an articulated model. In Cunado, Nixon, and Carter (1997), Hough transform has been used to extract the lines that represent legs in a sequence of video images. Change in inclination of these lines follow a simple harmonic that is used as the gait biometric. For smoothing the data and infilling the missing points, the method of least squares is used. To get the frequency components of the change in inclination of the legs, Fourier transform is applied and the transformed data is applied using k-nearest neighbor rule. This shows that the gait classification is dependent on the frequency content and the phase as well.

Another algorithm by Nixon et al. (Nixon, 1998) has combined canonical space transformation (CST) with the eigenspace transformation (EST) for feature extraction of temporal templates in a gait sequence. EST has been a potent metric in face and gait analysis but without using data analysis to increase classification capability. The combination of EST and CST reduces data dimensionality and optimizes the class separability of different gait sequences simultaneously, thus making the recognition of human gait by template matching faster and easier.

In Cunado (1999), the two algorithms just discussed have been merged and gait signature has been extracted directly from the evidence-gathering process by using a Fourier series to describe motion of the upper leg. Temporal evidence-gathering algorithms have been applied to extract the moving model from a sequence of images. It has been stated that this algorithm can also handle occlusion.

The new velocity Hough transformation (to extract moving conic sections) combined with a continuous formulation for arbitrary shape extraction has been presented in Grant, Nixon, and Lewis (1999). This algorithm shows better performance over contemporaneous single-image extraction algorithms. The performance is also found to be better when the sequence is having noise and occlusion.

An algorithm using the Generalized Symmetry Operators has been implemented in Hayfron-Acquah, Nixon, and Carter (2001a, 2001b). It relies on locating features by their symmetrical properties (which are unique for every individual) instead of locating the borders of a shape. The gait signatures have been derived for silhouettes and optical flow using Fourier transform. The algorithm has been applied on two different databases, and an accuracy of 95% has been achieved. This symmetry measure can handle problems like noise, missing frames, and occlusion also.

In Foster (2001) a new measure for biometrics called gait masks has been presented. These gait masks derive information from a sequence of silhouettes, such as how the silhouette changes over time in a chosen region of the body. The area changes are related to the nature of gait. Canonical analysis of the output has been done for recognition, and the accuracy is found to be 80%.

Analytical pose compensation algorithms can be applied on the algorithms based on the modeling of human walking. In Carter and Nixon (1999), a geometric correction has been done to the measurement of the hip rotation angle based on the known orientation to the camera. It has used the invariance properties of angles under geometric projections. Thus, using geometric analysis, invariant signatures have been generated for the automatic gait analysis.

Lee and Grimson (2002) describe a set of representations of features for gait appearance. Firstly, a set of image features is computed that Lee and Grimson (2002) are based on the moments extracted from the orthogonal-view video silhouettes of human walking motion. And then the image features are aggregated over time to create the gait sequence features.

Huang (2001) proposes a statistical algorithm for feature extraction from spatial and temporal templates to optimize the class separability and reduce the data dimensionality of different gait sequences simultaneously. Principal Component Analysis (PCA) is applied on the extracted templates for dimensionality reduction. In the canonical space, accumulated distance is used as the metric for gait recognition.

BenAbdelkader, Cutler, and Davis (2002) has encoded the planar dynamics of a walking person in a 2-D plot consisting of the pair-wise image similarities of the sequence of images of the person, and the standard classification of these plots is done for gait recognition. Background modeling is done for a number of frames to track the person, and a sequence of segmented images of the person is obtained. The recognition procedure uses PCA and k-nearest neighbor rule. Accuracy obtained is around 77-78%.

Tekalp, as discussed in Dockstader, Berg, and Tekalp (2002), has introduced a new model-based algorithm towards the 3-D tracking of human motion. A parametric body model has been characterized by hard- and soft-kinematics constraints. Hard constraints are derived from the actual body measurements, while soft constraints are taken from *a priori*, probabilistic distributions for each model parameter, based on the previous examples of human body configurations. This knowledge from soft kinematics is used to define a natural acceleration of body parameters towards their expected values, which is used to augment the potentially time-varying velocity of a classical dynamic-motion model. As a result, there is an increase in the tracking accuracy in the presence of occlusion and articulated movement.

Using gait as a biometric is a relatively new area of study within the realms of computer vision. It has been receiving growing interest within the computer-vision community, and a number of gait metrics has been developed. We use the term *gait recognition* to signify the identification of an individual from a video sequence of the subject walking. This does not mean that gait is limited to walking — the term can also be applied to running or any means of movement on foot. Gait as a biometric can be seen as advantageous over other forms of biometric identification

CONCLUSIONS

Demands for biometric-based personal authentication technologies are progressively increasing. There are high expectations of the applications to the network security field. To meet these demands, continuous research and development for video biometrics is going on, aiming at faster speeds, smaller size, and lower cost, and is expanding its application fields utilizing non-contact authentication and high-authentication accuracy.

Thus, in this chapter, we have presented a tutorial about the three video biometric traits, namely, face, eye, and gait. An extensive survey of algorithms used for the three has also been provided as a reference for the readers. Using these biometrics traits for video sequences, anyone can secure or authenticate user access.

ACKNOWLEDGMENT

The authors wish to acknowledge Professor J. Daugman of Cambridge University, UK, and Mr. S. Sukumar, India, for providing the iris images. This work has been done as a part of the project sponsored by the Department of Communication and Information Technology, India.

REFERENCES

Aggarwal, J.K., & Cai, Q. (1999). Human motion analysis: A review. *Computer Vision and Image Understanding, 73*(3), 428-440.

Ales, M., Kois, P., & Pospísil, J. (2001). Identification of persons by means of the Fourier spectra of the optical transmission binary models of the human irises. *Optics Communications, 192*, 161-167.

BenAbdelkader, C., Cutler, R., & Davis, L. (2002). Motion-based recognition of people in eigengait space. *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition.*

Bishop C. M. (1996). *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press.

Boashash, B., & Boles, W.W. (1998). A human identification technique using images of the iris and wavelet transform. *IEEE Transactions on Signal Processing, 46*(4), 1185-1188.

Bobick & Davis, J. (2001). The recognition of human movement using temporal templates. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 23*(3), 257-267.

Bulthoff, H., Little, J., & Poggio, T. (1989). A parallel algorithm for real-time computation of optical flow. *Nature, 337*, 549-553.

Carter, J.N., & Nixon, M.S. (1999). On measuring gait signatures which are invariant to their trajectory. *Measurement and Control, 32*(9), 265-269.

Cedras, C., & Shah, M. (1994). A survey of motion analysis from moving light displays. *Proceedings of IEEE Conference on Computer Vision & Pattern Recognition*. Seattle, WA, 214-221.

Chellappa R., Wilson, C. L., & Sirohey, S., (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE, 83*(5), 705-741.

Chellappa, R., Zhou, S., & Li, B. (2002). Bayesian methods for face recognition from video, *Proceedings of International Conference on Acoustics Speech and Signal Processing*.

Chen, W.S., & Yuan, S. (2003). A novel personal biometric authentication technique using human iris based on fractal dimension features. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*.

Chen, Z., & Lee H. I. (1992). Knowledge-guided visual perception of 3D human gait from a single image sequence. *IEEE Transaction on Systems, Man and Cybernetics*, *22*(2), 336-342.

Cootes, T. F., & Taylor C. J. (1992). Active shape models—Smart snakes. *Proceedings of the British Machine Vision Conference,* 266-275.

Craw, I., Tock, D., & Bennett, A. (1992). Finding face features. *Proceedings of the Second European Conference on Computer Vision*, 92-96.

Cunado, D., Nash, J.M., Nixon, M.S., & Carter, J.N. (1999) Gait extraction and description by evidence-gathering. *Proceedings of the Second International Conference on Audio and Video Based Biometric Person Authentication*, 43-48.

Cunado, D., Nixon, M.S., & Carter, J. N. (1997). Using gait as a biometric, via phase-weighted magnitude spectra. In J. Bigun, G. Chollet, & G. Borgefors (Eds.), *Proceedings of 1st International Conference on Audio and Video Based Biometric Person Authentication*, 95-102.

Cunado, D., Nixon, M.S., & Carter, J. N. (1999). Automatic gait recognition via model-based evidence gathering. In L. O' Gorman & S. Shellhammer (Eds.), *Proceedings AutoID '99: IEEE Workshop on Identification Advanced Technologies*, 27-30.

Cunado, D., Nixon, M.S., & Carter, J.N. (2003). Automatic extraction and description of human gait models for recognition purposes. *Computer Vision and Image Understanding*, *90*(1), 1-41.

Cutler, R., & Davis, L. (2000). Robust real-time periodic motion detection, analysis and applications. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *22*(8), 781-796.

Cutting, J., & Kozlowski, L. (1977). Recognizing friends by their walk: Gait perception without familiarity cues. *Bull. Psychonometric Society*, *9*, 353-356.

Cutting, J., Proffitt, D. & Kozlowski, L. (1978). A biomechanical invariant for gait perception. *Journal of Experimental Psychology: Human Perception and Performance*, *4*(3), 357-372.

Dargham, J.A., Chekima, A., Liau, C.F., & Lye, W.L. (2002). Iris recognition using self-organizing neural network. *Student Conference on Research and Development*, 169 -172.

Daugman J. G. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(11), 1148-1161.

Daugman, J. G. (2001). Statistical richness of visual phase information: Update on recognizing persons by iris patterns. *International Journal of Computer Vision*, *45*(1), 25-38.

Davies, C.J., & Nixon, M.S (1998). A Hough transform for detecting the location and orientation of three-dimensional surfaces via color encoded spots. *IEEE Transactions on Systems, Man, and Cybernetics*, *28*(1), 90-95.

Davis, R.B., & DeLuca, P.A. (1997). Clinical gait analysis—Current methods and future directions, In G. F. Harris & P. A. Smith (Eds.), *Human motion analysis*. Chapter 2, pp. 17-42. Piscataway, NJ: IEEE Press.

Della Vecchia, M.A., Chmielewski, T., Camus, T.A., Salganicoff, M., & Negin, M. (1998). Methodology and apparatus for using the human iris as a robust biometric. *Proceedings of the SPIE*, 3246, 65-74.

Dockstader, S.L., Berg, M.J., & Tekalp, A.M. (2002). Performance analysis of a kinematic human motion model. *Proceedings of the International Conference on Multimedia and Expo*, 885-888.

Dockstader, S.L., & Tekalp, A.M. (2001). On the tracking of articulated and occluded video object motion, *Real-Time Imaging, 7*(5), 415-432.

Dockstader, S.L., & Tekalp, A.M. (2002). A kinematic model for human motion and gait analysis. *Proceedings of International Conference on Multimedia*, Switzerland.

Dubuisson, M.-P,. & Jain, A.K. (1995). Contour extraction of moving objects in complex outdoor scenes. *International Journal of Computer Vision*, *14*(6), 83-105.

Foster, J., Nixon, M., & Prugel-Bennett, A. (2001). New area based metrics for automatic gait recognition. *Proceedings of the British Machine Vision Conference*, 233-242.

Foster, J., Nixon, M., & Prugel-Bennett, A. (2002). Gait recognition by moment based descriptors. *Proceedings 4th International Conference on Recent Advances in Soft Computing*, 78-84, Nottingham, UK.

Foster, J. P., Prugel-Bennett, A., & Nixon, M.S. (2001). New area measures for automatic gait recognition. *Proceedings of the BMVA Workshop Understanding Visual Behavior*.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition (2nd ed.)*. New York: Academic Press.

Gavrila, D. M. (1999). The visual analysis of human movement: A Survey. *Computer Vision and Image Understanding, 73*(1), 82-98.

Graf, H.P., Chen, T., Petajan, E., & Cosatto, E. (1995). Locating faces and facial parts. *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition,* 41-46.

Grant, M.G., Nixon, M.S., & Lewis, P.H. (1999). Finding continuous-model moving shapes by evidence gathering. *Proceedings of the IEE Colloquium Motion Analysis and Tracking*, 14, 1-4.

Grant, M.G., Nixon, M.S., & Lewis, P.H. (2002). Extracting moving shapes by evidence gathering. *Pattern Recognition, 35*(5), 1099-1114.

Gurianov, E.V., Zimnyakov, D.A., Galanzha, V.A. (2001). Iris patterns characterization by use of Wiener spectra analysis: Potentialities and restrictions. *Proceedings of the SPIE,* 4242, 286-290.

Hayfron-Acquah, J.B., Nixon, M.S., & Carter, J. N. (2001a). Automatic gait recognition by symmetry analysis. In J. Bigun & F. Smeraldi (Eds.), *Proceedings of the Audio and Video Based Biometric Person Authentication,* 272-277.

Hayfron-Acquah, J.B., Nixon, M.S., & Carter, J.N. (2001b). Automatic gait recognition via the generalized symmetry operator. *Proceedings of the BMVA Workshop Understanding Visual Behavior*.

Howell, J., & Buxton, H. (1996). Towards unconstrained face recognition from image sequences. *Proceedings of the Second International Conference on Automatic Face and Ge*sture *Recognition*, 224-229.

Huang, P.S. (2001). Automatic gait recognition via statistical approaches for extended template features. *IEEE Transactions on Systems, Man and Cybernetics,* Part B. *Cybernetics*, *31*(5).

Huang, P.S., Harris, C.J., & Nixon, M. S. (1998a). Visual surveillance and tracking of humans by face and gait recognition. *Proceedings of 7th IFAC Symposium on Artificial Intelligence in Real-Time Control*, 43-44.

Huang, P.S., Harris, C.J., & Nixon, M.S. (1998b). Recognizing humans by gait using a statistical approach for temporal templates. *Proceedings of International Conference on Systems, Man and Cybernetics*, 4556-4561.

Huang, P.S., Harris, C.J., & Nixon, M.S. (1998c). Canonical space representation for recognizing humans by gait and face. *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation,* 180-185.

Huang, P.S., Harris, C.J., & Nixon, M.S. (1998d). A statistical approach for recognizing humans by gait using spatial-temporal templates. *Proceedings of International Conference on Image Processing*, 3, 178-182.

Huang, P.S., Harris, C.J., & Nixon, M.S. (1998e). Comparing different template features for recognizing people by their gait. *Proceedings of Ninth British Machine Vision Conference,* 639-648.

Huang, P.S., Harris, C.J., & Nixon, M.S. (1999). Recognizing humans by gait via parametric canonical space. *Journal of Artificial Intelligence in Engineering*, *13*(4), 359-366.

Jafar, M.H., Aboul, A., & Hassanien, E. (2003). An iris recognition system to enhance e-security environment based on wavelet theory. *AMO - Advanced Modeling and Optimization, 5*(2), 93-104.

Jain, A.K., Bolle, R., & Pankanti, S. (1999). *BIOMETRICS personal identification in networked society*. Boston, MA: Kluwer Academic Press.

Johansson, G. (1975). Visual motion perception. *Scientific American, 232*, 76-88.

Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, *1*(4), 321-331.

Kois, P., Ales, M., & Pospisil, J. (2001). Human iris structure by the method of coherent optical Fourier transform. *Proceedings of the SPIE* , 4356, 394-400.

Kruger, V., Gross, R., & Baker, S. (2002). Appearance-based 3-D face recognition from video. *Symposium of the German Association for Pattern Recognition*.

Kruger, V., & Zhou, S. (2002). Exemplar-based face recognition from video. *Proceedings of the European Conference on Computer Vision*, Copenhagen, Denmark.

Lam, K., & Yan, H. (1994). Fast algorithm for locating head boundaries. *Journal of Electronic Imaging, 3*(4), 351-359.

Lee, L., & Grimson, W. (2002a). Gait analysis for recognition and classification. *Proceedings of 5th International Conference on Automatic Face and Gesture Recognition*.

Lee. L. & Grimson, W.E.L. (2002b). *Gait appearance for recognition.* MIT AI Laboratory.

Li, M., Tan, T., & Wang, Y. (2002a). Iris recognition based on multichannel Gabor filtering. *Proceedings of the International Conference on Asian Conference on Computer Vi*sion, 1-5.

Li, M., Tan, T., & Wang, Y. (2002b). Iris recognition using circular symmetric filters, *Proceedings of the 16th International Conference on Pattern Recognition*, 2, 414-417.

Lim, S., Lee, K., Byeon, O., & Kim, T. (2001). Efficient iris recognition through improvement of feature vector and classifier. *Journal of Electronics and Telecommunication Research Institute, 23*(2), 61-70.

Little, J., & Boyd, J. (1998). Recognizing people by their gait: The shape of motion. *Videre, 1*(2), 1-32.

Liu, F., & Picard, R. (1998). Finding periodicity in space and time. *Proceedings of International Conference on Computer Vision*, 376-383.

Machala, L., & Pospisil, J. (2001). Alternatives of the statistical evaluation of the human iris structure. *Proceedings of the SPIE*, 4356, 385-393.

Mallat S. G. (1991). Zero-crossing of a wavelet transform. *IEEE Transactions on Information Theory*, *37*(14), 1019-1033.

Meyer, D., & Neimann, H. (1998). Features for Optical Flow Based Gait Classification using HMM. University of Erlangen-Nuremberg.

Moghaddam, B., & Pentland, A. (1997). Probabilistic visual learning for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 696-710.

Murase, H., & Sakai, R. (1996). Moving object recognition in eigenspace representation: Gait analysis and lip reading. *Pattern Recognition Letters*, *17*, 155-162.

Nastar, C., & Mitschke, M. (1998). Real-time face recognition using feature combination. *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition.*

Nakajima, C., Pontil, M., Heisele, B., & Poggio, T. (2000). *People Recognition in Image Sequences by Supervised Learning.* MIT Report.

Nixon, M.S. (2000). Approaches to automatic gait recognition: A new biometric. In *Proceedings Natural Computing Applications Forum Spring Meeting*.

Nixon, M.S., Carter, J.N., Cunado, D., Huang, P.S., & Stevenage, S.V. (1999). Automatic gait recognition. In A.K. Jain, R. Bolle, & S. Pankanti (Eds.), Chapter 11, 231-250. Boston, MA: Kluwer Academic Publishers.

Niyogi, S.A., & Adelson, E.H. (1994). Analysis and recognizing walking figures in xyt. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. Seattle, WA, 469-474.

Noh, S.-I., Kwanghuk, P., Lee, C., & Kim, J. (2002). Multi-resolution independent component identification. *Proceedings of International Technical Conference on Circuits/Systems, Computers and Communications*.

Polana, R., & Nelson, R. (1993). Detecting activities. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2-7.

Samal, A., & Iyengar, P.A. (1992). Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition, 25*(1), 65-77.

Sanchez-Avila, C., Sanchez-Reillo, R., & de Martin-Roche, D. (2001). Iris recognition for biometric identification using dyadic wavelet transform zero-crossing. *Proceedings of the 35th IEEE International Carnahan Conference on Security Technology*, 272-277.

Senior, A.W. (1999). Recognizing faces in broadcast video. *Proceedings of IEEE Workshop on Real-Time Analysis and Tracking of Face and Gesture in Real-Time Systems*, 105-110.

Shakhnarovich, G., Lee, L., & Darrell, T. (2001). Integrated face and gait recognition from multiple views. *Proceedings of Computer Vision and Pattern Recognition*.

Sharman, K.J., Nixon, M.S., & Carter, J.N. (2000). Towards a markerless human gait analysis system. *Proceedings of the XIX Congress of the International Society for Photogrammetry and Remote Sensing*, 713-719.

Shutler, J., Grant, M., Nixon, M.S., & Carter, J.N. (2002). On a large sequence-based human gait database. *Proceedings of the 4th International Conference on Recent Advances in Soft Computing*, 66-72, Nottingham, UK.

Shutler, J.D., Nixon, M.S., & Harris, C.J. (2000). Statistical gait recognition via velocity moments. *Proc. of the IEE Colloquium*: *Visual Biometrics* (00/018), 11, 1-5.

Sidenbladh, H., Black, M.J., & Fleet, D.J. (2000a). Stochastic tracking of 3D human figures using 2D image motion. *Proceedings of the European Conference on Computer Vision*, Dublin, Ireland, 702-718.

Sidenbladh, H., Black, M.J., & Fleet, D.J. (2000b). Learning image statistics for Bayesian tracking, *Proceedings of the International Conference on Computer Vision*, Canada, 2, 709-716.

Stauffer, C., & Grimson, W. (1999). Adaptive background mixture models for real-time tracking. *Proceedings of Computer Vision and Pattern Recognition*, 246-252.

Stevenage, S., Nixon, M.S., & Vince, K. (1999). Visual analysis of gait as a cue to identity. *Applied Cognitive Psychology*, *13*(6), 513-526.

Tisse, C.-L., Torres, L., & Robert, M. (2002). Person identification technique using human iris recognition. *Proceedings of the 15th International Conference on Vision Interface*.

Torres, L., & Vila, J. (2002). Automatic face recognition for video indexing applications. *Pattern Recognition*, *35*, 615-625.

Venkatesh, B.S., Palanivel, S., & Yegnanarayana, B. (2002). Face detection and recognition in an image sequence using eigenedginess. *Proceedings of Indian Conference on Computer Vision, Graphics and Image Processing*, 97-101.

Wildes, R.P. (1999). Iris recognition: An emerging biometric technology. *Proceedings of the IEEE*, *85*(9), 1348-1363.

Wren, C. R., Clarkson, B.P., & Pentland A. (2000). Understanding purposeful human motion, *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 378-383.

Yacoob, Y. & Davis, L.S. (2000). Learned models for estimation of rigid and articulated human motion from stationary or moving camera. *International Journal of Computer Vision*, *12*(1), 5-30.

Yam, C.Y., Nixon, M.S., & Carter, J.N. (2002). Gait recognition by walking and running: A model-based approach. *Proceedings of the Asian Conference on Computer Vision*, 1-6.

Yam, H., & Lam, K.M. (1998). An analytic-to-holistic approach for face recognition based on a single frontal view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 673-686.

Yang, M.-H., Kriegman, J., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *24*(1), 34-58.

Yong, Z., Tieniu, T., & Wang, Y. (2000). Biometric personal identification based on iris patterns. *Proceedings of the IEEE International Conference on Pattern Recognition*, 2801-2804.

Zhao, W.Y., Chellappa, R., Rosenfeld, A., & Phillips, P.J. (2000). *Face recognition: A literature survey*. UMD CfAR Technical Report CAR-TR-948.

ELECTRONIC REFERENCES

http://cms.hhs.gov/coverage/ 8b3-ee8.asp

http://nlpr-web.ia.ac.cn/english/irds/irisdatabase.htm

http://www.biometricgroup.com.

http://www.cl.cam.ac.uk/users/jgd1000/

http://www.whyfiles.org/163amd_eye/

Gait Pattern Database

Chapter VII

Video Presentation Model

Hun-Hui Hsu, Tamkang University, Taiwan, ROC

Yi-Chun Liao, Tamkang University, Taiwan, ROC

Yi-Jen Liu, Tamkang University, Taiwan, ROC

Timothy K. Shih, Tamkang University, Taiwan, ROC

ABSTRACT

Lecture-on-Demand (LOD) multimedia presentation technologies among the network are most often used in many communication services. Examples of those applications include video-on-demand, interactive TV and the communication tools on a distance-learning system, and so on. We describe how to present different multimedia objects on a Web-based presentation system. Using characterization of extended media-streaming technologies, we developed a comprehensive system for advanced multimedia content production: support for recording the presentation, retrieving the content, summarizing the presentation, and customizing the presentation. This approach significantly impacts and supports the multimedia presentation authoring processes in terms of methodology and commercial aspects. Using the browser with the Windows Media Services allows those students to view live video of the teacher giving his or her speech, along with synchronized images of presentation slides, and all the annotations/ comments. In our experience, this very approach is sufficient to the use of distance learning environment.

INTRODUCTION

Multimedia presentation technologies among the network are most often used in many communication services. Examples of those applications include video-on-demand, interactive TV and the communication tools on a distance learning system, and so on. To control and demonstrate different types of multimedia objects is one of the important functions in a distributed multimedia presentation system.

Sugata Mukhopadhyay and Brian Smith (1999) developed an authoring system in which the author classifies the synchronization of timed-timed, timed-untimed, and timed-untimed. For synchronization problems of the three types, the synchronization algorithm is also provided.

In Lui, Huang, Wu, Chu, and Chen (2002), the authors propose a Web-based Synchronized Multimedia Lecture system (WSML). It focuses on the synchronization of the navigation events of mouse track, pen stroke, dynamic annotation, Scrolling, and Highlight. These events are recorded automatically based on Synchronized Multimedia Integration Language (SMIL) (World Wide Web Consortium, 2004).

We also looked at the following commercial products related to multimedia authoring or presentation designs: (1) Authorware by Macromedia, Inc.; (2) Multimedia Viewer by Microsoft; (3) Multimedia Toolbook by Asymetrix Corporation; (4) Hypermedia Authoring and Playback System by ITRI; (5) Action! By Macromedia, Inc.; (6) Audio visual connection by IBM; and (7) Astound by Gold Disk Inc.

Most systems allow users to cut and paste presentation objects or actions via button click and drawing. Multimedia Viewer also provides a set of medium editing tools. Presentation objects produced by these tools can be linked together by a script language supporting functions, data, structure, and commands. A summary of these systems follows.

- *Features:* the key research goals are as follows (as shown in Figure 1)
  - *Based on Web environment:* For teaching and training, these systems are designed to combine HTML lecture notes and video notes.
  - *Real-time editing by authors:* “Editing” means that the layout of the presentation is left to the authors. We found that the complicated operations are not used in the systems.
  - *Synchronization problem:* It is a big challenge to synchronize the multimedia objects on the Web. All of the proposed systems have a solution to this problem respectively.
  - *Create multimedia documents:* Multimedia documents are the output results of the systems and usually combine different media streams.
- *User Interface:*
  - The normal browser-user interface consists of video, slides, and slide index (as shown in Figure 2).

None of the above system, however, allows dynamic presentations. That is, a presentation generated by the above systems will stay as the form in which it was created. Different audiences watch the same presentation over and over again. Not many systems

Figure 1. Process flow of the system

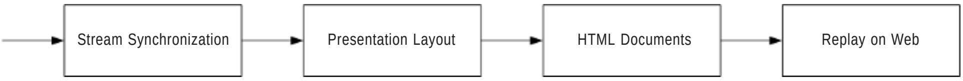
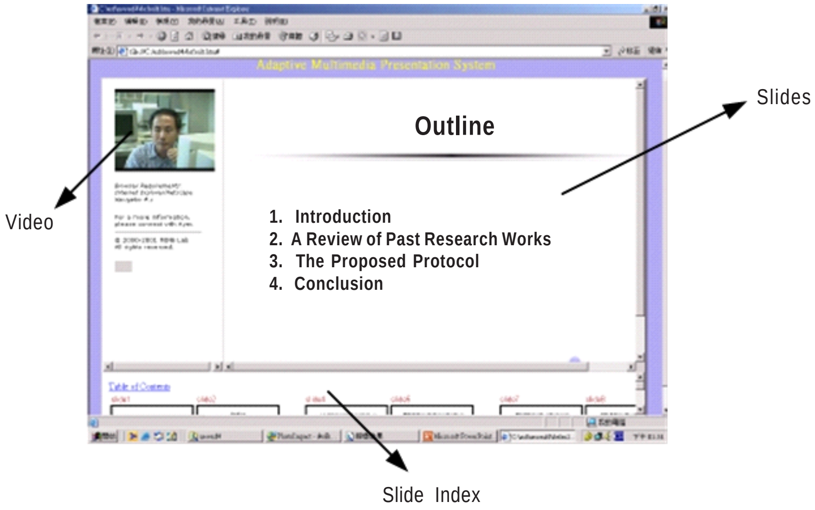


Figure 2. General browsing interface



Outline

1. Introduction
2. A Review of Past Research Works
3. The Proposed Protocol
4. Conclusion

focus on the Web-based presentation designs. Our system helps the presentation designers to design his/her presentation step-by-step until he/she published the presentation in the air.

There are some important design considerations during construction of a multimedia presentation. The first is how to model and describe the media properties during the presentation period. A multimedia presentation basic requirements include demonstrating the media's spatial, temporal, and user-interaction properties. Second is the presentation advance requirement. The user always wants alternative adaptation operation facilities, such as authoring, retrieving, abstracting/summarizing, or even rearranging the presentation performances. Unfortunately, we were rarely able to find existing presentation software that could satisfy these advanced requirements.

We have been working on multimedia presentation for several years in related popular commercial or nonprofit projects (Chang, Hassanein, & Shieh, 1998; Chang, Wang, Chan, Deng, Lee, & Wang, 2001; Multimedia Micro-University Virtual Society, http://www.mine.tku.edu.tw/mmvs; Shih & Deng, 2000; Shih, Deng, Wang, & Liou, 2001, 2001). Over the past year, we have identified and compared methods to work more closely together and to provide adaptability and reusability. An effective presentation design procedure should not only involve sequential flow of actions, but also parallel/concurrent and user-interactive actions. Additionally, the design should include a number of high-level concerns, such as goals and focus of the presentation, the user's context and current task, and the media selection to represent the information in a way that corresponds to these concerns. The first step of the research is to design a multimedia content model that is built upon the existing models. A multimedia content model is a model comprised of information coded in at least one time-dependent medium (e.g., video, audio…) and in one time-independent medium (e.g., text, image…). Multimedia document architecture demonstrates the relationships among the individual components represented as models (Steinmetz & Nahrstedt, 1995). It includes the presentation model, manipulation model, and representation model. The presentation model illustrates the media elements and how they are to be processed during running time. The manipulation model describes all the possible operations allowed for creation, change, and deletion of multimedia information. The representation model not only defines the protocols for exchanging this information among different computers, but also the formats for storing the data. It contains the relationships between the individual media elements that need to be considered during presentation. Structure implicates the basic requirements and advanced requirements while these models operate their functions.

SYSTEM DEVELOPMENT

The adaptive video presentation management system includes four major modules (Figure 3): (1) Video and Audio Encoder, (2) Synchronization Controller, (3) Post-Processing Manager, and (4) Multimedia Presentation Browser. Most of the developed systems provide real-time processing functions (Encoding and Synchronization). Our system extends the general authoring tools to manage the post-processing functions such as material reusing, combination, and auxiliary resources insertion.

- *Video/Audio Encoding:* Mandatory in authoring and rendering, using uncompressed files or files compressed with the Windows Media Audio, Sipro Labs ACELP, MPEG-3, or another codec. For easily publishing on the Web, the

Figure 3. Framework of the adaptive video presentation system

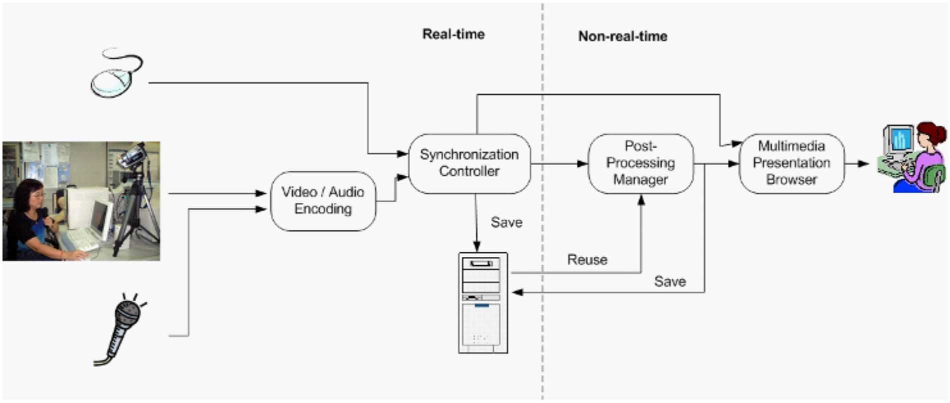
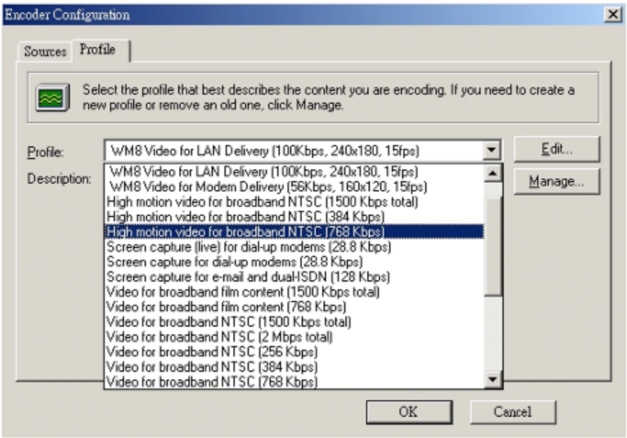


Figure 4. Different encoding modes.



Select the profile that best describes the content you are encoding. If you need to create a new profile or remove an old one, click Manage.

Profile: WM8 Video for LAN Delivery (100Kbps, 240x180, 15fps)

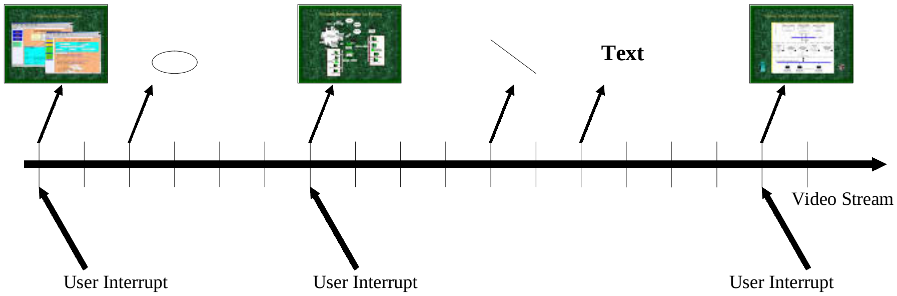
Description:

- WM8 Video for LAN Delivery (100Kbps, 240x180, 15fps)
- WM8 Video for Modem Delivery (56Kbps, 160x120, 15fps)
- High motion video for broadband NTSC (1500 Kbps total)
- High motion video for broadband NTSC (384 Kbps)
- High motion video for broadband NTSC (768 Kbps)
- Screen capture (live) for dial-up modems (28.8 Kbps)
- Screen capture for dial-up modems (28.8 Kbps)
- Screen capture for e-mail and dual-ISDN (128 Kbps)
- Video for broadband film content (1500 Kbps total)
- Video for broadband film content (768 Kbps)
- Video for broadband NTSC (1500 Kbps total)
- Video for broadband NTSC (2 Mbps total)
- Video for broadband NTSC (256 Kbps)
- Video for broadband NTSC (384 Kbps)
- Video for broadband NTSC (768 Kbps)

documents must have compatibility for different network environments. The proposed system provides several encoding modes for different end users (as shown in Figure 4).

- *Synchronization Controller:* The synchronization mechanism is based on time-stamps, as they are recorded while the user is operating the PowerPoint software. These time-stamps are embedded within an Advanced Streaming Format (ASF) record. During the playback, these stamps trigger windowing events, such as line drawing, slide changing, text drawing, etc. (Figure 5). On the other hand, the ASF record can accept user interrupts. Thus, the presentation can be restarted from any slide in the video. The time-stamp-based synchronization mechanism and the ASF from Microsoft are now widely used in similar projects and products.

Figure 5. Time-stamps and window messages



- *Post-Processing Manager:* Given a video file, if the author would like to pick a particular clip from the original file, the usual method is to use a commercial video-editing tool to cut the clip and recode the file. Another example is that if the author needs two clips from two different files, and he has a succession of operations. From our experience, the operations are complicated and not suitable for an amateur. Post-processing manager is responsible for reusing and reorder the existing video files. We also have a user-friendly interface, and the author only drags and drops the clips to playing sequence (Figure 6).
- *Multimedia Presentation Browser:* The final layout of the presentation is also left to the author. He can choose the background of the multimedia object in presentation browser in playing. Here the layout also includes some descriptions of the video sequence (Figure 7). The interface of the presentation browser has a high influence on the audience. If the quality of presentation is not sufficient and interesting, you can't capture the readers' attention to deliver your message (Lisle, Isensee, & Dong, 1998).

Figure 6. Drag and drop feature

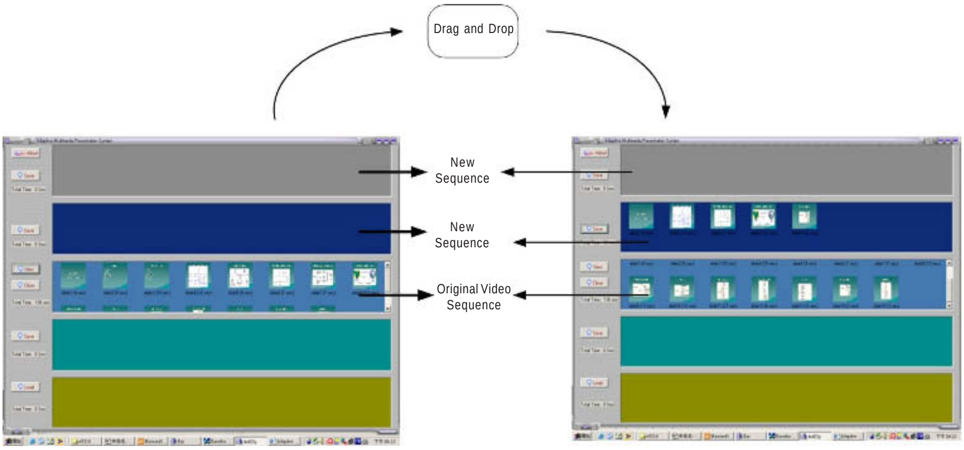
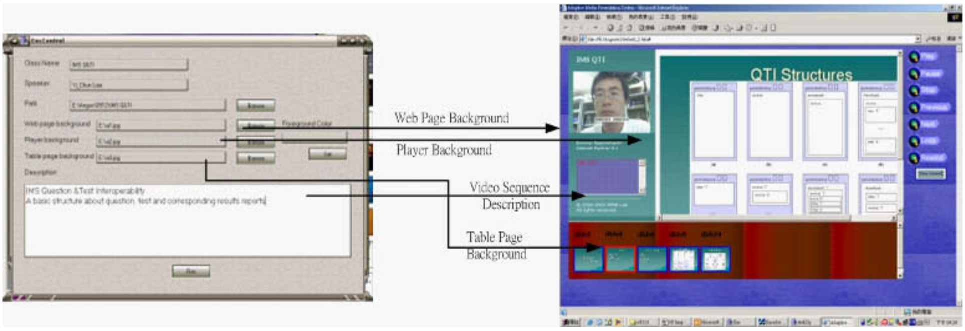
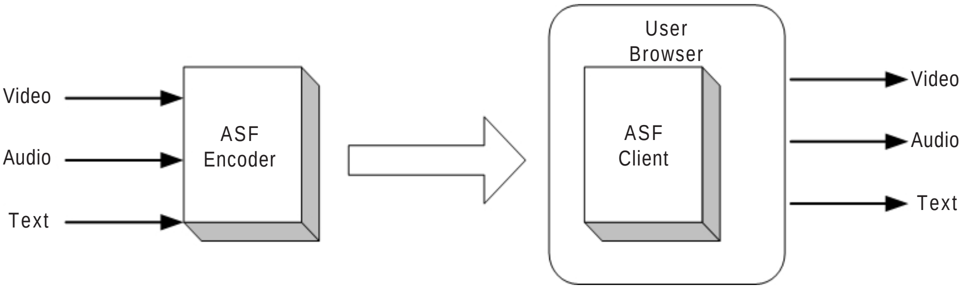


Figure 7. Final layout of Browser



IMS Question &Test Interoperability
A basic structure about question, test and corresponding results reports

Figure 8. Stream delivery and presentation of ASF file



Technical Support

In our system, we used the Windows Media Codec as the media-streaming solution. Windows Media Codec for creating advance stream format (ASF) (Windows Media Encoder, 2004) content uses compression/decompression algorithms (codec) to compress audio and/or video media, either from live sources or other media formats, to fit on a network's available bandwidth. The ASF is a data format for streaming audio and video content, images, and script commands in packets over a network (as shown in Figure 8). ASF content can be an .asf file or a live stream generated by Windows Media Encoder. ASF content that is in the process of being delivered over a network is called an ASF stream.

A Multiple-Level Content Tree for Abstraction

Given a Web-based multimedia presentation, the corresponding multiple-level content trees can be constructed, as shown in Figure 9. Teaching material can be taken as a multimedia presentation (e.g., collection of text, video, audio, image, etc.) with some kind of sequence fashion. The multiple-level content tree-approach may be used to arrive at an efficient summarizing method. A content tree is a finite set of one or more nodes such that there is a particularly designated node called the root. The level of a node is

Figure 9. Example of multiple-level content tree

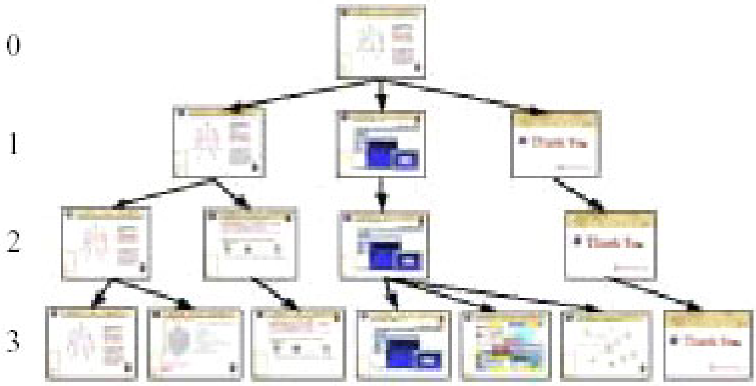
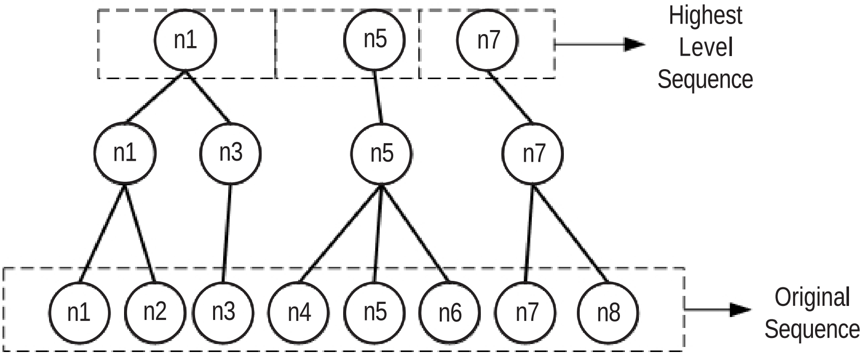


Figure 10. Presentation tree structure



defined by initially letting the root be at level 0. If a node is at level q, then its children are at level q+1. Since a node is composed of a presentation segment, the siblings ordered from left to right represent a presentation with some sequence fashion. The higher level gives the longer presentation. Consequently, this approach gives flexible teaching material; accordingly, it is a very good fit for the Web-based multimedia presentation.

The Abstractor utilizes the content tree to organize the information, as shown in Figure 9. There are a number of primitive operations that can be applied to the content tree. Algorithms to initialize a content tree, attach a node, detach a node, and calculate the presentation time at a given level, have been developed. The content tree serves as an internal data structure for all these operations.

An Example of Building Tree Structure

We define a unit in the presentation sequence. This unit will be used as a node in a tree structure (shown in Figure 10). The importance of presentation units is viewed from the lowest level to the highest level as most important. According to the Bottom-Up method, the node in the lowest level should be added first. Given an original playing sequence, there are many nodes in which some nodes are necessary and some nodes have lower priority. Using the higher priority nodes, we construct a new level. In the tree structure, a node not only has priority respectively but also includes a time value. Time value is presentation time in a presentation sequence. So we can get total presentation time of an original sequence: *n1.TimeValue+n2.TimeValue+…+n8.TimeValue* (Figure 10). The time of the highest level sequence is shortest; the lowest level sequence is longest. The adaptability of the presentation time is also a key function in such an authoring system.

Process Flow of Authoring

In this section, we will focus on the synchronization of the multimedia stream object. ”Synchronization” means that when we record multimedia streams, we not only store the object information, but also capture the temporal relations between media objects in our system. There are two steps that we need in the process of streams synchronization: (1) the user Setup phase; and (2) the recording and integrating objects synchronization

Table 1. Message transaction

Transaction(message)	Operations		
	From	**To**	**Function**
Call_Configure	User	WModer	Request for a new service
EncodeToFile_Request	WMOutput	Archies	Manage media data
Archies_Message	Archies	WMSource	To establish contact with WModer
EncodeFromDevice_Request	WMSource	Video,Audio	Manage media device
Device_Message	Video,Audio	WMProfile	To establish contact with WModer
Profile_Request	WMProfile	Archies	Manage profile
Archies_Message	Archies	WModer	To establish contact with WModer
Start_Request	WModer	Indexer Maker	Request for synchronization
MarkerInsert_Request	Indexer Maker	Archies	Insert Markers into media data
Archies_Message	Archies	Marker Maker	To establish contact with Marker Maker
IndexerInsert_Request	Marker Maker	Archies	Insert Indexers into media data
Archies_Message	Archies	Publisher	To establish contact with Marker Maker
Publisher_Request	Publisher	User	Request for replay

Archies: Media stream server

control phase. The processes will be illustrated with two time-based scenarios. The messages between the objects of illustrations are defined in Table 1.

The components of the Video/Audio codec:

- *WMOutput*: for output of the media stream.
- *WMSource:* for integration of input devices.
- *WMProfile*: for managing the compression bit rate.
- *WModer*: for converting between the different streams.

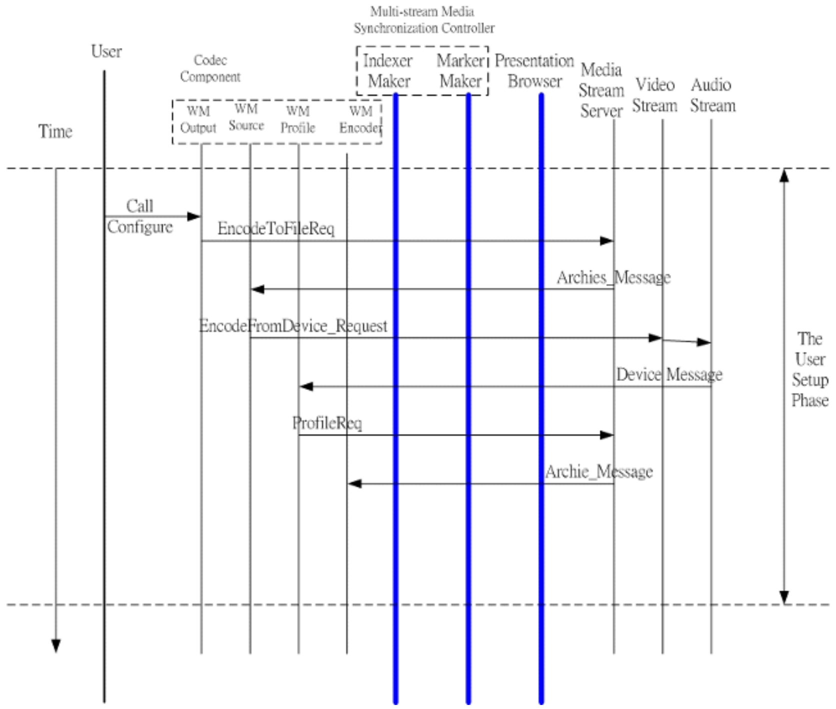
The components of the synchronization controller:

- *Indexer Maker:* inserting the Indexer into the completed multi-stream file for the end users to control slide index.
- *Multi-Stream Marker Maker:* inserting the Marker into the raw video data for synchronizing the multi-stream (Figure 2).

User Setup Phase

This phase enables the operations of configuration of authoring processes. The major component of the Video/Audio codec is used to facilitate the users' selection of sources/devices that will be encoded (as shown in Figure 11). The subcomponents of the Audio/Video codec are the major providers. *WMsource* will be called immediately to drive

Figure 11. Scenario of the user setup phase



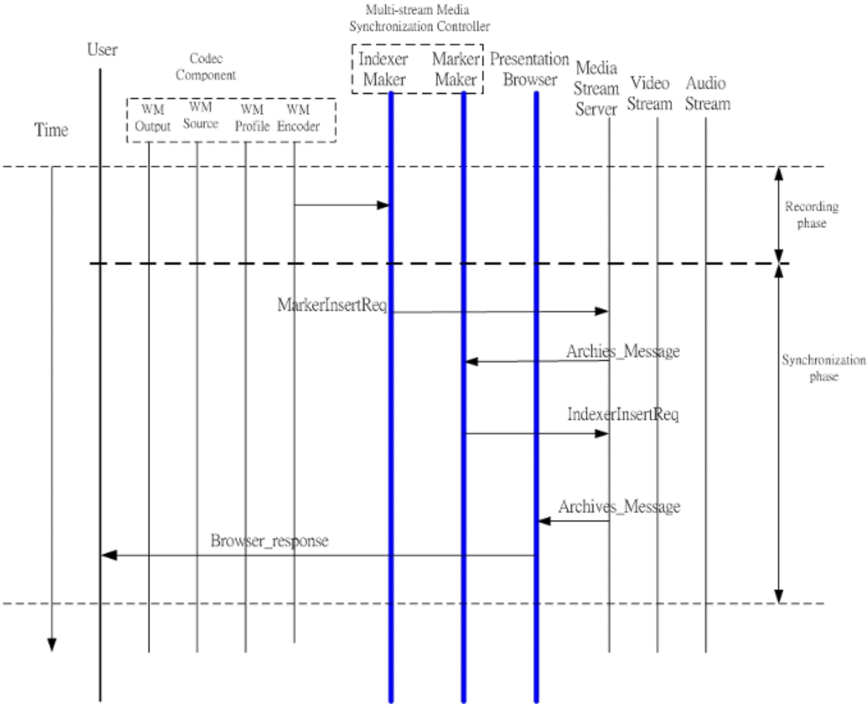
the Audio/Video devices or to open the existing files. In this phase, *WMOutput* will be used to store the output media documents after finishing the processes of recording. According to the *WMProfile*, the author uses WMEncoder to produce the well documents (ASF file format).

Recording and Integrating Objects Synchronization Control Phase

The phase during authoring processes is illustrated in Figure 12. When the user finishes the configuration phase, the multimedia media synchronization controller is invoked to coordinate an audio stream, a video stream and a script stream and to create the source group. A user can create multiple source groups, but all must contain the same media types and only one can be encoded at a time. Then *MediaEncoder* captures a multimedia stream from a file or input device. By using the audio/video codec components, the system will either loads streams from a media file or capture live media streams from an input device.

Then the synchronization controller will select the indexer profiler for the re-encoding presentation session. An indexed profile specifies a synchronization codec, and identifies the number and bit rate of the encoded output streams. Only one indexer profile can be assigned to all of the source groups in the encoding presentation session. The synchronization controller will integrate the source groups and the profile into a script command by using the insertion facility of the marker maker to perform the

Figure 12. Scenario of the synchronization phase



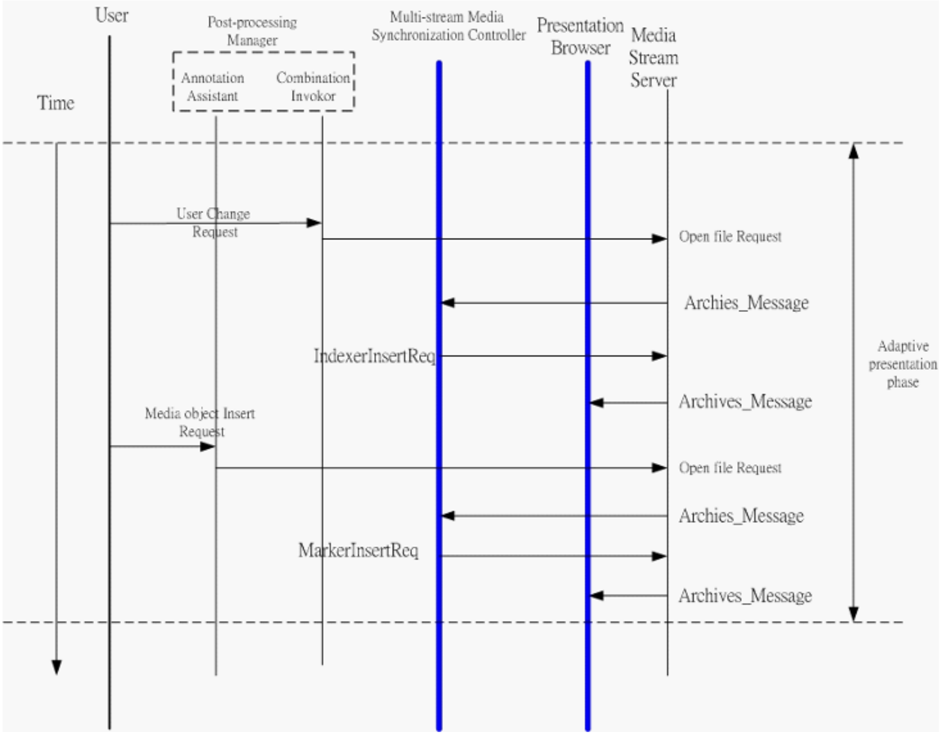
synchronization processes. The script command is a grouped special instruction that is included in presentation data streams and delivered to the client together. The presentation browser component in the client side will parse the script commands to a device or an application, and then interprets them. Script commands are used for such tasks as interactive exploration of information presentation.

Adaptive Post-Processing Processes

To address the problem from our investigation, this section will describe the adaptability and the dynamics of our system. For the dynamics of the user's requirements, the multimedia presentation system should enable the user to weave, reconstruct, or combine the existing multimedia documents. This feature has been illustrated in section 2.2.1. Figure 13 shows the processes that the user would use to combine two files. The *Combination Invokor* is called upon to open a file with the *IndexerProfile*. Here the two files are limited to containing the same media types, such as a presentation video stream, a presentation audio stream, a slide stream, and a script command stream. Then, the synchronization controller is used to re-synchronize the new presentation (Figure 14).

Another contribution of our system is that it considers the scalability of the presentation. We involve the *Annotation Assistant* to insert additional resources such as a video clip, an image, a text file, or a URL. The process is that a user requests the object insertion. This request will add a media object into a presentation file. These auxiliary

Figure 13. Process of dynamic change and adaptive insertion



media objects will be attached to a presentation unit (a node in tree structure). For an audience, presentation data is a major sequence in playing; those auxiliary objects are regarded as a sun-sequence, and it may be ignored (Figure 15).

Figure 14. Combination of two presentation resources

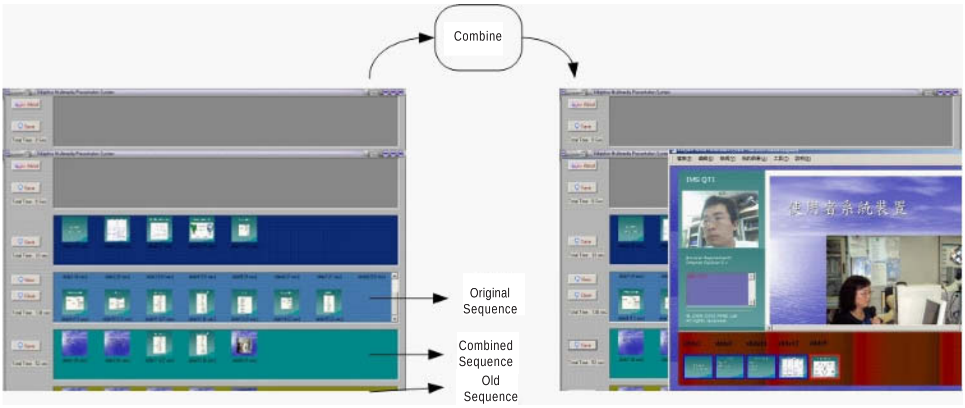
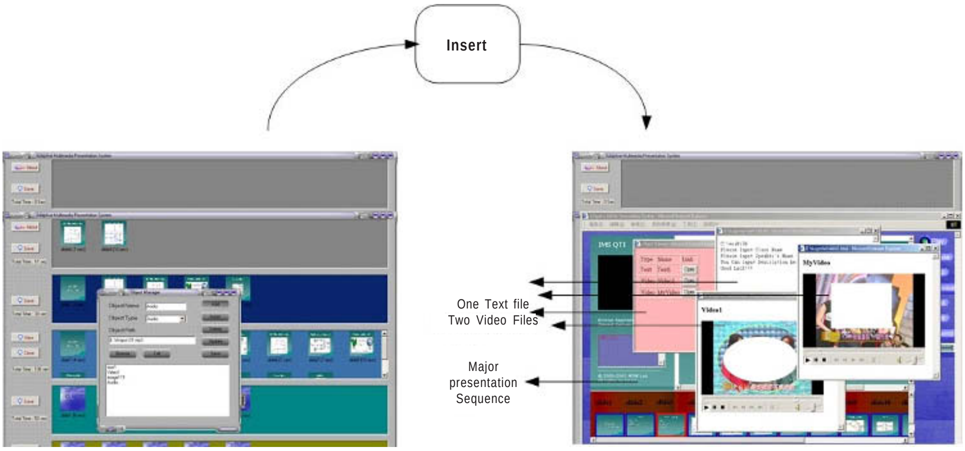


Figure 15. Inserting the auxiliary media stream



System User Interface

The majority of the system user interface consists of the configuration module. The configuration module provides the user with the facilities to select the sources/devices from which he or she would like to encode and to select to the preferred output of the encoded content. The user can either encode a media file (video/audio) or use attached devices (video camera or microphone) to produce the orchestrated media contents. In addition, the user can select either to broadcast their encoded content in real time after configuring the server HTTP port and the URL for Internet/LAN connections. The user can select the profile that best describes the content being encoding. This profile means the different bandwidth will be configured. The higher bit-rate means the content will be encoded to a higher resolution content. The different bandwidth profile selection window was also included. Figure 16(a) shows the recording and editing interface of the system, with automatic uploading and broadcasting lectures shown in Figure 16(b).

Figure 16(a). A multi-level content tree of the Web-based

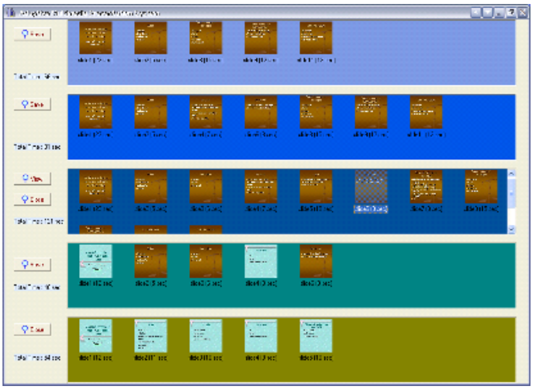
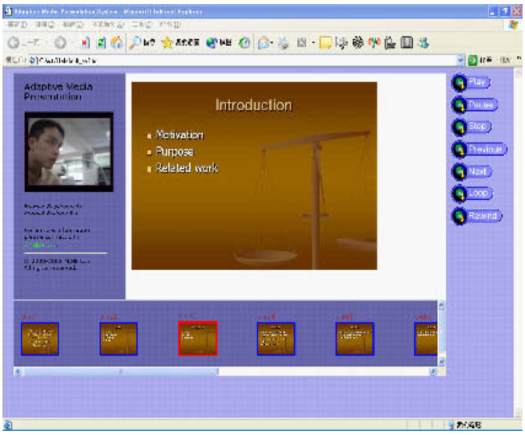


Figure 16(b). Video presentation playback with a synchronized control mechanism



CONCLUSIONS

We have addressed the issues involve in allowing a user to add time-stamp information during the authoring phase. In other situations, the issue involves the scalability and the dynamics. We have not only proposed a framework of a Web-based multimedia presentation system, but also implemented this system. We use the time-stamp as the media synchronization model. With the easy-to-use interfaces, the configuration and the operation steps of the system are clear and definite. The main goal of our system is to provide a feasible method to record and represent a lecture/presentation in the distributed environment. Using the browser with the Windows Media Services allows users to view live video of the teacher giving his or her speech, along with synchronized images of presentation slides and all the annotations/comments. The system can be used for general-purpose presentations as well as distance learning, advertisement, and others.

REFERENCES

Chang, F., Wang, T.-H., Chan, Y.-W., Deng, L. Y., Lee, I., & Wang, Y. (2001). Internet-based distance education: An agent-based distance learning project in Tamkang University as an example. *Proceedings of the 7th International Conference on Distributed Multimedia Systems (MMS 2001)*, Taipei, Taiwan.

Chang, S. K., Hassanein, E., & Hsieh, C. Y. (1998). A multimedia micro-university. *IEEE Multimedia Magazine*, *5*(3), 60-68.

Lisle, L., Isensee, S., & Dong, J. (1998). Developing a multimedia product for the World Wide Web. In *Designing effective and usable multimedia system*. Norwell, MA: Kluwer Academic Publishers.

Liu, K.-Y., Huang, N., Wu, B.-H., Chu, W.-T., & Herng-Yow, C. (2002). WSML system: Web-based synchronization multimedia lecture system. *ACM Multimedia'02*, France.

Microsoft, Inc. (2001). Windows Media encoder, Windows Media SDK. *http://www.microsoft.com*

Mukhopadhyay, S., & Smith, B. (1999). Passive capture and structuring of lectures. *ACM Multimedia'99*, Orlando, FL.

Realplayer Encoder (2002). RealNetworks Inc. Retrieved on the World Wide Web at: *http://www.real.com*

Shih, T., & Deng, L. (2000). Distributed multimedia presentation with floor control mechanisms in a distance learning system. *Proceedings of the 2000 International Conference on Information Society in the 21st Century: Emerging Technologies and New Challenges,* Japan.

Shih, T., Deng, L., Wang, J., & Liou, A. (2001). Maintaining persistent look-and-feel for roaming student with mobile agent in distance learning. *Proceedings of the 7th International Conference on Distributed Multimedia Systems (MMS 2001),* Taipei, Taiwan.

Steinmetz, R., & Nahrstedt, K. (1995). *Multimedia: Computing, communications and applications* (pp. 481-542). Englewood Cliffs, NJ: Prentice Hall.

Sun Microsystems, Inc. (2004). Java Media Framework (JMF) API. Retrieved on the World Wide Web at: *http://java.sun.com/products/javamedia/jmf/index.html*

World Wide Web Consortium (W3C) (2004). Synchronized multimedia integration language (SMIL), 2.0 Specification. Retrieved on the World Wide Web at: *http://www.w3.org/*

Section IV

Video Shot Boundary Detection

Chapter VIII

Video Shot Boundary Detection

Waleed E. Farag, Zagazig University, Egypt

Hussein Abdel-Wahab, Old Dominion University, USA

ABSTRACT

The increasing use of multimedia streams nowadays necessitates the development of efficient and effective methodologies for manipulating databases storing this information. Moreover, in its first stage, content-based access to video data requires parsing of each video stream into its building blocks. The video stream consists of a number of shots, each one a sequence of frames pictured using a single camera. Switching from one camera to another indicates the transition from a shot to the next one. Therefore, the detection of these transitions, known as scene change or shot boundary detection, is the first step in any video-analysis system. A number of proposed techniques for solving the problem of shot boundary detection exist, but the major criticisms to them are their inefficiency and lack of reliability. The reliability of the scene change detection stage is a very significant requirement because it is the first stage in any video retrieval system; thus, its performance has a direct impact on the performance of all other stages. On the other hand, efficiency is also crucial due to the voluminous amounts of information found in video streams. This chapter proposes a new robust and efficient paradigm capable of detecting scene changes on compressed MPEG video data directly. This paradigm constitutes the first part of a Video Content-based Retrieval (VCR) system that has been designed at Old Dominion University. At first, an abstract representation of the compressed video stream, known as the DC *sequence, is extracted, then it is used as input to a Neural Network Module that performs the shot boundary-detection task. We have studied experimentally the performance of the proposed paradigm and have achieved higher shot boundary detection and lower false alarms rates than other techniques. Moreover, the efficiency of the system outperforms other approaches by several times. In short, the experimental results show the superior efficiency and robustness of the proposed system in detecting shot boundaries and flashlights — sudden lighting variation due to camera flash occurrences — within video shots.*

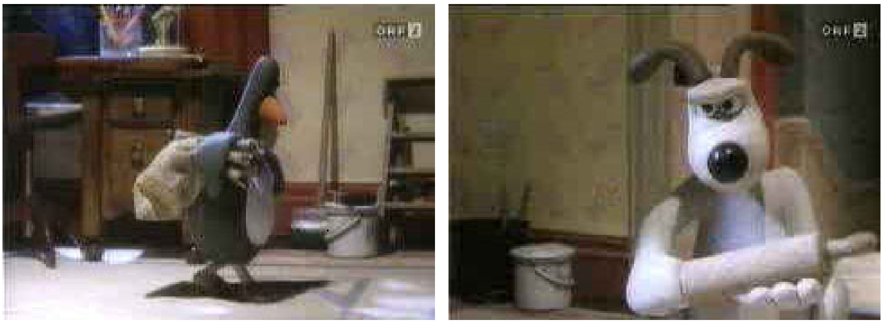
INTRODUCTION

The recent explosive growth of digital video applications entails the generation of vast amount of video data; however, the technologies for organizing and searching video databases are still in their infancy. The first step in indexing video databases (to facilitate efficient access) is to analyze the stored video streams. Video analysis can be classified into two stages (Rui, Huang, & Mehrotra, 1998): shot boundary detection and key frames extraction. The purpose of the first stage is to partition a video stream into a set of meaningful and manageable segments, whereas the second stage aims to abstract each shot using one or more representative frames. We will address the problem of shot boundary detection in this chapter while the problem of selecting key frames from segmented shots will be dealt with in another chapter.

In general, successive frames in motion pictures bear great similarity among themselves, but this generalization is not true at the boundaries of shots. A frame at a boundary point of a shot differs in background and content from its successive frame that belongs to the next shot (Figure 1). In a nutshell, two frames at a boundary point will differ significantly as a result of switching from one camera to another, and this is the basic principle upon which most automatic algorithms for detecting scene changes depend.

Due to the huge amount of data contained in video streams, almost all of them are transmitted and stored in compressed format. While there are large numbers of algorithms for compressing digital video, the MPEG format (ISO/IEC, 1999; LeGall, 1991; Mitchell, Pennebaker, Fogg, & LeGall, 1997) is the most famous one and the current international standard. In MPEG, spatial compression is achieved through the use of a Discrete Cosine Transform (DCT)-based algorithm similar to the one used in the JPEG standard (Pennebaker & Mitchell, 1993; Wallace, 1991). In this algorithm, each frame is divided into a number of blocks (8X8 pixel), then the DCT transformation is applied to these blocks. The produced coefficients are then quantized and entropy-encoded, a technique that achieves the actual compression of the data. On the other side, temporal compression is accomplished using a motion compensation technique that depends on the similarity between successive frames on video streams. Basically, this technique codes the first picture of a video stream (I frame) without reference to neighboring frames, while successive pictures (P or B frames) are generally coded as differences to that reference frame(s). Considering the large amount of processing power required in the manipulation of raw digital video, it becomes a real advantage to work directly upon compressed data and avoid the need to decompress video streams before manipulating them.

Figure 1. Differences in content and background between two successive frames at a shot boundary



In this chapter, a novel and robust algorithm for detecting shot boundaries is introduced (Farag & Abdel-Wahab, 2001, 2002). The first module of the algorithm extracts an abstract description of a video frame that is known as the DC frame. DC, a term from Electrical Engineering that stands for Direct Current, is the first coefficient of the discrete cosine transform that is proportional to the average intensity of the block. A DC frame is generated by using the DC component of each DCT block and neglecting the AC coefficients, the other DCT terms. The DC coefficient has been chosen due to its central role. A series of these frames is called the DC sequence. This way, we managed to abstract the MPEG video stream by deriving its DC sequence without the need to decode the original stream. A modified version of the DC sequence is then used as input to a Neural Network Module (NNM) that performs the task of detecting shot boundaries found into that stream.

First, we select some of the representative video clips from our database (with known shot boundary positions) as a training set and derive the DC sequences for them. These sequences are then merged to form one DC sequence to be used in training the neural network. The training is performed as an off-line activity until the network reaches an acceptable training error level. The result of the training process is a group of connection weights that store the appropriate mapping from the input space to the output space. These weights are stored to be used in the recall phase. Now, the process of detecting shot boundaries of a video clip is a matter of deriving the DC sequence of that clip. Then, using it as input to the network to recall the information stored into the connection weights during what is know as the recall phase of the neural network. That recall phase yields a set of shot boundary positions (if any).

The rest of this chapter is organized as follows. The next section briefly reviews a number of related approaches for detecting scene changes. The concept of representing the video stream using its DC sequence is introduced after that, along with our algorithm for extracting that sequence. Then, the design of the NNM that is used to detect scene cuts will be expounded. Experimental results are given near the end of the chapter, followed by conclusion.

RELATED WORK

Video data are rich sources of information and in order to model these data, the information content of the data has to be analyzed. As mentioned before, video analysis is divided into two stages. The first stage is the division of the video sequence into a group of shots (shot boundary detection), while the second stage is the process of selecting key frame(s) to represent each shot. Generally speaking, there are two trends in the literature to segment video data. The first one works in the uncompressed domain, while the other one works in the compressed domain. The first trend will be discussed first.

Methods in the uncompressed domain can be broadly classified into five categories: template-matching, histogram-based, twin-comparison, block-based, and model-based techniques. In template-matching techniques (Hampapur, Jain, & Weymouth, 1994; Zhang, Kankanhalli, Smoliar, & Tan, 1993), each pixel at the spatial location (i,j) in frame f_m is compared with the pixel at the same location in frame f_n , and a scene change is declared whenever the difference function exceeds a pre-specified threshold. Using this metric, it becomes difficult to distinguish between a small change in a large area and a large change in a small area. Therefore, template-matching techniques are sensitive to noise, object motion, and camera operations. One example of the use of histogram-based techniques is presented in Tonomura, (1991), where the histogram of a video frame and a difference function (S) between f_n and f_m are calculated. If S is greater than a threshold, a cut is declared. That technique uses equation (1) to calculate the difference function and declare a cut if the function is greater than a threshold.

$$S(f_m, f_n) = \sum_{i=1}^{N} \left| H(f_m, i) - H(f_n, i) \right| \qquad \textbf{(1)}$$

The rationale behind histogram-based approaches is that two frames that exhibit minor changes in the background and object content will also show insignificant variations in their intensity/color distributions. In addition, histograms are invariant to image rotation and change slowly under the variations of viewing angle, scale, and occlusion (Swain & Ballard, 1991). Hence, this technique is less sensitive to camera operations and object motion compared to template matching-based techniques.

Another technique that is called twin comparison has been proposed by Zhang, Kankanhalli, Smoliar, and Tan (1993). This technique uses two thresholds, one to detect cuts and the other to detect potential starting frames for gradual transitions. Unfortunately, this technique works upon uncompressed data and its inefficiency is the major disadvantage. A different trend to detect shot boundary is called a block-based technique (Idris & Panchanathan, 1997) that uses local attributes to reduce the effect of noise and camera flashes. In this trend, each frame f_m is partitioned into a set of r blocks and rather than comparing a pair of frames, every sub-frame in f_m is compared with the corresponding sub-frame in f_n. The similarity between f_n and f_m is then measured. The last shot boundary-detection technique working upon uncompressed data is termed model-based segmentation (Idris & Panchanathan, 1997), where different edit types, such as cuts, translates, wipes, fades, and dissolves are modeled by mathematical functions. The essence here is not only identifying the transition but also the type of the transition.

On the other hand, methods for detecting shot boundaries that work in the compressed domain have been investigated. The main purpose of works in this trend is to increase efficiency. Again, we can roughly divide these methodologies into three categories. The first category (Chen, Taskiran, Albiol, Delp, & Bouman, 1999; Lee, Kim, & Choi, 2000; Yeo & Liu, 1995b) uses DCT coefficients of video-compression techniques (Motion JPEG, MPEG, and H.261) in the frequency domain. These coefficients relate to the spatial domain, hence they can be used for scene change detection. In Chen et al. (1999), shot boundary detection is performed by first extracting a set of features from the DC frame. These features are placed in a high-dimensional feature vector that is called the Generalized Trace (GT). The GT is then used in a binary regression tree to determine the probability that each frame is a shot boundary. Yeo and Liu (1995b) use the pixel differences of the luminance component of DC frames in MPEG sequences to detect shot boundaries. Lee et al. (2000) derive binary edge maps from AC coefficients and measure edge orientation and strength using AC coefficients correlations, then match frames based on these features.

The second category makes use of motion vectors. The idea is that motion vectors exhibit relatively continuous changes within a single camera shot, while this continuity is disrupted between frames across different shots. Zhang et al. (1993) have proposed a technique for cut detection using motion vectors in MPEG videos. This approach is based on counting the number of motion vectors M in predicted frames. In P-frames, M is the number of motion vectors, whereas in B-frames, M is the smaller of the counts of the forward and backward nonzero motion. Then, M<T will be an effective indicator of a camera boundary before or after the B-and P-frames, where T is a threshold value close to zero.

The last category working into the compressed domain merges the above two trends and can be termed hybrid Motion/DCT. In these methods, motion information and the DCT coefficients of the luminance component are used to segment the video (Meng, Juan, & Chang, 1995).

Other approaches that cannot be categorized into any of the above two classes are reviewed below. Vasconcelos and Lippman (1997) have modeled the time duration between two shot boundaries using a Bayesian model and the Weibull distribution, then they derived a variable threshold to detect shot boundaries. A knowledge-based approach is proposed by Meng, et al. (1995), where anchorperson shots are found by examining intrashot temporal variation of frames. In order to increase the robustness of the shot boundary detection, Hanjalic and Zhang (1999) proposed the use of statistical model to detect scene changes.

In summary, techniques that work upon uncompressed video data lack the necessary efficiency required for interactive processing. On the other hand, although the other techniques that deal directly with compressed data are more efficient, their lack of reliability is usually a common problem. To address these shortcomings, we proposed a reliable and very efficient technique to solve the problem of shot boundary detection of video data.

DC SEQUENCE EXTRACTION

Compressed MPEG files are the most commonly used forms for storage and transmission of audiovisual information. At the receiver side, the MPEG files need to be decoded in order to properly display the received information. In general, the decoding process is a highly time- consuming task, especially if it is done using software only and without any specialized hardware units. Consequently, working upon compressed data is a real advantage. The first step in any video retrieval system is to analyze the input files (MPEG files, in our case) to detect shot boundaries. In order to achieve this objective while trying to enhance the efficiency of the proposed algorithm, our methodology attempts to detect shot boundaries directly from compressed data and avoids the requirement of decoding the video file first. This goal is achieved through two steps:

- Generating what is known as the DC sequence from the compressed data; and
- Using the above-generated sequence in training a feedforward neural network that is used afterwards during its recall phase to detect shot boundaries.

Deriving Formulas for Extracting the DC Sequence

This section formulizes the problem of extracting the DC sequence from MPEG files and introduces our proposed solution to solve it. To encode MPEG files, each frame in the original video file is divided into 8X8 blocks; then the DCT transform is applied to individual blocks. In addition to the motion information, the encoded transform coefficients are the main constituents of the compressed file. The first coefficient of the DCT of a block is termed the DC coefficient, and it is directly proportional to the average intensity of that block. The main concept is to use these DC coefficients to derive an abstract description of a frame directly from the compressed data without the need for decoding. Each block will be represented by only one term (its average intensity derived from the DC term), and the composition of these terms will form what is called a DC frame. A sequence of such frames is termed the DC sequence. This sequence still bears a high similarity to the original frame sequence (Yeo & Liu, 1995a), with the added advantage of the ability to directly and very efficiently derived it from compressed data.

The general idea of using the DC sequence has been proposed by Shen and Delp (1995) and Yeo and Liu (1995b). The extraction of the DC frame from an I frame is trivial and can be calculated for each block as follows:

$$DC_I = \frac{1}{8} DC_{encoded} \qquad \textbf{(2)}$$

Where:

DC_I: The derived DC for a specific block.
$DC_{encoded}$: The encoded value in that block (the first coefficient of the DCT).

As shown above, deriving the DC terms from I pictures is a trivial task, but for B and P pictures, it is not the same. One proposed solution in Shen and Delp (1995) is to calculate the DC of a block in B or P frames using equation (3). Equation (4), given below, is the mathematical definition of the term DC_{ref}.

$$DC_{P/B} = DC_{ref} + DC_{diff} \tag{3}$$

$$DC_{ref} = \frac{1}{64}\sum_{i=0}^{3} N_i DC_i \tag{4}$$

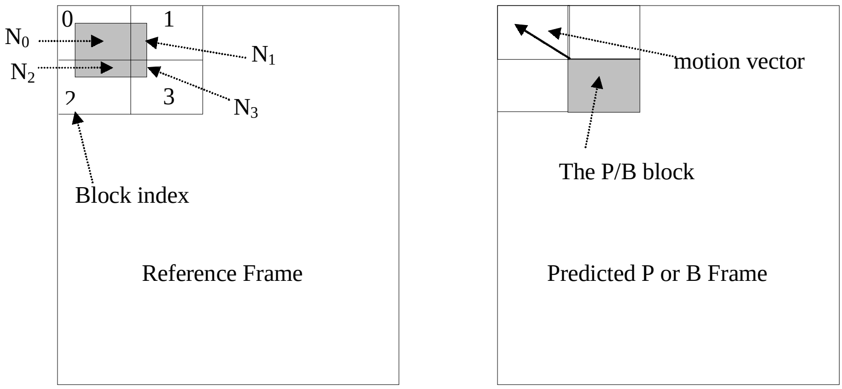
Where:

DC_{diff}: The encoded DC coefficient of a block in a P or B frame.
DC_{ref}: The average of the DC coefficients of the reference frame blocks (at max four) overlapping with the predicted block.
N_i : The intersecting area between the block in the P or B frame and the ith block in the reference frame (Figure 2).
DC_i: The DC coefficient of the ith block in the reference frame.

Note that, right horizontal displacement and downward vertical displacement are considered positive displacements in MPEG terminology (Mitchell et al., 1997). Given the information in MPEG data and the proposed formulas described above, what is required is to determine two pieces of information in order to derive the DC sequence properly; these are:

- The intersecting areas of the blocks in the reference frame with the predicted block in the P or B frame. These areas are denoted N_0-N_3.
- The row and column indexes of each of the four intersecting blocks to be used in determining DC_i values, given the row and column indexes of the predicted block.

Figure 2. Four intersecting blocks in the reference frame are to the left, while the right shape shows the predicted block and the motion vector



By analyzing the geometry of Figure 2, the following formulas are derived to calculate the four areas. It is important to note that a different set of formulas is used for each combination of the signs of the motion vectors.

1. Case of +ΔX and +ΔY (right horizontal and down vertical displacements)

$$N_0 = (L - |\Delta X|) * (L - |\Delta Y|)$$

$$N_1 = |\Delta X| * (L - |\Delta Y|)$$

$$N_2 = (L - |\Delta X|) * |\Delta Y|$$

$$N_3 = |\Delta X| * |\Delta Y|$$

2. Case of +ΔX and -ΔY

$$N_0 = (L - |\Delta X|) * |\Delta Y|$$

$$N_1 = |\Delta X| * |\Delta Y|$$

$$N_2 = (L - |\Delta X|) * (L - |\Delta Y|)$$

$$N_4 = |\Delta X| * (L - |\Delta Y|)$$

3. Case of -ΔX and +ΔY

$$N_0 = |\Delta X| * (L - |\Delta Y|)$$

$$N_1 = (L - |\Delta X|) * (L - |\Delta Y|)$$

$$N_2 = |\Delta X| * |\Delta Y|$$

$$N_3 = (L - |\Delta X|) * |\Delta Y|$$

4. Case of -ΔX and -ΔY

$$N_0 = |\Delta X| * |\Delta Y|$$

$$N_1 = (L - |\Delta X|) * |\Delta Y|$$

$$N_2 = |\Delta X| * (L - |\Delta Y|)$$

$$N_4 = (L - |\Delta X|) * (L - |\Delta Y|)$$

Where:

ΔX: The horizontal component of the motion vector.
ΔY: The vertical component of the motion vector.
L: The length of the side of a block (all blocks are squares and have the same dimension, which is 8X8).

Three types of DC sequences — one for each color components used in MPEG (Y, Cr, Cb) — are extracted, but we use only the Y component because human eyes are more sensitive to the luminance component than the chrominance ones (Mitchell et al., 1997). If the block is bi-directionally predicted (has both forward and backward motion vectors), we propose to apply equation (4) to both the forward and backward cases and take the average value. This is a similar technique to how the MPEG algorithm handles the reconstruction of pixel values during the decoding phase. Thus, in case of bi-directionally predicted blocks, equation (5) below will be used to evaluate DC_{ref}.

$$DC_{ref} = \frac{1}{2}\left(\frac{1}{64}\sum_{i=0}^{3} N_i DC_i(\,forward) + \frac{1}{64}\sum_{i=0}^{3} N_i DC_i(backward)\right) \qquad \textbf{(5)}$$

To specify which blocks in the reference frame will contribute to the DC_{ref} formulas, equations (4) and (5), we need to determine the row and column indexes of intersecting blocks in the reference picture and relate this information to the row and column index of the block under investigation. The influencing factors are the signs of the motion vectors. To derive the required relations, an investigation of the position of the considered block in relation to the other four intersecting blocks is performed. This investigation yields four sets of relations, one for each possible combination of motion vectors signs. These relations for row and column indexes of the four overlapping blocks (B_0-B_3) are listed in Table 1, assuming the row and column of the predicted block are R and C respectively.

By knowing the intersecting areas and the positions of the overlapping blocks in the reference frame, both equations (4) and (5) can be evaluated. One issue still needs consideration, that is, the case of large motion vectors, the focus of the next section.

Handling the Case of Large Motion Vectors Magnitudes and Other Checks

The results in Table 1 assume that the magnitude of the Motion Vector (MV) cannot exceed the length of the block side (*L*), but in actual MPEG-coded files, this happens frequently and consequently the relations in TABLE 11 need to be adapted to account for such situations. To perform this adaptation [in case of magnitude (MV) > *L*], we calculate the value of a variable we called *addTerm*. This value will be added to the calculated rows and columns indexes in Table 1. In case of horizontal displacement, *DX*, the following algorithm is used to determine the value of the *addTermX*.

Table 1. Rows and columns indexes for overlapping blocks as functions of the signs of motion vectors and the position of the predicted block

	+ΔX and +ΔY		+ΔX and -ΔY		-ΔX and +ΔY		-ΔX and -ΔY	
	Row	Col	Row	Col	Row	Col	Row	Col
B_0	R	C	R-1	C	R	C-1	R-1	C-1
B_1	R	C+1	R-1	C+1	R	C	R-1	C
B_2	R+1	C	R	C	R+1	C-1	R	C-1
B_3	R+1	C+1	R	C+1	R+1	C	R	C

```
If ( abs($\Delta X$) >= L ) {
    addTermX = $\Delta X/L$
    $\Delta X = \Delta X\%L$
}
else {
    addTermX = 0
    $\Delta X = \Delta X$
}
```

The value of the *addTermX* will be added to all the calculated columns derived above in Table 1. The same procedure is applied in case of vertical displacement, *ΔY*, with the difference that its additional term, called *addTermY*, will be added to the calculated rows in Table 1. The final values of rows and columns of overlapping blocks are calculated below taking into account the case of large motion vectors magnitudes.

Determining Columns Indexes

To determine columns indexes, the following algorithm is employed.

```
If ($\Delta X > 0$)     // case of positive $\Delta X$
    c0 = c2 = C+addTermX
    c1 = c3 = C+1+addTermX
Else if ($\Delta X < 0$)    //case of negative $\Delta X$
    c0 = c2 = C-1+addTermX
    c1 = c3 = C+addTermX
Else                  // case of $\Delta X = 0$ no horizontal displacement
    c0 = c2 = C+addTermX
```

Determining Rows Indexes

To determine rows indexes, the following algorithm is employed.

```
If ($\Delta Y > 0$)     // case of positive $\Delta Y$
    r0 = r1 = R+addTermY
    r2 = r3 = R+1+addTermY
Else if ($\Delta Y < 0$)    //case of negative $\Delta Y$
    r0 = r1 = R-1+addTermY
    r2 = r3 = R+addTermY
Else                  // case of $\Delta Y = 0$ no vertical displacement
    r0 = r1 = R+addTermY
```

Special Cases

The following rules will be used to handle other special cases.

If $\Delta X = 0$ *Then* $N_1 = N_3 = 0$
If $\Delta Y = 0$ *Then* $N_2 = N_3 = 0$

If $\Delta X = 0$ *and* $\Delta Y = 0$ (the case of no motion compensation), then the DC will be calculated using equation (6).

$$DC_{P\,/\,B} = DC_{encoded} \qquad \textbf{(6)}$$

Where:

$DC_{encoded}$: The value of the DC of that predicted block embedded into the coding.

Boundary Condition Check

One important issue is to test the calculated values for row and column indexes to make sure that none of them is located outside the reference picture(s). To enforce this, the following checks (for luminance blocks) are performed for all the calculated values. Similar checks can be performed in case of chrominance blocks.

```
If ( row < 0 )
row = 0
else if ( row >= (mb_height * 2) )
row = (mb_height * 2) -1
if ( col < 0 )
col = 0
else if ( col >= (mb_width * 2) )
col =  (mb_width * 2) -1
```

Where:

row: The calculated row index.
col: The calculated column index.
mb_width: The number of macroblocks in a row.
mb_height: The number of macroblocks in a column.

Handling the Case of Skipped Macroblocks (MBs)

The MPEG algorithm employs a smart way to encode a macroblock that has all its DCT values equal to zero through the use of a macroblock increment value. The skipped macroblock can occurs only in P or B frames and cannot occur in I frames. One more condition is that the first and last macroblock in a slice has to be coded (Mitchell et al., 1997). A skipped macroblock simply means that the value of the difference that is supposed to be coded is zero, so this area is exactly the same as its corresponding one in the reference picture. Our implementation of the DC extraction algorithm has to take this skipping into account, so the following steps are applied:

- If the frame is P frame, copy the DC values of the corresponding MB from the reference picture to the current position.
- If the frame is B frame and both forward and backward predictions are used, calculate the average of the DC values in both prediction frames then copy it as in the previous step.

- If the frame is B frame and either a forward or backward prediction is used, copy the DC from the corresponding block in the reference picture to the current position.

Handling the Case of Non-Coded Blocks

Our algorithm has to also take care of another situation, that is, the case of non-coded blocks specified in the Coded Block Pattern (CBP). The MPEG algorithm uses CBP to signify whether a block in a macroblock is coded or not. A block that is not coded means that all of its DCT coefficients are zeros. For coded blocks, the normal algorithm will be used to calculate the DC sequence. Otherwise, if the block is not coded, our algorithm proceeds as follows:

- If the macroblock is Intracoded, all the blocks in a MB should be coded, so no special procedure will be taken.
- If the macroblock is Intercoded, the values of the encoded DC values are assumed to be equal to zeros in any further calculation.

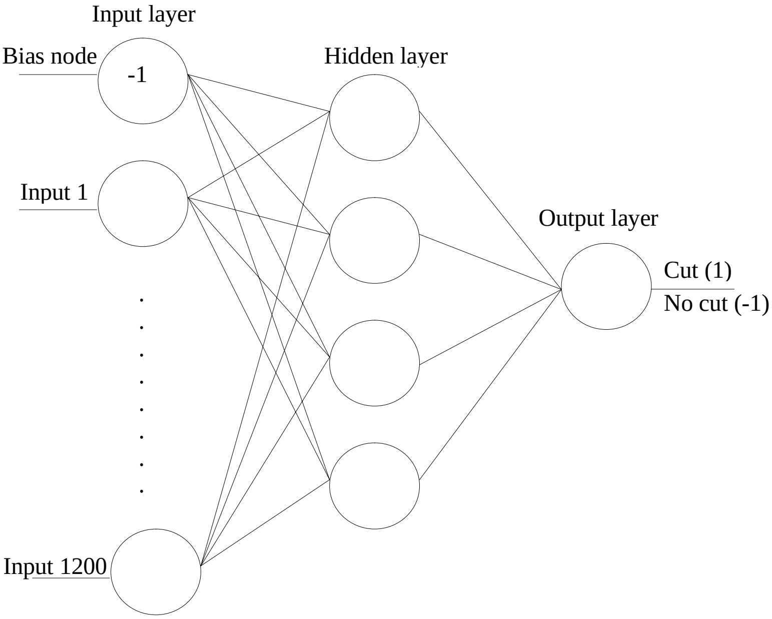
DETECTING SHOT BOUNDARIES

The DC sequence extracted in the previous section is used as input to a NNM. The choice of the neural network (Beale & Jackson, 1991; Zurada, 1992) as a shot-boundary detection methodology is based on its desirable generalization and fault-tolerance properties. Detecting shot boundaries in a video stream is a hard task, especially when we consider the multiplicities of video types (action movie, romantic movies, sports, newscast, etc.) and the different characteristics of each type. Many of the current shot boundary-detection algorithms fail to detect shot boundaries in cases of fast camera or object motion or when a large object occupies the whole scene for a while. This lack of robustness in currently available techniques motivates us to propose a robust and efficient methodology to detect scene changes in various genres of video streams. The first step in the design of the NNM is to determine a proper architecture of the network capable of solving the problem at hand (Farag, Ziedan, Syiam & Mahmoud, 1997). Three architectures are investigated. In the first one (shown in Figure 3), the differences between corresponding DC terms in two adjacent DC frames are calculated, and each difference value is used as input to a node at the input layer. Thus, for the jth element in the training/test set, the input to the input node i is given by equation (7).

$$Input_i(j) = DC_i(j) - DC_i(j+1) \qquad \textbf{(7)}$$

The second architecture diagrammed in Figure 4 uses only one node (in addition to the bias node) at the input layer. The input to that node is the sum of absolute differences between corresponding DC values in two successive DC frames. Equation (8) defines the value of the neural network input for the jth element of the training/test set, where *n* is the number of DC terms in a DC frame.

Figure 3. Structure of the first proposed neural network for shot boundary detection, assuming an input MPEG video dimension of 320x240



$$Input(j) = \sum_{i=0}^{n-1} \left| DC_i(j) - DC_i(j+1) \right| \tag{8}$$

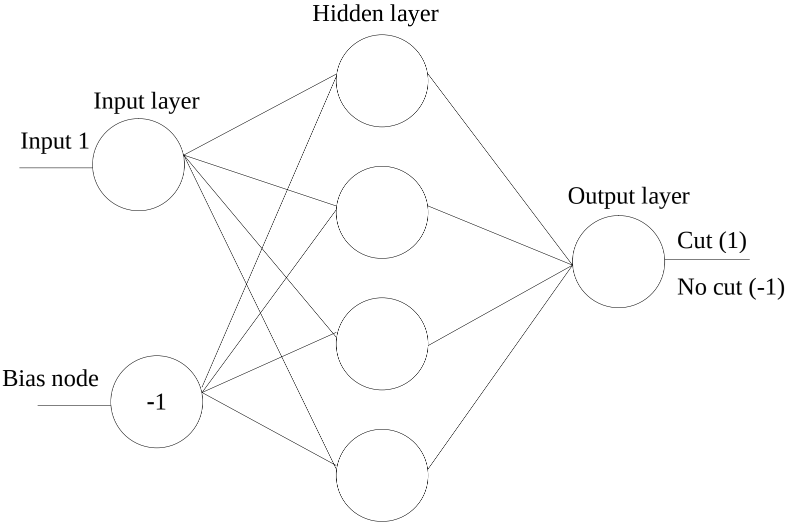
The last considered network structure is illustrated in Figure 5 and employs three input nodes. Each one of them accepts input as the previous architecture, but for DC frame difference I (difference between j and j+1), I+1, and I+2 respectively. The inputs to this structure are formulated in equations (9), (10), and (11) respectively.

$$Input_1(j) = \sum_{i=0}^{n-1} \left| DC_i(j) - DC_i(j+1) \right| \tag{9}$$

$$Input_2(j) = \sum_{i=0}^{n-1} \left| DC_i(j+1) - DC_i(j+2) \right| \tag{10}$$

$$Input_3(j) = \sum_{i=0}^{n-1} \left| DC_i(j+2) - DC_i(j+3) \right| \tag{11}$$

Figure 4. Structure of the second proposed neural network for shot boundary detection

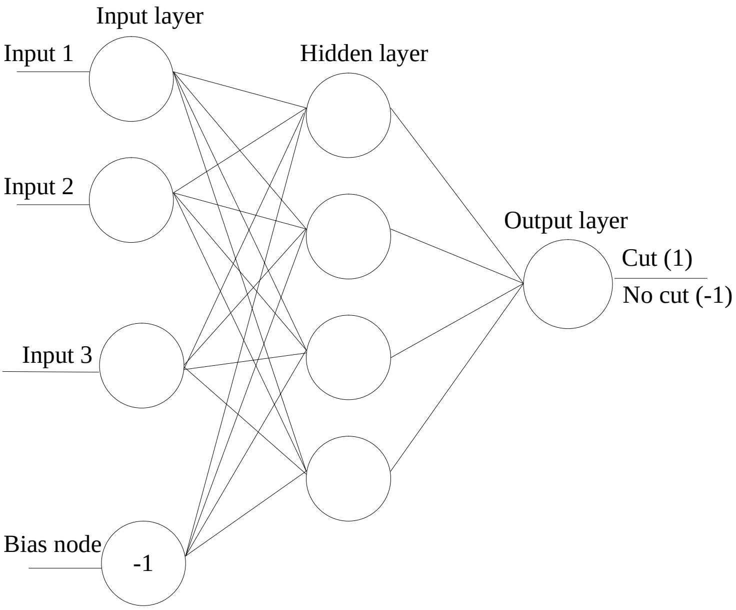


The actual difference among the three architectures is the dimension of the pattern space presented to the network in order for it to learn the required classification task. In the first case ,the dimension of the pattern space is very large (it depends upon the input MPEG dimension); this implies complex network and longer training and recall time. In the other two architectures, the input dimension is small (1 and 3, respectively). Our evaluation of these three architectures yields the following remarks:

- The dependence of the first architecture upon the dimension of the input MPEG clip results in presenting a difficult problem that has a very large input space dimension to the network.
- The large input space dimension entails the use of complex networks that require a considerable amount of processing power to be trained or used in retrieving the embedded mapping information.
- Large and complex neural networks usually face difficulties to converge to acceptable training error levels.
- Even in cases where a complex network manages to converge, the performance of their recall phases is almost the same as the performance of other architectures.
- Both the second and the third structures are simple networks, and there is no noticeable difference in the performance of their test phases.

Due to all of the above remarks, we opt to employ the second structure, which is by far the simplest and most efficient one, as will be illustrated in the next section. To train

Figure 5. Structure of the third proposed neural network for shot boundary detection



the neural network, we use a modified version of the back-propagation algorithm proposed in Rumelhart, Hinton, & Williams, 1986); this proposed algorithm works as follows:

- Determine the size of the network to be trained. This includes the number of hidden layers and the number of neurons in each layer.
- Determine the type of the activation function. In our case, we use the bipolar sigmoid activation function defined in equation (12).

$$f(x) = \frac{2}{1+\exp(-\lambda x)} - 1 \tag{12}$$

Where

λ: The slope, steepness coefficient, of the activation function.

- Determine the values of the training set and its size.
- Determine the values of the momentum back-propagation algorithm parameters. These are the learning rate (h) and the momentum coefficient (a).
- Randomly initialize all weights and thresholds.

- Present the first element of the training set to the network and initialize an error accumulator (E=0.0).
- Calculate the actual output of each neuron in all layers using equation (13).

$$y_{pj} = f\left(\sum_{i=1}^{n} w_{ji} x_i\right) \tag{13}$$

Where

y_{pj}: Output of node j in any layer in response to presenting pattern p.
w_{ji}: The weight-connecting node i in the previous layer to node j in this layer.
x_i: Output of node i in the previous layer.
f: The bipolar sigmoid activation function defined in equation (12).
n: The number of nodes in the previous layer.

- Use the responses of each layer as inputs to the next layer.
- Accumulate the error due to the actual output of each neuron in the output layer using equation (14).

$$E = E + \frac{1}{2}(d_{pk} - o_{pk})^2 \tag{14}$$

Where

d_{pk}: Desired output of node k at the output layer in response to input pattern p.
o_{pk}: Actual output of node k at the output layer in response to input pattern p.

- Calculate the error signal at any node j for a pattern p using equation (15) for the output layer and equation (16) for hidden layer(s).

$$\delta_{pj} = \frac{1}{2}\lambda(1 - o_{pj}^2)(d_{pj} - o_{pj}) \tag{15}$$

$$\delta_{pj} = \frac{1}{2}\lambda(1 - o_{pj}^2)\sum_{k=1}^{K} \delta_{pk} w_{kj} \tag{16}$$

Where

δ_{px}: The error signal at node x as a result of presenting pattern p.
w_{kj}: The weight connecting node j in this layer to node k in the next layer.
K: The number of the neurons in the next layer.

Table 2. Neural network and back-propagation algorithm parameters used in training and testing

# of input nodes	# of hidden layers	# of nodes in each hidden layer	K	η	α	λ
structure dependent	1	4	1	0.4-0.8	0.2-0.4	1.0

- Adapt each layer weights using equation (17).

$$w_{ji}(t+1) = w_{ji}(t) + \eta \delta_{pj} o_{pj} + \alpha \Delta w_{ji}(t) \qquad \textbf{(17)}$$

Where

$\Delta w_{ji}(t)$: Weight adaptation at time t.

- For all of the remaining elements in the training set, go and repeat execution starting from equation (13).
- If $E < Emax$, then store the connection weights, then exit, otherwise proceed.
- If the number of iterations exceeds a maximum value, stop declaring convergence failure, otherwise initialize the error accumulator ($E=0.0$) and repeat execution starting from equation (13).

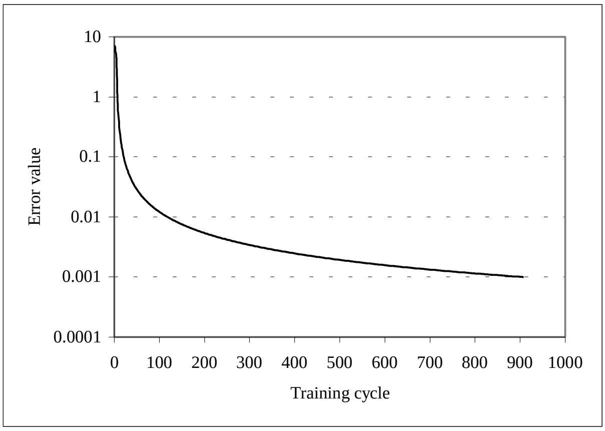
To perform the recall phase of the network, a similar algorithm to the one just described is used, but without any weight adaptation. Instead, the resulting weights produced during the training phase will be used to calculate the output of the network in the feedforward direction via equation (13).

To determine proper values for the parameters of the back-propagation algorithm and the neural network, many combinations of different values have been tested in order to select those that give better results. These parameters include the number of hidden layers, number of neurons in each hidden layer, the learning rate, the momentum coefficient, the slope of the sigmoid function, and the number of nodes at the input and the output layers. Table 2 shows the best values obtained for these parameters out of our experimentations.

EXPERIMENTAL RESULTS

At first, the network is trained using a combination of two video clips. The first clip is a soccer match video that has one cut, while the second one is a wrestling clip that has two cuts. The network learned the classification task quickly and stored the mapping between the inputs and outputs into its connection weights. In spite of the small training set used, the convergence behavior of the network was very good, as shown in Figure. 6, which depicts the learning error throughout the course of the training phase. The Y-axis of Figure 6 has a logarithmic scale to illustrate the rapid decay in learning error as training cycle increases. The next step is to test the generalization of the network during

Figure 6. Error value throughout the neural network training phase



the recall phase. Many video clips (about 60) from various situations have been used in the testing phase. The results were very good in that the network was able to detect almost all cuts in these clips although they had never been presented to the network before. Table 3 shows shot boundary detection results for twelve clips from our database.

At this point, some analysis of the robustness of the obtained results is worthwhile. Consider the first clip in TABLE 33, the soccer video, where the frame difference graph for this clip is shown in Figure. 77. This clip has a dimension of 352x288 with 171 frames. There are two cuts at Indexes 9 and 58, assuming the first frame difference index is 0. It can be observed from the graph that the two cuts are distinguished as two high peaks but, at the same time, there are a number of local peaks with comparable heights to the two cuts, for example, the one at 65. These local high peaks are the results of fast object motion in addition to fast panning of the camera. Many algorithms that use threshold to detect shot boundary will be fooled by these False Alarms (FAs). Moreover, other

Table 3. Shot boundary detection results for various video genres

Video name	# of frames	# of cuts	# of detected cuts	False alarms
soccer	171	2	2	0
racing-boats	263	4	4	1
action-movie	178	2	2	0
carton	331	8	8	0
celebration	457	1- (5)	1-(4)	0
comedy	655	4	4	0
ads	759	10	10	0
class	2793	6	6	0
news-cast	2321	20	20	2
conf-discussion	4783	19	19	0
documentary	5094	41	35	2
tv-show	6139	72	72	0

algorithms that use different rules of thumb, such as the cut peak should be twice as large as any surrounding peak (Yeo & Liu, 1995b), will miss the actual cut due to the presence of these local peaks. The robustness of our algorithm is evident in that particular clip where it detects only the two cuts and discards all false alarms.

Take from Table 3 the celebration clip as another example where Figure 8 illustrates the frame difference for such a clip. This clip has a dimension of 320x240 with 457 frames. There is only one cut at 78, assuming the first frame index is 0. This cut is observable as the first high peak in Figure 8, the frame difference diagram. There are also five high peaks in the diagram at 99, 137, 162, 187, and 268; actually each one of these peaks is two adjacent peaks, for instance, there are almost two similar-value peaks at 99 and 100. Due to the large difference in lighting incurred by the occurrence of a flashlight, the frame difference diagram will have two consecutive large peaks of almost the same value for each occurrence. These peaks are a property in the frame difference diagram that indicates the presence of flashlights at these points; hence, we use this property to detect their occurrences.

The detection performance observed from investigating Table 3 is very good. All cuts have been detected except in the documentary clip that has a lot of lighting variations that causes some misses. FAs are minimal; they happen in the racing-boats clip where a large object occupies the whole scene, and in the newscast and documentary videos where dissolve-like transitions are incorrectly detected as cuts. Four flashlights out of five are detected in the celebration clip because of the weakness of the missed one. Based on the results in Table 3 the overall detection percent is almost 97%, with 2.6% false alarms. These detection rates outperform the reported results of other proposed methods for detecting shot boundaries (Hanjalic & Zhang, 1999; Lee, Kim, & Choi, 2000). Moreover, we use various types of videos, and most of the used clips contain very fast camera work or object motion. Other algorithms fail under these fast motions; on the contrary, ours performs very well in all these situations.

A longer list of the segmentation results we obtained by applying the proposed algorithm to other video clips in our database is given in Table 4. These results support

Figure 7. Frame difference graph for the soccer match

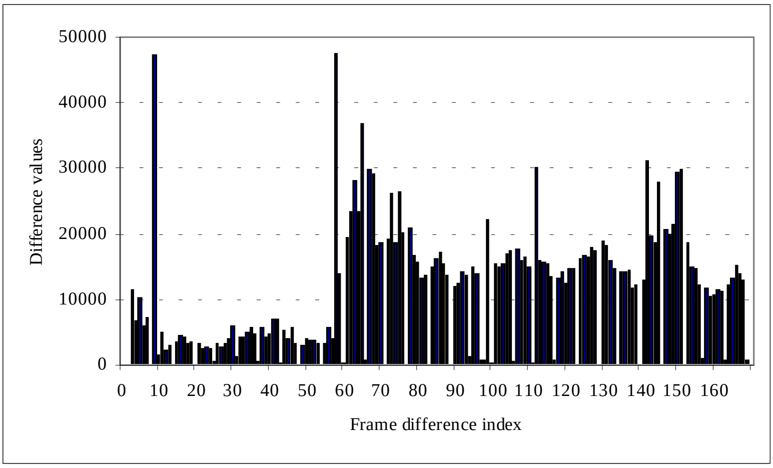
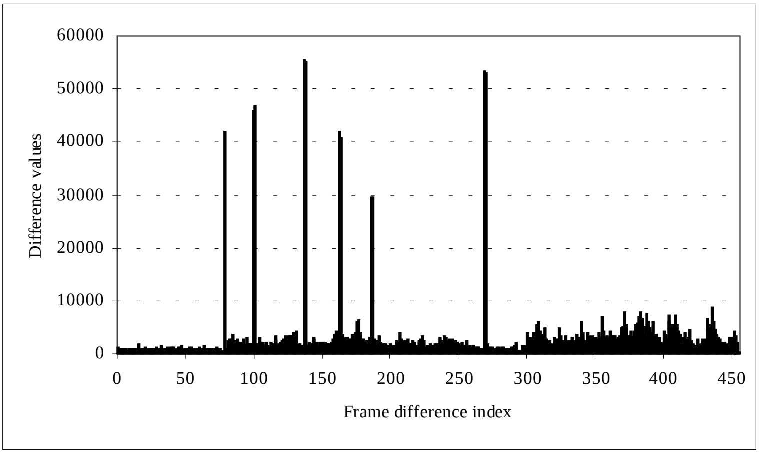


Figure 8. Frame difference graph for the celebration clip



the effectiveness of the proposed technique and its generalization ability even when applied to wider range of video data with different contents and characteristics.

To evaluate the efficiency of our system, we compare its performance with other systems proposed by different researchers. Lee et al. (2000) compared the performance of three techniques to detect shot boundaries, and they called them DC, FB, and PM,

Table 4. Detailed shot boundary detection results for various video clips

Video name	Dimension	# of frames	# of cuts (at positions)	# of detected cuts	False alarms
news-cast	160x120	2321	20	20 (same)	2
tv-show	160x120	6139	72	72 (same)	0
smg-npa-3	160x120	1115	16	16 (same)	0
srose-p2	160x120	3535	22	22 (same)	0
srose-p4	160x120	1678	10	10 (same)	0
class	176x112	2793	6 (442, 804, 1452, 1866, 2306, 2641)	6 (same)	0
adecco	176x112	273	0	0	0
conf-discussion	176x120	4783	19	19 (same)	0
enterprise	176x144	400	0	0	0
crawle	176x144	235	1 (87)	1 (same)	0
dbvath-qcif	176x144	179	1 (121)	1 (same)	0
hoey-v-kill	176x144	310	0	0	0
carton	304x224	331	8	8 (same)	0
v-hi	320x240	1800	2 (316, 987)	2 (same)	0
documentary	320x240	5094	41	35	2
tv-accident	320x240	5841	28	23	2
baby-f	320x240	245	0	0	0
tennis1	320x240	79	0	0	0
celebration	320x240	457	1 (5 flashes)	1 (4 flashes)	0
racing-boats	320x240	263	4 (25, 68, 142, 218)	4 (same)	1
mov0008	320x240	378	0	0	0
speed167	320x240	311	3 (151, 177, 292)	3 (same)	0
sprint	320x240	479	1 (210)	1 (same)	0

Table 4. Detailed shot boundary detection results for various video clips (cont.)

Video name	Dimension	# of frames	# of cuts (at positions)	# of detected cuts	False alarms
ah6a27	320x240	253	1 (68)	1 (same)	0
necktw	320x240	183	2 (46, 160)	2 (same)	0
sf-anna	320x240	1300	2 (314, 525)	2 (same)	0
corks	352x240	768	9 (68, 152, 179, 215, 462, 483, 516, 648, 675)	9 (same)	0
ads	352x240	759	10 (65, 97, 258, 407, 437, 579, 614, 650, 674)	10 (same)	0
action-movie	352x240	178	2 (51, 104)	2 (same)	0
close	352x240	751	0	0	0
hyakutake	352x240	526	0	0	0
fish	352x240	343	1 (296)	1 (same)	0
comedy	352x240	655	4 (48, 364, 500, 614)	4 (same)	0
lions	352x240	298	3 (205, 252, 278)	3 (same)	0
sq	352x240	255	0	0	0
039	352x240	1036	11	11 (same)	4 (large objects)
Adver-sound	352x260	123	3 (20, 43, 77)	3 (same)	0
04-eg4.mpg	352x288	171	2 (9, 58)	2 (same)	0
crawley	352x288	283	1 (95)	1 (same)	0
ah6a4b	352x288	181	1 (34)	1 (same)	0
fai2-3	352x288	253	1 (193)	1 (same)	0
dbvath_cif	352x288	175	1 (129)	1 (same)	0
hoey_v_kill_cif	352x288	377	1 (327)	1 (same)	0

respectively. The average frames/second achieved by each of these techniques (averaged over four video clips) is listed in Table 5. To achieve an accurate comparison with the data listed in Table 5, we tried our best to perform all our measurements under the same conditions, such as operating system, type of the processor, and processor speed. After that, we used our technique to measure the number of frames that can be processed in one second for a number of clips from our database and list the results in Table 6. Figure 9 shows a comparison between the average efficiency of the three methods evaluated in Lee et al. (2000) and the performance of our proposed technique where the efficiency of our system is obvious.

Analysis of the results in Table 6 indicates that our system managed to achieve an average of about 89.3 frames/sec (averaged over 12 video clips containing about 24,000 frames). This achieved speed is more than eight times faster than the fastest method to detect shot boundary reported in TABLE 66, and the performance gab is also evident in Figure 9. Although our algorithm was implemented in Java Language (Sun Microsystems, 2003), which is inherently much slower than other native programming languages, it achieves a dramatic speed up over the fastest method reported in the literature, a sound evident of the efficiency of the proposed technique. An important factor that contributes the major share to this efficiency is our novel use of the instantaneous recall phase of the neural network to accomplish the shot boundary detection task.

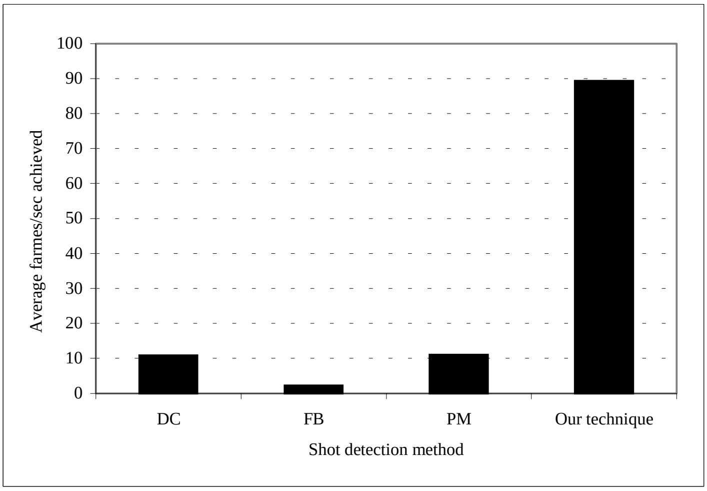
Table 5. Average speed comparison of three scene change detection methods from Lee et al. (2000)

Method name	Average frames/sec	# of clips used in the average
DC	10.9	4
FB	2.3	4
PM	11.1	4

Table 6. Detection time and frames per second for some clips from our database

Video name	# of frames	Detection time (sec)	# of frames/sec
soccer	171	7.6	22.4
racing-boats	263	6.7	39.3
action-movie	178	5.5	32.5
carton	331	5.1	64.8
celebration	457	15.3	29.9
comedy	655	18.7	35.1
ads	759	20.9	36.3
class	2793	15.5	180.3
news-cast	2321	10.6	219.3
conf-discussion	4783	23.6	202.6
documentary	5094	35.7	172.0
tv-show	6139	137.4	37.1

Figure 9. Speed comparison of different shot detection techniques



CONCLUSIONS

This work introduces an efficient and robust system for detecting video scene changes, an essential task in fully content analysis systems. The first module extracts the DC sequence directly from compressed data without the need for decoding. Subsequently, the second module receives DC frame differences as inputs, then recalls the information stored into the neural network weights to determine the outputs. The algorithm has been tested on wide varieties of video genres and proved its robustness in almost all cases, where detection percent of 97% was achieved with 2.6% FAs. Better generalization of the neural network can be achieved by increasing the number of video clips used in the training phase and by varying their contents. Networks that generalize better provide superior performance and widen the applicability of our technique to cover every possible video genre. Moreover, the efficiency of the developed technique has been tested by measuring the number of frames per second that the system can process. The test yields an average value of about 89.3 frames/second. This average value represents a dramatic increase in speed in comparison to the speed of other systems that perform the same shot boundary detection task. In a nutshell, the effectiveness of the proposed paradigm has been proven as a robust and very efficient way to identify scene changes in compressed MPEG video streams. The proposed video detection paradigm introduced in this chapter is the first stage in a Video Content-based Retrieval (VCR) system that has been designed at Old Dominion University.

REFERENCES

Beale, R., & Jackson, T. (1991). *Neural computing: An introduction*. New York: IOP Publishing.

Chen, J., Taskiran, C., Albiol, A., Delp, E., & Bouman, C. (1999). ViBE: A video indexing and browsing environment. *Proceedings of SPIE/IS&T Conf. Multimedia Storage and Archiving Systems IV, 3846*, 148-164.

Farag, W., & Abdel-Wahab, H. (2001). A new paradigm for detecting scene changes on MPEG compressed videos. *Proceedings of IEEE International Symposium on Signal Processing and Information Technology*, 153-158.

Farag, W., & Abdel-Wahab, H. (2002). A new paradigm for analysis of MPEG compressed videos. *Journal of Network and Computer Applications*, *25*(2), 109-127.

Farag, W., Ziedan, I., Syiam, M., & Mahmoud, M. (1997). Architecture of neural networks using genetic algorithms. *Proceedings of 5th International Conference on Artificial Intelligence Applications (5th ICAIA)*.

Hampapur, A., Jain, R., & Weymouth, T. (1994). Digital video segmentation. *Proceedings of ACM International Conference on Multimedia*, 357-364.

Hanjalic, A., & Zhang, H.-J. (1999). Optimal shot boundary detection based on robust statistical models. *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, 710-714.

Idris, F., & Panchanathan, S. (1997). Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, *8*(2), 146-166.

ISO/IEC 11172-2:1993/Cor 2:(1999): Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s, Part 2: Video.

Lee, S.-W., Kim, Y.-M., & Choi, S. (2000). Fast scene change detection using direct feature extraction from MPEG compressed videos. *IEEE Trans. Multimedia*, *2*(4), 240-254.

LeGall, D. (1991). MPEG: A video compression standard for multimedia applications. *Comm. of ACM*, *34*(4), 46-58.

Low, C., Tian, Q., & Zhang, H.-J. (1996). An automatic news video parsing, indexing, and browsing system. *Proceedings of ACM International Conference on Multimedia*, 425-426.

Meng, J., Juan, Y., & Chang, S.-F. (1995). Scene change detection in an MPEG compressed video sequence. *Proceedings of IS&T/SPIE Symposium on Digital Video Compression: Algorithms and Technologies*, *2419*, 14-25.

Mitchell, J., Pennebaker, W., Fogg, C., & LeGall, D. (1997). *MPEG video: Compression standard*. London: Chapman and Hall.

Pennebaker, W., & Mitchell, J. (1993). *JPEG still image data compression standard*. New York: Van Nostrand Reinhold.

Rui, Y., Huang, T., & Mehrotra, S. (1998). Browsing and retrieving video content in a unified framework. *Proceedings of IEEE Workshop on Multimedia Signal Processing*, 9-14.

Rumelhart, D.E., Hinton, G., & Williams, R. (1986). Learning internal representation by error propagation. In D.E. Rumelhart, G.E. Hinton, GE, & J.L. McClelland (Eds.), *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Shen, K., & Delp, E. (1995). A fast algorithm for video parsing using MPEG compressed sequences. *Proceedings of IEEE International Conference on Image Processing*, *2*, 252-255.

Sun Microsystems (2003). Java language. Retrieved on the World Wide Web at: *http://java.sun.com/products*

Swain, M., & Ballard, D. (1991). Color indexing. *Computer Vision*, *7*(1), 11-32.

Tonomura, Y. (1991). Video handling based on structured information for hypermedia systems. *Proceedings of ACM International Conference on Multimedia Information Systems*, 333-344.

Vasconcelos, N., & Lippman, A. (1997). Towards semantically meaningful spaces for the characterization of video content. *Proceedings of IEEE International Conference on Image Processing*, *2*, 542-545.

Wallace, G. (1991). The JPEG still picture compression standard. *Communications of ACM*, *34*(4), 30-44.

Yeo, B.-L., & Liu, B. (1995a). On the extraction of DC sequence from MPEG compressed video. *Proceedings of IEEE International Conference on Image Processing*, *2*, 260-263.

Yeo, B.-L., & Liu, B. (1995b). Rapid scene analysis on compressed video. *IEEE Trans Circuits and Systems for Video Technology*, *5*(6), 533-544.

Zhang, H.-J., Kankanhalli, A., Smoliar, S., & Tan, S. (1993). Automatically partitioning of full-motion video. *Multimedia Systems*, *1*(1), 10-28.

Zhang, H.-J., Tan, S., Smoliar, S., & Gong, Y. (1995). Automatic parsing and indexing of news video. *Multimedia Systems*, *2*, 256-266.

Zurada, J. (1992). *Introduction to artificial neural systems*. St. Paul, MN: West Publishing Company.

Chapter IX

Innovative Shot Boundary Detection for Video Indexing

Shu-Ching Chen, Florida International University, USA

Mei-Ling Shyu, University of Miami, USA

Chengcui Zhang, University of Alabama at Birmingham, USA

ABSTRACT

Recently, multimedia information, especially video data, has been made overwhelmingly accessible with the rapid advances in communication and multimedia computing technologies. Video is popular in many applications, which makes the efficient management and retrieval of the growing amount of video information very important. Toward such a demand, an effective video shot boundary detection method is necessary, which is a fundamental operation required in many multimedia applications. In this chapter, an innovative shot boundary detection method using an unsupervised segmentation algorithm and the technique of object tracking based on the segmentation mask maps *is presented. A series of experiments on various types of video types are performed, and the experimental results show that our method can obtain object-level information of the video frames as well as accurate shot boundary detection, which are very useful for video content indexing.*

INTRODUCTION

Unlike traditional database systems that have text or numerical data, a multimedia database or information system may contain different media such as text, image, audio, and video. Video, in particular, has become more and more popular in many applications such as education and training, video conferencing, video-on-demand (VOD), and news services. The traditional way for the users to search for certain content in a video is to fast-forward or rewind, which are sequential processes, making it difficult for the users to browse a video sequence directly based on their interests. Hence, it becomes important to be able to organize video data and provide the visual content in compact forms in multimedia applications (Zabih, Miller, & Mai, 1995).

In many multimedia applications such as digital libraries and VOD, video shot boundary detection is fundamental and must be performed prior to all other processes (Shahraray, 1995; Zhang & Smoliar, 1994). A video shot is a video sequence that consists of continuous video frames for one action, and shot boundary detection is an operation to divide the video data into physical video shots. Many video shot boundary detection methods have been proposed in the literature. Most of them use low-level global features in the matching process between two consecutive frames for shot boundary detection, for example, using the luminance pixel-wise difference (Zhang, Kankanhalli, & Smoliar, 1993), luminance or color histogram difference (Swanberg, Shu, & Jain, 1993), edge difference (Zabih et al., 1995), and the orientation histogram (Ngo, Pong, & Chin, 2000). However, these low-level features cannot provide satisfactory results for shot boundary detection since luminance or color is sensitive to small changes. For example, Yeo and Liu (1995) proposed a method that uses the luminance histogram difference of DC images, which is very sensitive to luminance changes. There are also approaches focusing on the compressed video data domain. For example, Lee, Kim, and Choi (2000) proposed a fast scene/shot change detection method, and Hwang and Jeong (1998) proposed the directional information retrieving method by using the discrete cosine transform (DCT) coefficients in MPEG video data.

In addition, dynamic and adaptive threshold determination is also applied to enhance the accuracy and robustness of the existing techniques in shot cuts detection (Alattar, 1997; Gunsel, Ferman, & Tekalp, 1998; Truong, Dorai, & Venkatesh, 2000). In Gunsel et al. (1998), the unsupervised clustering algorithm proposed a generic technique that does not need threshold setting and allows multiple features to be used simultaneously; while an adaptive threshold determination method that reduces the artifacts created by noise and motion in shot change detection was proposed by Truong et al. (2000).

In this chapter, we present an innovative shot boundary detection method using an unsupervised image-segmentation algorithm and the object-tracking technique on the uncompressed video data. In our method, the image-segmentation algorithm extracts the segmentation mask map of each video frame automatically, which can be deemed as the clustering feature map of each frame and where the pixels in each frame have been grouped into different classes (e.g., two classes). Then the difference between the segmentation mask maps of two frames is checked. Moreover, due to camera panning and tilting, we propose an object-tracking method based on the segmentation results to enhance the matching. The cost for object tracking is almost trivial since the segmentation results are already available. In addition, the bounding boxes and the positions of the segments within each frame obtained from the segmentation are used for our key frame representation. In order to reduce the computational cost, we also apply the traditional pixel-level comparison for pre-processing, in addition to segmentation and object tracking. The basic idea in pixel-level comparison is to compute the differences in values of corresponding pixels between two successive frames. One threshold is used to determine whether the value of the corresponding pixels has changed, while another threshold is used to measure the percentage of changed pixels between two successive frames. If the percentage of changes exceeds some pre-defined threshold, then a new shot cut is detected. This method is very simple, but the disadvantage is that it is very sensitive to object and camera movements. To overcome its shortcomings, pixel-level comparison is embedded into the techniques of object tracking and image segmentation in our method. The advantages of our shot boundary detection method are:

1. It is fully unsupervised, without any user interventions.
2. The algorithm for comparing two frames is simple and fast.
3. The object-level segmentation results can be further used for video indexing and content analysis.

We begin with a literature review and the motivations of the proposed framework. Then the unsupervised image-segmentation and object-tracking techniques are introduced. After that, our experimental results are presented, and the future trends are discussed. Finally, a brief conclusion is given.

BACKGROUND

In this section, the existing approaches for video shot detection and their relative advantages and disadvantages are discussed. Video segmentation is the first step for automatic indexing of digital video for browsing and retrieval. The goal is to separate the video into a set of shots that can be used as the basis for video indexing and browsing. Most of the algorithms process uncompressed video data, while some of them operate directly on the compressed video data.

A survey on video indexing and video segmentation in uncompressed data domain was presented by Gargi, Kasturi, and Antani (1998). The shot boundary detection algorithms in the uncompressed domain process uncompressed video, and a similarity measure between successive frames is defined (Nagasaka & Tanaka, 1995; Zhang et al., 1993). Lots of the approaches use pixel-level comparison to compute the differences in values of corresponding pixels between two successive frames; however, it is very sensitive to object and camera movements. In our method, pixel-level comparison is embedded into the techniques of object tracking and image segmentation in order to overcome its shortcomings and to reduce the computation cost. Other kinds of comparison techniques used in the uncompressed domain are block-wise comparison and histogram-based comparison. Block-wise comparison reduces the sensitivity to object and camera movements by utilizing the local characteristics (e.g., mean and variance intensity values) of the blocks. In this kind of approach, each frame is divided into several blocks that are compared with their corresponding blocks in the successive frame. If the number of changed blocks exceeds some threshold, then a shot cut is detected. This method is more robust, but it is still sensitive to fast object movement or camera panning. Moreover, it is also highly possible to introduce an incorrect matching between two blocks that have the same mean and variance values but with totally different contents, due to the fact that the mean and variance values of a block are not good enough to represent the block's characteristics (Xiong & Lee, 1998). In our method, the idea of block matching is partially adopted in the object-tracking technique. Instead of dividing a frame into fixed size of blocks absolutely, an innovative image-segmentation method is employed to cluster the pixels in a frame into multiple classes (normally two classes) and obtain the segments (blocks). These segments (blocks) are then tracked and matched between two successive frames. On the other hand, histogram-based comparison is based on the premise that since the object moving between two successive frames is relatively small, there will not be a big difference between their histograms. It is more robust to small rotations and slow variations (Pass & Zahib, 1999; Swain, 1993). However, two successive frames may have similar histograms but with different contents.

In the compressed domain, there are also many shot boundary detection algorithms, especially in the MPEG format. It is suitable for video shot boundary detection because the encoded video stream already contains many features, including the DCT coefficients, motion vectors, etc. Arman, Hsu, and Chiu (1993) use the DCT coefficients of I frames as the similarity measure between successive frames; while the dc-images are used to compare successive frames, where the (*i*,*j*) pixel value of the DC-image is the average value of the (*i*,*j*) block of the image (Yeo & Liu, 1995). Hwang and Jeong (1998) utilized the changes of directional information in the DCT domain to detect the shot breaks automatically. The DCT coefficient-based method was further improved by Lee et al. (2000), who used the binary edge maps as a representation of the key frames so that two frames could then be compared by calculating a correlation between their edge maps. Its advantage is that it gives directly the edge information such as orientation, strength, and offset from the DCT coefficients, and its disadvantage is similar to all the compressed domain-based methods, that is, sensitivity to different video compressing formats.

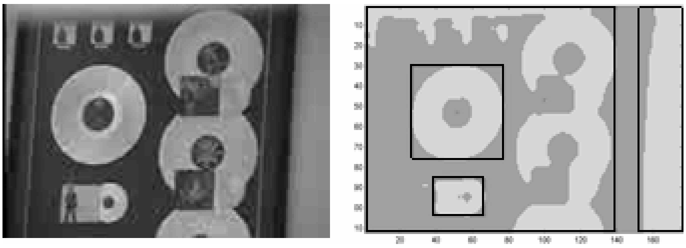
PROPOSED SHOT BOUNDARY DETECTION METHOD

In this section, we first explain how the unsupervised segmentation algorithm and object tracking work, and then provide the steps of the shot change detection method based on the discussion.

Segmentation Information Extraction

In this chapter, we use an unsupervised segmentation algorithm to partition the video frames. First, the concepts of a class and a segment should be clarified. A class is characterized by a statistical description and consists of all the regions in a video frame that follow this description; a segment is an instance of a class. In this algorithm, the partition and the class parameters are treated as random variables. This is illustrated in Figure 1. The light gray areas and dark gray areas in the right segmentation-mask map represent two different classes respectively. Considering the light gray class, there are a total of four segments within this class (the CDs, for example). Notice that each segment

Figure 1. Examples of classes and segments (The original video frame is on the left, and the segmentation mask map of the left frame is on the right.)



is bounded by a bounding box and has a centroid, which are the results of segment extraction. The details of segment extraction will be discussed in a later section.

The method for partitioning a video frame starts with a random partition and employs an iterative algorithm to estimate the partition and the class parameters jointly (Chen, Sista, Shyu, & Kashyap, 1999, 2000; Chen, Shyu, & Kashyap, 2000). The intuition for using an iterative way is that a given class description determines a partition and, similarly, a given partition gives rise to a class description. So the partition and the class parameters have to be estimated iteratively and simultaneously from the data.

Suppose there are two classes — *class*1 and *class*2. Let the partition variable be $\boldsymbol{c} = \{\boldsymbol{c_1}, \boldsymbol{c_2}\}$, and the classes be parameterized by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$. Also, suppose all the pixel values y_{ij} (in the image data Y) belonging to class k $(k=1,2)$ are put into a vector $\boldsymbol{Y_k}$. Each row of the matrix Φ is given by $(1, i, j, ij)$ and a_k is the vector of parameters $(a_{k0}, \ldots, a_{k3})^T$.

$$y_{ij} = a_{k0} + a_{k1}i + a_{k2}j + a_{k3}ij,\ \forall (i,j)\, y_{ij} \in \boldsymbol{c_k} \quad \textbf{(1)}$$

$$\boldsymbol{Y_k} = \Phi a_k \quad \textbf{(2)}$$

$$\hat{a}_k = (\Phi^T \Phi)^{-1} \Phi^T \boldsymbol{Y_k} \quad \textbf{(3)}$$

The best partition is estimated as that which maximizes the a posteriori probability (MAP) of the partition variable given the image data Y. Now, the MAP estimates of $\boldsymbol{c} = \{\boldsymbol{c_1}, \boldsymbol{c_2}\}$ and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ are given by

$$\begin{aligned}(\hat{c},\hat{\theta}) &= Arg \max_{(c,\theta)} P(c,\theta \mid Y) \\ &= Arg \max_{(c,\theta)} P(Y \mid c,\theta) P(c,\theta) \quad \textbf{(4)}\end{aligned}$$

Let $J(\mathrm{c},\theta)$ be the functional to be minimized. With the above assumptions, this joint estimation can be simplified to the following form:

$$(\hat{c},\hat{\theta}) = Arg \min_{(c,\theta)} J(c_1, c_2, \theta_1, \theta_2) \quad \textbf{(5)}$$

$$J(c_1, c_2, \theta_1, \theta_2) = \sum_{y_{ij} \in c_1} -\ln p_1(y_{ij}; \theta_1) + \sum_{y_{ij} \in c_2} -\ln p_2(y_{ij}; \theta_2) \tag{6}$$

Thus, the problem of segmentation becomes the problem of simultaneously estimating the class partition and the parameter for each class. With regard to the parameter estimation, we can use equation (3) to directly compute the parameter for each assigned set of class labels without any numerical optimization methods. For the class partition estimation, we assign pixel y_{ij} to the class that gives the lowest value of $-\ln p_k(y_{ij} \mid \theta_k)$. The decision rule is:

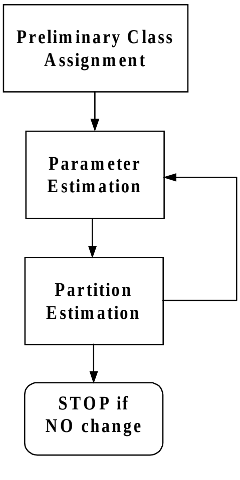
$$y_{ij} \in \hat{c}_1 \text{ if } -\ln p_1(y_{ij}) \le -\ln p_2(y_{ij}) \tag{7}$$

$$y_{ij} \in \hat{c}_2 \text{ otherwise} \tag{8}$$

Just as shown in Figure 2, the algorithm starts with an arbitrary partition of the data in the first video frame and computes the corresponding class parameters. Using these class parameters and the data, a new partition is estimated. Both the partition and the class parameters are iteratively refined until there is no further change in them. We note here that the functional *J* is not convex. Hence, its minimization may yield a local minimum, which guarantees the convergence of this iterative algorithm. Since the successive frames do not differ much due to the high temporal sampling rate, the partitions of adjacent frames do not differ significantly. The key idea is then to use the method successively on each frame of the video, incorporating the partition of the previous frame as the initial condition while partitioning the current frame, which can greatly reduce the computing cost.

We should point out that the SPCPE algorithm could not only simultaneously estimate the partition and class parameters, but also estimate the appropriate number of

Figure 2. Flowchart of SPCPE algorithm



the classes in the mean time by some easy extension of the algorithm. Moreover, it can handle multiple classes rather than two. In our experiment, we just use two classes in segmentation since two classes are efficient and good enough for our purpose in this application domain.

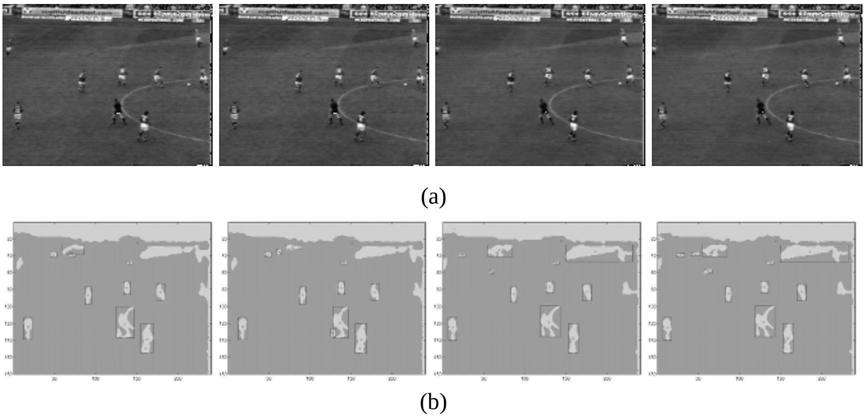
Object Tracking

The first step for object tracking is to identify the segments in each class in each frame. Then the bounding box and the centroid point for that segment are obtained. For example, Figure 3(b) shows the segmentation mask maps of the video sequence in Figure 3(a). In this figure, the player, soccer ball, and the signboard belong to Class 2 while the ground belongs to Class 1. As shown in Figure 3(b), the segments corresponding to the ball, player, and signboard are bounded by their minimal bounding boxes and represented by their centroid points.

Line Merging Algorithm (LMA) for Extracting Segments

Unlike the traditional way to do segment extraction such as the *seeding and region growing* method used by Sista and Kashyap (1999), we use a computationally simple and fast method called a *line merging algorithm (LMA)* to extract the segments from the segmented frames. The basic idea is to scan the segmented frame either row-wise or column-wise. If the number of rows (columns) is less than the number of columns (rows), then row-wise (column-wise) is used respectively. For example, as shown in Figure 4, suppose the pixels with value '*1*' represent the segment we want to extract; we scan the segmented frame row by row. By scanning the first row, we get two lines and let each line represent a new segment so that we have two segments at the beginning. In scanning Rows 2 to 4, we merge the new lines in each row with the lines in previous rows to form the group of lines for each segment. At Row 5, we get one line and find out that it can be merged with both of the two segments, which means we must merge the two previously obtained segments to form a new segment, so that now we have only one big segment.

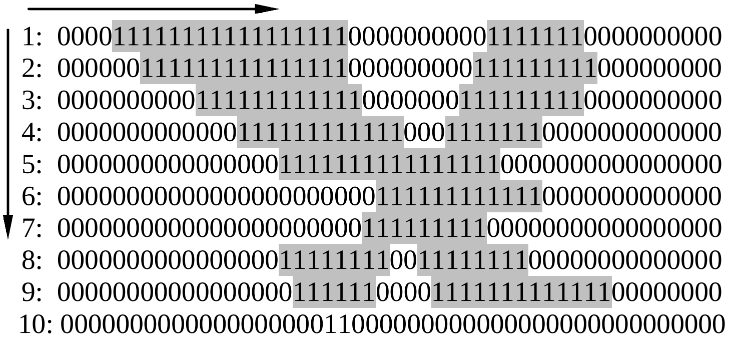
Figure 3. Object tracking: (a) Example of video sequence; (b) Segmentation mask maps and bounding boxes for (a)



(a)

(b)

Figure 4. Segmentation mask map



```
 1: 000011111111111111111000000000011111110000000000
 2: 000000111111111111111000000000111111111000000000
 3: 000000000011111111111100000001111111110000000000
 4: 000000000000011111111111100011111110000000000000
 5: 000000000000000011111111111111110000000000000000
 6: 000000000000000000000001111111111110000000000000
 7: 000000000000000000000011111111100000000000000000
 8: 000000000000000011111111001111111100000000000000
 9: 000000000000000001111110000111111111111100000000
10: 000000000000000000011000000000000000000000000000
```

Similarly, at Row 8, two lines belong to the same segment because they can be merged with the same line in Row 7.

The pseudo codes for the *line merging algorithm (LMA)* are listed in Algorithm 1 and Algorithm 2.

Algorithm 1.

Algorithm: GetSegments(V, i, A[i])→ to get the new lines of each row.

V: the input vector of segmented frame of row 'i';

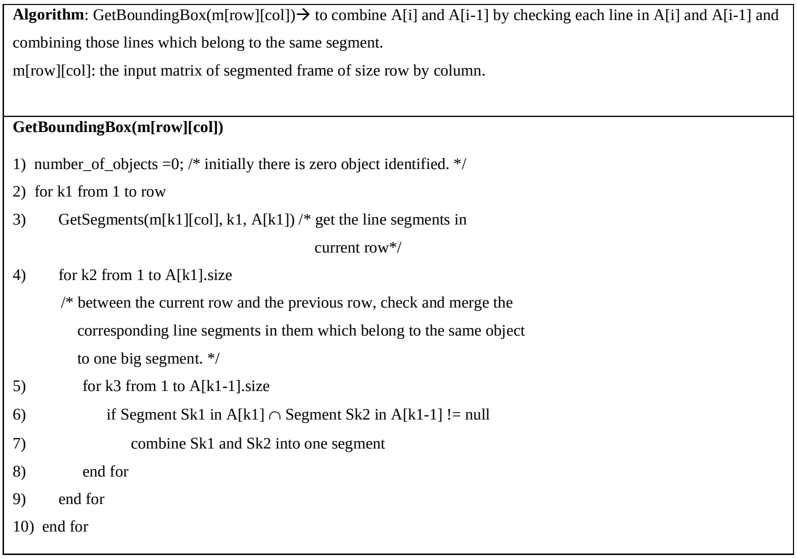
'i': the current row we are scanning;

A[i]: a list to store the segments.

GetSegments(V, i, A[i])

```
Number_of_segments = -1;
Segment D[col/2]; /* D is the temporary variable to store the line segments
                     in row i. The maximal size of D is col/2. */
for j from 1 to col
  if V[j] == 1
    if j == 1 /* if the first line segment is at the beginning of the current row,
                 add it to array D and increase the number of line segments. */
      number_of_segments++;
      D[number_of_segments].data = data; /* data contains the i and j values */
    else if V[j-1] == 0 /* detect a new line segment and add it to array D */
      number_of_segments++;
      D[number_of_segments].data = data;
    else D[number_of_segments].data += data;
    /* collect all the pixels belonging to the same line segment together. */
    end if;
end if;
for k from 0 to number_of_segments /* copy the line segments in D to the
                                      data structure in A[i]. */
    A[i].Add(D[k]);
end for;
```

Algorithm 2.



Algorithm: GetBoundingBox(m[row][col])→ to combine A[i] and A[i-1] by checking each line in A[i] and A[i-1] and combining those lines which belong to the same segment.
m[row][col]: the input matrix of segmented frame of size row by column.

GetBoundingBox(m[row][col])

```
number_of_objects =0; /* initially there is zero object identified. */
for k1 from 1 to row
    GetSegments(m[k1][col], k1, A[k1]) /* get the line segments in
                                            current row*/
    for k2 from 1 to A[k1].size
     /* between the current row and the previous row, check and merge the
        corresponding line segments in them which belong to the same object
        to one big segment. */
        for k3 from 1 to A[k1-1].size
            if Segment Sk1 in A[k1] ∩ Segment Sk2 in A[k1-1] != null
                combine Sk1 and Sk2 into one segment
        end for
    end for
end for
```

Compared with the *seeding and region growing* method, the proposed algorithm extracts all the segments and their bounding boxes as well as their centroids within one scanning process, while the *seeding and region growing* method needs to scan the input data for an indeterminate number of times depending on the layout of the segments in the frame. Moreover, the proposed algorithm needs much less space than the *seeding and region growing* method.

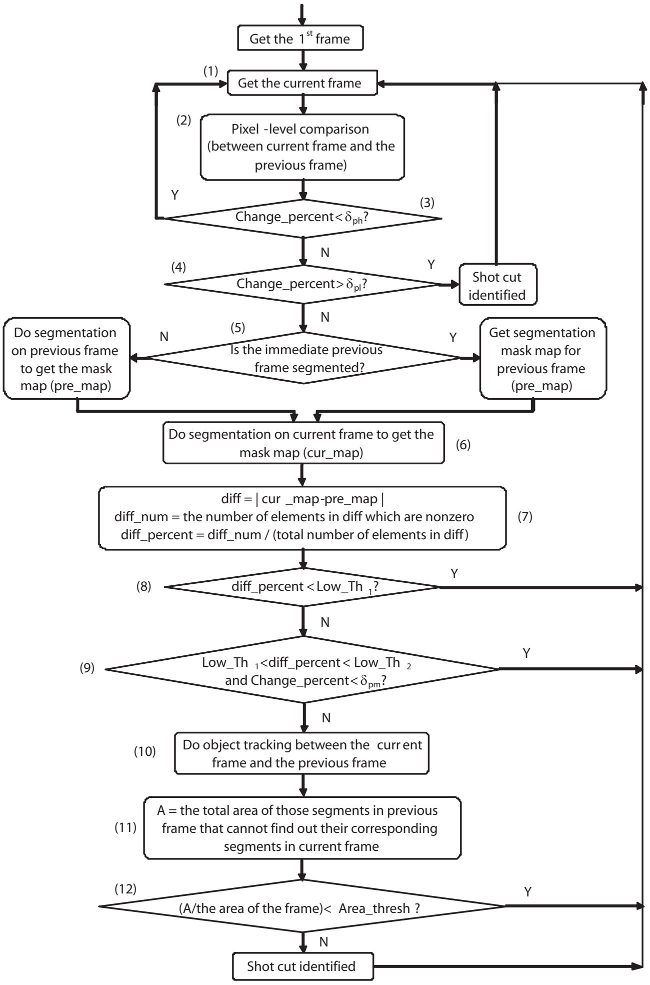
The next step for object tracking is to connect the related segments in successive frames. The idea is to connect two segments that are spatially the closest in the adjacent frames (Sista & Kashyap, 1999). In other words, the Euclidean distances between the centroids of the segments in adjacent frames are used as the criteria to track the related segments. In addition, size restriction should be employed in determining the related segments in successive frames.

In fact, the proposed object-tracking method can be called a "block-motion tracking" method since it is an extension of the macroblock-matching technique used in motion estimation (Furht, Smoliar, & Zhang, 1995; Gall, 1991) between successive frames. The proposed object- tracking method is based on the segmentation results and goes much further than the macroblock- matching technique because it can choose the appropriate macroblocks (segments) within a specific frame by segmentation and track their motions instead of fixed-size and pre-determinate macroblocks.

Shot Boundary Detection Method

Our method combines three main techniques together: segmentation, object tracking, and the traditional pixel-level comparison method. In the traditional pixel-level comparison approach, the gray-scale values of the pixels at the corresponding locations in two successive frames are subtracted, and the absolute value is used as a measure of dissimilarity between the pixel values. If this value exceeds a certain threshold, then the

Figure 5. Flowchart of the proposed shot boundary detection method


Get the 1st frame
(1)
Get the current frame
(2)
Pixel -level comparison (between current frame and the previous frame)
Y
(3)
Change_percent < δph?
N
(4)
Y
Change_percent > δpl?
Shot cut identified
N
Do segmentation on previous frame to get the mask map (pre_map)
N
(5)
Is the immediate previous frame segmented?
Y
Get segmentation mask map for previous frame (pre_map)
Do segmentation on current frame to get the mask map (cur_map)
(6)
diff = | cur _map-pre_map |
diff_num = the number of elements in diff which are nonzero
diff_percent = diff_num / (total number of elements in diff)
(7)
(8)
diff_percent < Low_Th 1?
Y
N
(9)
Low_Th 1<diff_percent < Low_Th 2 and Change_percent < δpm?
Y
N
(10)
Do object tracking between the curr ent frame and the previous frame
(11)
A = the total area of those segments in previous frame that cannot find out their corresponding segments in current frame
(12)
(A/the area of the frame)< Area_thresh ?
Y
N
Shot cut identified

pixel gray scale is said to have changed. The percentage of the pixels that have changed is the measure of dissimilarity between the frames. This approach is computationally simple but sensitive to digitalization noise, illumination changes, and the object moving. On the other hand, the proposed segmentation and object- tracking techniques are much less sensitive to the above factors. In our method, we use the pixel-level comparison for pre-processing. By applying a strict threshold for the percentage of changed pixels, we want to make sure that we will not introduce any incorrect shot cuts that are identified by pixel-level comparison by fault. The advantage to combining the pixel-level comparison is that it can alleviate the cost of computation because of its simplicity. In other word, we apply the segmentation and object-tracking techniques only when it is necessary.

Figure 5 shows the flowchart of the proposed shot boundary detection. The steps are given in the following:

1. Do the pixel-level comparison between the currently processed video frame and the immediate preceding frame (see chart boxes 1 and 2 in Figure 5). Let the percentage of change be *change_percent* and check this variable (chart box 3). If the current frame is not identified as a shot cut, which means that *change_percent*<d_{ph}, then go on to process the next video frame (chart box 1). Otherwise go to step 2 (chart box 4).
2. If *change_percent*>d_{pl} (chart box 4), the current frame is identified as a shot cut. Go to step 1 and process the next frame (chart box 1). Otherwise go to step 3 (chart box 5).
3. Do the segmentation on the previous frame only if the previous frame has never been segmented (chart box 5). If the previous frame has been segmented before, we only need to obtain its segmentation mask map directly. Then do segmentation on the current frame (chart box 6). Get the current and the previous segmentation mask maps for these two frames. Let the variable *cur_map* represent the current segmentation mask map's value and variable *pre_map* represent the value of the previous segmentation mask map. Note that the variables *cur_map* and *pre_map* can be deemed as two matrices. Go to step 4 (chart box 7).
4. *diff* = |*cur_map*-*pre_map*|; where the variable *diff* is the point-to-point subtraction between two successive segmentation mask maps.
   diff_num = the number of elements in *diff* which are nonzero;
   diff_percent = *diff_num* / (total number of elements in *diff*); where the variable *diff_percent* is the percentage of changes between the two successive segmentation mask maps.
   Go to step 5 (chart box 8).
5. Check the variable *diff_percent* (chart box 8).

```
If diff_percent < $Low\_Th_1$
    Not shot boundary. Go to step 1 and process the next frame (chart box 1).
Else
    If $Low\_Th_1$ < diff_percent < $Low\_Th_2$ and change_percent<$d_{pm}$  // chart box 9
        Not shot boundary. Go to step 1 and process the next frame (chart box 1).
    Else
        Do object tracking between the current frame and the previous frame. Let variable A be the total area of those segments in the previous frame that cannot find out their corresponding segments in the current frame;  // chart boxes 10, 11
        If (A/the area of the frame)<Area_thresh  // chart box 12
            Not shot boundary. Go to step 1 and process the next frame (chart box 1).
        Else
            The current frame is identified as shot cut.
            Go to step 1 and process the next frame (chart box 1).
        End if;
    End if;
End if;
```

(Here, d_{ph}, d_{pl}, d_{pm}, Low_Th_1 and Low_Th_2 are threshold values for variables *change_percent* and *diff_percent* that are derived from the experiential values.)

Table 1. Video data used for experiments

Name	Type	Number of Frames	Shot Cuts
News1	News	731	19
News2	News	1262	26
News3	News	4225	90
Labwork	Documentary	495	15
Robert	MTV	885	26
Carglass	Commercial	1294	29
Aussie2g2	Sports	511	19
Flo1	Sports	385	8
Flo2	Sports	406	10
AligoISA	Sports	418	11

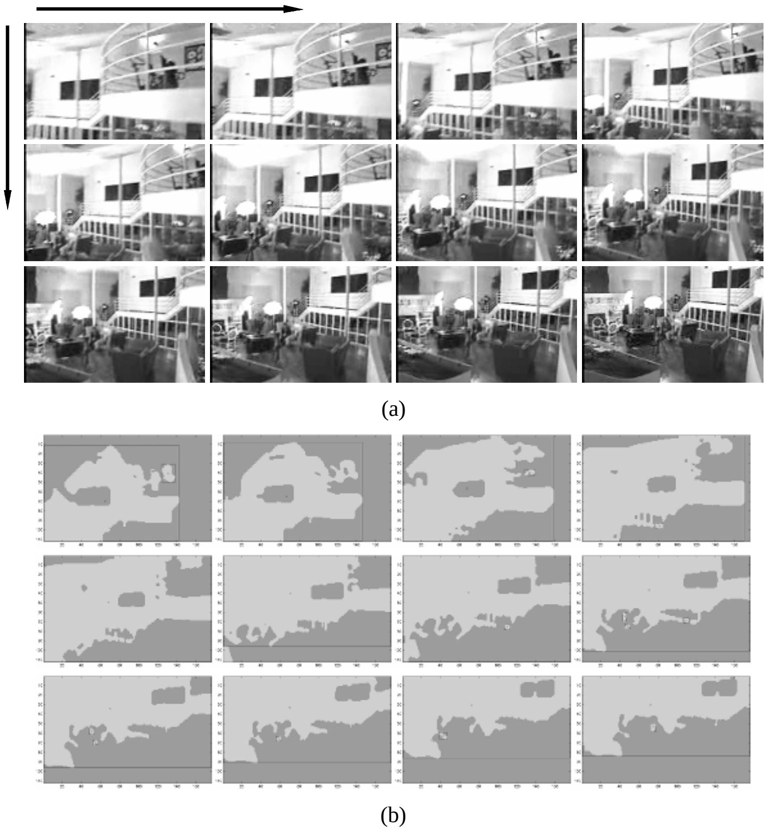
EXPERIMENTAL RESULTS

We have performed a series of experiments on various video types such as the TV news videos (in MPEG-1 format) that include FOX 25 LIVE NEWS, ABC 7 NEWS and WNBC NEWS. Other video types used in our experiment include the music MTV video, documentary video, and sports video such as the soccer game. The average size of each frame in the sample video clips is 180 rows and 240 columns. All the MPEG video clips are downloaded from the URLs listed in the Appendix. Table 1 gives the statistics of all the video clips used. The experimental results demonstrate the effectiveness of the proposed shot boundary detection algorithm. Next, we will see how the proposed method detects the different types of shot boundaries that cannot be correctly identified by the traditional pixel-level comparison method.

Case 1. Camera Panning and Tilting

Figure 6 gives an example of the camera panning while tilting. Figure 6(a) is the original video sequence, and Figure 6(b) is the corresponding segmentation mask maps for (a). In this case, the pixel-level comparison will identify too many incorrect shot cuts since the “objects” in the shot move and turn from one frame to another. But as we can see from Figure 6, the segmentation mask maps can still represent the contents of the video frames very well. Since the segmentation mask maps are binary data, it is very simple and fast to compare the two mask maps of the successive frames. Moreover, by combining the object-tracking method, most of the segment movements can be tracked so that we know that there is no major shot boundary if the segments in two successive frames can be tracked and matched well according to the object-tracking method mentioned in Section 2.2.

Figure 6. An example of a video sequence for camera panning and tilting: (a) the temporal order of the sequence is from the top-left to the right-bottom; (b) the corresponding segmentation mask maps for the video sequence shown in (a)


(a)

(b)

Case 2. Zoom In and Zoom Out

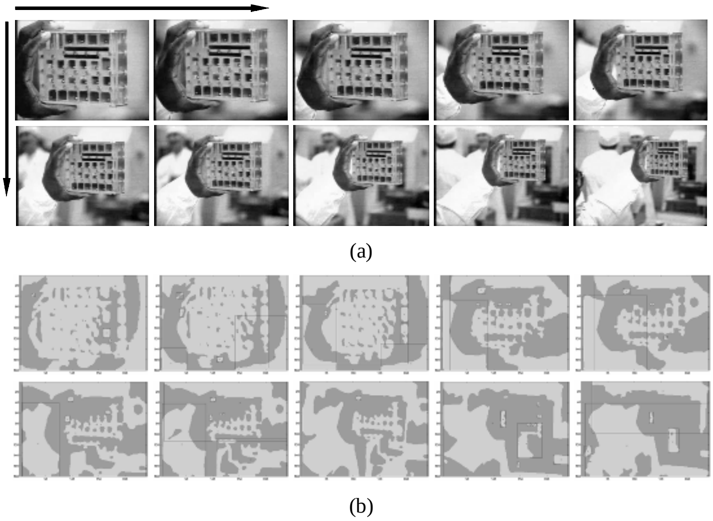
Figure 7 gives an example of a video sequence of camera zooming out. Similarly, we also apply the combination of the segmentation and object tracking to identify this sequence as a single shot.

Case 3. Fade In and Fade Out

Figure 8 gives an example of a video sequence for shots fading out. We still can identify this video sequence as one shot (the shot cut is marked by dotted border in Figure 8). This is a good example to show that the proposed segmentation, together with object-tracking technique, is not sensitive to luminance changes.

In Figure 9, a fancier example of a video sequence is given to show the effectiveness of the proposed method. In this example, one shot is fading in, while another shot is fading out continuously. By applying the proposed method, this sequence is divided into three

Figure 7. An example of a video sequence of zooming out: (a) the temporal order of the sequence is from the top-left to the right-bottom; (b) the corresponding segmentation mask maps for the video sequence shown in (a)



(a)

(b)

Figure 8. An example of a video sequence of fading in (the frame with dotted border is shot cut detected by the proposed method)

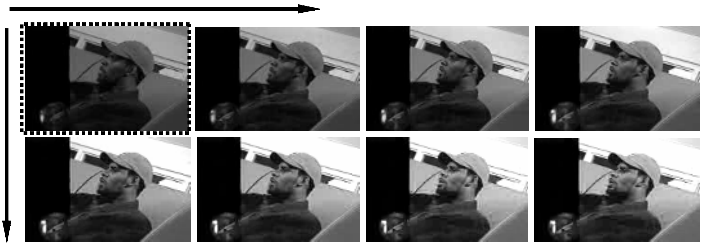
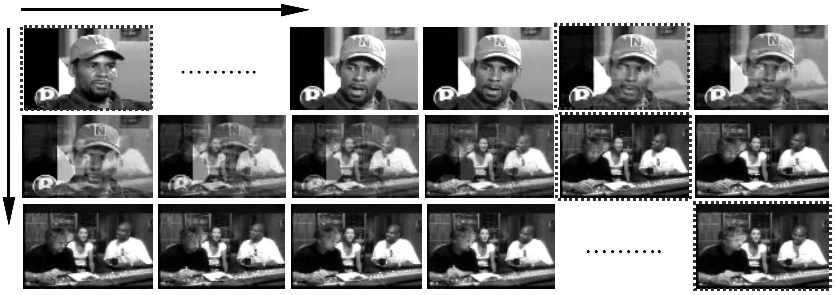


Figure 9. An example of a video sequence of continuously transforming from one shot to another shot (the frames with dotted borders are shot cuts detected by the proposed method)



different shots, and the identified shot cuts are marked by dotted borders as shown in Figure 9. The first shot and the third shot are clearly and correctly identified, while the second shot cut represents the intermediate transforming process from the first shot to the third shot. In our experiments, this kind of video sequences can be divided into either two or three shots. In case of two shots, the intermediate transforming sequence belongs to either the previous shot or the following shot.

The performance is given in terms of *precision* and *recall* parameters. N_C means the number of correct shot boundary detections, N_E means the number of incorrect shot boundary detections, and N_M means the number of missed shot detections.

$$precision = \frac{N_C}{N_C + N_E}$$

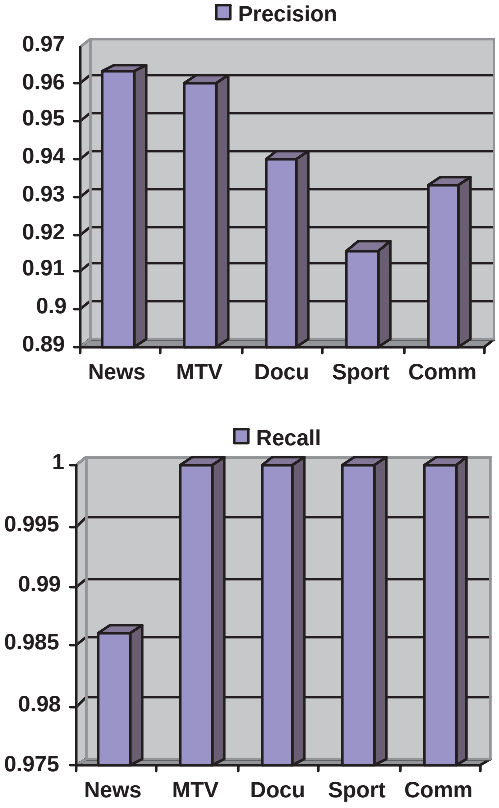
$$recall = \frac{N_C}{N_C + N_M}$$

The summary of the proposed method is shown in Table 2 and Figure 10 via the *precision* and *recall* parameters. In our experiments, the *recall* and the *precision* values are both above ninety percent. Our results are comparable to other techniques such as the PM method in Lee et al. (2000) and DC method in Yeo and Liu (1995). Moreover, the *recall* results seem very stable and promising because most of the *recall* results are 100 percent. The DC method is very sensitive to luminance and color change, but the proposed method is not. As seen in Table 2, the precision values for sports and commercial videos are a little lower (but still above 90 percent) than other types of videos because there are lots of fast movements and fancy transformation between successive frames. As mentioned before, the method of using low-level features is very sensitive to

Table 2. Precision and Recall Parameters

Name	Type	Precision	Recall	Computation Reduce by Pixel-level Comparison
News1	News	0.95	1.00	72%
News2	News	0.96	0.96	75%
News3	News	0.98	1.00	80%
Labwork	Documentary	0.94	1.00	80%
Robert	MTV	0.96	1.000	70%
Carglass	Commercial	0.933	1.000	60%
Aussie2g2	Sports	0.950	1.000	70%
Flo1	Sports	0.889	1.000	60%
Flo2	Sports	0.909	1.000	67%
AligoISA	Sports	0.910	1.000	53%

Figure 10. Average results of parameters Precision *and* Recall *for different types of video clips (News, MTV, Documentary, Commercial and Sports)*



luminance and color change, but our segmentation-based method is not. One thing that should be mentioned here is that even it is efficient to simply compare the segmentation mask maps, the employment of the object-tracking technique is very useful in case of camera panning and tilting. It helps to reduce the number of incorrectly identified shot cuts. Another thing is that by combining the pixel-level comparison, the number of the video frames that need to do segmentation and object tracking is greatly reduced. As can be seen from Table 2, the percentage of the reduced frames that do not need segmentation and object tracking is between fifty percent and eighty percent.

Moreover, the process produces not only the shot cuts, but also the object-level segmentation results. Each detected shot cut is selected as a key frame and has been modeled by the features of its segments such as the bounding boxes and centroids. Based on this information, we can further structure the video content using some existing multimedia semantic model such as the multimedia augmented transition network (MATN) model (Chen, Shyu, & Kashyap, 2000).

FUTURE TRENDS

Video shot segmentation is the first step towards automatic annotation and indexing of digital video for efficient browsing and retrieval. It is an active area of research combining the techniques from image processing, computer vision, pattern recognition, etc. Some limitations of the existing work, that imply the future trends, are summarized as follows:

1. *Distinguish gradual transitions from object and camera motions*: While early work focused on shot cut detection, more recent work tries to deal with gradual transition detection. However, although many existing algorithms can confirm the existence of gradual transitions, they still have difficulties in recognizing the different types of gradual transitions due to the following three reasons: (a) the increasing varieties in gradual transition styles; (b) the similar temporal change patterns between gradual transitions and camera/object motions; and (c) the existence of long transitions in which the differences between frames decreases when the transition length increases. In addition to transition detection, the recognition of camera motion is another important issue in video segmentation. Further improvement to these issues can be achieved by combining multi-modal information such as audio and text (closed captioned) information. Moreover, since one single technique cannot capture all the rich information of a video, an integrated approach combining multiple techniques is a possible and practical solution to this problem.
2. *Benchmark video sequences and evaluation criteria*: It is critical to develop a unified evaluation criteria and benchmarks for video segmentation and video database management systems that allows the evaluation and comparison of various techniques. The benchmark video sequences should contain various types of videos including enough representatives of different types of object motions, camera operations, and gradual transitions. The evaluation criteria need to be quantified and take into consideration the special requirements of specific application domains.
3. *Adjust the thresholds automatically*: Since there are many thresholds involved in video shot boundary detection methods, especially in those methods combining several techniques, how to adjust the thresholds automatically with respect to different characteristics of different video sequences is a big challenge. The development of the methods that can automatically adjust the thresholds by the self-learning process is desired.
4. *Independence of specific video formats*: Almost all of the "real-time" video shot boundary detection methods are conducted in compressed-domain, especially in the MPEG format. Some embedded parameters, such as the DC coefficients, can be directly used for shot boundary detection such that the effort of decoding the video data can be reduced. However, this kind of methods are highly encoder-dependent in that different types of compressed video data (MPEG, RealPlayer, etc.) need different techniques for video parsing and shot detection even though they contain the same content. A format-independent technique is more desirable when considering the fast emergence of new compressed video formats.

CONCLUSIONS

In this chapter, an innovative video shot boundary detection method is presented. Our shot boundary detection method uses the unsupervised segmentation algorithm, object-tracking technique, and a matching process that compares the segmentation mask maps between two successive video frames. The unsupervised segmentation algorithm is applied to automatically extract the significant objects (or regions) of interests and the segmentation mask maps of each video frame. The object-tracking technique is employed as a complement to handle the situations of camera panning and tilting without any extra overhead. Experiments on various different types of sample MPEG-1 video clips were performed. The experimental results show that, unlike other methods that use the low-level features of the video frames, our method is not sensitive to the small changes in luminance or color. Also, our method has high precision and recall values. Most importantly, our method can obtain object-level information of the video frames and accurate shot boundary detection, which are very useful for video content indexing.

ACKNOWLEDGMENTS

For Shu-Ching Chen, this research was supported in part by NSF CDA-9711582, NSF EIA-0220562, NSF HRD-0317692, and the office of the Provost/FIU Foundation. For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260. For Chengcui Zhang, this research was supported in part by SBE-0245090 and the UAB ADVANCE program of the Office for the Advancement of Women in Science and Engineering.

REFERENCES

Alattar, A.M. (1997). Detecting fade regions in uncompressed video sequences. In *Proceedings of 1997 IEEE International Conference on Acoustics Speech and Signal Processing*, 3025-3028.

Arman, F., Hsu, A., & Chiu, M.-Y. (1993). Image processing on compressed data for large video databases. In *Proceedings of First ACM International Conference on Multimedia*, 267-272.

Chen, S.-C., Shyu, M.-L., & Kashyap, R. L. (2000). Augmented transition network as a semantic model for video data. *International Journal of Networking and Information Systems, Special Issue on Video Data*, *3*(1), 9-25.

Chen, S.-C., Sista, S., Shyu, M.-L., & Kashyap, R.L. (1999). Augmented transition networks as video browsing, models for multimedia databases and multimedia information systems. *11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'99)*, 175-182, November 9-11.

Chen, S.-C., Sista, S., Shyu, M.-L., & Kashyap, R. L. (2000). An indexing and searching structure for multimedia database systems. *IS&T/SPIE conference on Storage and Retrieval for Media Databases 2000*, 262-270, January 23-28.

Furht, B., Smoliar, S.W., & Zhang, H.J. (1995). *Video and image processing in multimedia systems*. Boston, MA: Kluwer Academic Publishers.

Gall, D.L. (1991). MPEG: A video compression standard for multimedia applications. *Communications of the ACM, 34*(1), 46-58.

Gargi, U., Kasturi, R., & Antani, S. (1998). Performance characterization and comparison of video indexing algorithms. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 559-565.

Gunsel, B., Ferman, A.M., & Tekalp, A.M. (1998). Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging, 7*(3), 592-604.

Hwang, T.-H., & Jeong, D.-S. (1998). Detection of video scene breaks using directional information in DCT domain. *Proceedings of the 10th International Conference on Image Analysis and Processing*, 882-886.

Lee, S.-W., Kim, Y.-M., & Choi, S.-W. (2000). Fast scene change detection using direct feature extraction from MPEG compressed videos. *IEEE Trans. on Multimedia, 2*(4), 3178-3181.

Nagasaka, A., & Tanaka, Y. (1995). Automatic video indexing and full-video search for object appearances. *In Visual Database Systems II*, 113-127. New York: Elsevier.

Ngo, C.-W., Pong, T.-C., & Chin, R. T. (2000). Motion-based video representation for scene change detection. *Proceedings of the International Conference on Pattern Recognition (ICPR'00)*, 1827-1830.

Pass, G., & Zabih, R. (1999). Comparing images using joint histograms. *ACM Multimedia Systems, 7*(3), 234-240.

Shahraray, B. (1995). Scene change detection and content-based sampling of video sequences. In *Proceedings of SPIE'95, Digital Video Compression: Algorithm and Technologies*, 2419, 2-13, San Jose, CA.

Sista, S., & Kashyap, R.L. (1999). Unsupervised video segmentation and object tracking. *IEEE International Conference on Image Processing*.

Swain, M. J. (1993). Interactive indexing into image databases. In *Proceedings of SPIE Conference Storage and Retrieval in Image and Video Databases*, 173-187.

Swanberg, D., Shu, C.F., & Jain, R. (1993). Knowledge guided parsing in video database. In *Proceedings of SPIE'93, Storage and Retrieval for Image and video Databases*, 1908, 13-25, San Jose, CA.

Truong, B. T., Dorai, C., & Venkatesh, S. (2000). New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *Proceedings of the 8th ACM International Conference on Multimedia*, 219-227.

Xiong, W., & Lee, J.C.-M. (1998). Efficient scene change detection and camera motion annotation for video classification. *Computer Vision and Image Understanding, 71*(2), 166-181.

Yeo, B., & Liu, B. (1995). Rapid scene analysis on compressed video. *IEEE Trans. Circuits Systems Video Technol., 5*(6), 533-544.

Zabih, R., Miller, J., & Mai, K. (1995). A feature-based algorithm for detecting and classifying scene breaks. In *Proceedings of ACM Multimedia '95*, 189-200.

Zhang, H., Kankanhalli, A., & Smoliar, S.W. (1993). Automatic partitioning of full-motion video. *Multimedia System, 1*, 10-28.

Zhang, H., & Smoliar, S.W. (1994). Developing power tools for video indexing and retrieval. In *Proceedings of SPIE'94, Storage and Retrieval for Image and video Databases II*, 2185, 140-149, San Jose, CA.

APPENDIX

URL1. www.ibroxfc.co.uk
URL2. http://hsb.baylor.edu/courses/Kayworth/fun_stuff/
URL3. Cincinnati Dockers, www.cincinnatidockers.com
URL4. www.mormino.net/videos/index.php3

Chapter X

A Soft-Decision Histogram from the HSV Color Space for Video Shot Detection

Shamik Sural, Indian Institute of Technology, Kharagpur, India

M. Mohan, Indian Institute of Technology, Kharagpur, India

A.K. Majumdar, Indian Institute of Technology, Kharagpur, India

ABSTRACT

In this chapter, we describe a histogram with soft decision using the Hue, Saturation, Intensity Value (HSV) color space for effective detection of video shot boundaries. In the histogram, we choose the relative importance of hue and intensity depending on the saturation of each pixel. In traditional histograms, each pixel contributes to only one component of the histogram. However, we suggest a soft-decision approach in which each pixel contributes to two components of the histogram. We have done a detailed study of the various frame-to-frame distance measures using the proposed histogram and an Red, Green, Blue (RGB) histogram for video shot detection. The results show that the new histogram has a better shot-detection performance for each of the distance measures. A Web-based application has been developed for video retrieval, which is freely accessible to interested users.

INTRODUCTION

A shot is a continuous sequence of frames captured from the same camera in a video (Hampapur, Jain, & Weymouth, 1994; Nagasaka & Tanaka, 1992). Shot detection is the process of identifying changes in the scene content of a video sequence. Shot detection is fundamental to any kind of video retrieval application since it enables segmentation of a video into its basic components. There are different types of shots, including hard cut, dissolve, and wipe (Patel & Sethi, 1997; Shahraray, 1995). Hard cut is an instantaneous transition from one scene to the next. A dissolve is a transition from one scene to another in which frames from the first scene gradually merge with the frames from the second scene. This is achieved by making a slow transition between a scene and a constant image (fade-out) or between a constant image and a scene (fade-in). Wipe is another common scene break in which a line moves across the screen, with the new scene appearing behind the line.

Ahanger and Little (1996) provide a survey of the field of video indexing. Boreczky and Rowe (1996) compared five different indexing algorithms, namely, the global histogram, region histogram, global histogram using twin-comparison thresholding, motion-compensated pixel difference, and DCT-coefficient difference with respect to shot-change detection. The histograms are generated from grayscale images. Their data set consists of 233 minutes of motion-JPEG video of various types digitized in 320X240 resolution at 30 frames/s. Some manual tuning is done to find suitable ranges of thresholds to generate the operating curves.

Dailianas, Allen, and England (1995) compared algorithms based on histogram differencing, moment invariants, pixel-value changes and edge detection. Lienhart (1998) also did a critical evaluation of the different shot boundary detection algorithms. Gargi, Kasturi, and Strayer (1996), and Gargi,, Kasturi, Strayer, and Antari (2000) compared algorithms based on histograms and MPEG algorithms. They showed that performance of the histogram-based algorithms is better than that of the other various approaches to shot detection.

In this chapter, we describe a novel method of histogram generation from the Hue, Saturation, Intensity Value (HSV) -color space and study its performance in the shot boundary detection problem.

BACKGROUND

Analysis of the HSV Color Space

A three-dimensional representation of the HSV color space is a hexacone (Shapiro & Stockman, 2001), where the central vertical axis represents intensity, I. Hue, H, is defined as an angle in the range [0,2p] relative to the red axis with red at angle 0, green at 2p/3, blue at 4p/3 and red again at 2p. Saturation, S, is the depth or purity of color and is measured as a radial distance from the central axis with value between 0 at the center to 1 at the outer surface. For S=0, as one moves higher along the intensity axis, one goes from Black to White through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all pixels, even with different hues, look alike and as we increase the saturation towards 1, they tend to get separated out and are visually perceived as the true colors represented by their hues.

For low values of saturation, we can approximate a color by a gray value specified by the intensity level, while for high saturation, the color can be approximated by its hue

(Sural, Qian, & Pramanik, 2002). The saturation threshold that determines this transition is once again dependent on the intensity. For low intensities, even for a high saturation, a color is close to the gray value and vice versa. Saturation gives an idea about the depth of color, and the human eye is less sensitive to its variation compared to variation in hue or intensity. We, therefore, use the saturation value of a pixel to determine whether the hue or the intensity is more pertinent to human visual perception of the color of that pixel and ignore the actual value of the saturation.

We would like to emphasize the fact that our approach treats the pixels as a distribution of "colors" in an image where a pixel may be of a "gray color" (i.e., somewhere between black and white, both inclusive) or of a "true color" (i.e., somewhere in the red→green→blue→red spectrum). These visual properties of the HSV space as described above are central to our method of histogram generation for video shot detection. By doing this, we are able to capture both color and intensity information in the same histogram. This makes our histogram more suitable for video shot detection compared to other existing methods.

Video Shot Detection

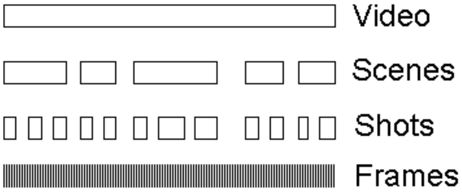
A video consists of a number of scenes that in turn is a logical grouping of shots into a semantic unit. A shot includes a number of frames that are the individual consecutive images captured from the same camera. The structural relation between a video, scenes, shots, and frames is shown in Figure 1.

Since a complete video in its basic form is just a collection of individual frames, we need to identify the group of adjacent frames that are captured by the same camera having temporal proximity. This group of adjacent frames captured using the same camera is called a shot. Evidently, detection of individual shots is a primary requirement in video processing. The effectiveness of the subsequent steps is critically dependent on the success of the shot detection step.

Shot-detection algorithms can be classified according to the features used for processing into uncompressed and compressed domain algorithms (Kuo & Chen, 2000; Lienhart, 1998; Yeo & Liu, 1995). Algorithms in the uncompressed domain utilize information directly from the spatial video domain. On the other hand, algorithms in the compressed domain extract shot boundary information from the transformed frequency coefficients stored in the compressed video files.

Pixel comparison, which is also called template matching, was introduced to evaluate the differences in intensity or color values of corresponding pixels in two successive frames (Hampapaur et al., 1994). The main drawback of this method is that it

Figure 1. Structural relation between video, scene, shot and frame



is very sensitive to object and camera movements and noises. In the block-based approaches, each frame is divided into a number of blocks that are compared against their counterparts in the successive frames. Obviously, this approach provides a better tolerance to slow and small motions between frames. The frame-based approach summarizes the whole frame into one measure and compares this with the same measure for the next frame (Patel & Sethi, 1997).

To further reduce the sensitivity to object and camera movement and thus provide a more robust shot detection technique, histogram comparison was introduced (Shahraray, 1995). A histogram of a frame is calculated as the frequencies of pixel occurrences of similar color in a given color space. Histograms do not depend on the spatial layout of picture but are dependent only on the pixel frequencies.

In the edge-detection method, a frame is turned into a grayscale image. An edge-detection algorithm is then applied to that image, and the difference value is calculated for two adjacent frames. If the difference is above a certain threshold, a shot change is detected at that frame position.

Yeo and Liu (1995) proposed an initial method for compressed domain shot detection using a sequence of reduced images extracted from DC coefficients in the transformation domain called the DC sequence. An interesting approach was also proposed by Lee, Kim, and Choi (2000), where they exploited information from the first few AC coefficients in the transformation domain and tracked binary edge maps to segment the video. The macroblock-based approach works on compressed MPEG digital video (Pei & Chou, 2002). In an MPEG video, the frame is split into fixed regions called macroblocks. The three types of macroblocks are I, B, and P macroblocks. I macroblock is encoded independently of other macroblocks. P encodes not the region but the motion vector and the error block of the previous frame. B is the same as above except that the motion vector and error block are encoded from the previous or the next frame.

COLOR HISTOGRAM GENERATION WITH SOFT DECISION

Color is considered an important feature of any image and many content-based image retrieval systems make effective use of image color (Deng, Manjunath, Kenney, Moore, & Shin, 2001; Gevers & Smeulders, 2000; Smeulders, Worring, Santini, Gupta, & Jain, 2000; Wang, Li, & Wiederhold, 2001). A standard way of generating a color histogram from an image is to concatenate "N" higher order bits of the Red, Green and Blue values in the RGB space (Swain & Ballard, 1991). It is also possible to generate three separate histograms, one for each channel, and to concatenate them into one as proposed by Jain and Vailaya (1996). In the QBIC system (Niblack et al., 1993), each axis of the RGB color space is quantized in a predefined number "K" of levels, giving a color space of K^3 cells. After computing the center of each cell in the Mathematical Transform of Munsell (MTM) coordinates, a clustering procedure partitions the space in super-cells. The image histogram represents the normalized count of pixels falling in each super-cell. Ortega et al. (1998) have used the HS co-ordinates to form a two-dimensional histogram from the HSV color space. The H and S dimensions are divided into N and M bins, respectively, for a total of NxM bins.

In contrast to the above methods, we generate a one-dimensional histogram from the HSV space in which a perceptually smooth transition of color is obtained in the feature vector. In the proposed histogram, each pixel in an image contributes weighted values of its hue and intensity based on its saturation. The generated histogram consists of "true color" components and "gray color" components, which store contributions from the hue and the intensity of each pixel. Earlier, we proposed a histogram with a hard-decision threshold to determine if we should increment the count of a "true color" component or a "gray color" component in the histogram for each pixel (Sural, Qian, & Pramanik, 2002). However, this method does not completely model human perception of boundary colors near the saturation threshold.

As an example, if the value of threshold is 0.2, a pixel having a saturation of 0.19 is considered a "gray color" pixel. This may be the correct representation for some human observers, but for others the pixel could visually appear to be a "true color" pixel. In order to capture this fuzzy nature of the human perception of color, we felt the need for a soft decision in determining the dominant property of a pixel. In our current work, we update two components of the histogram — one "true color" component and one "gray color" component for each pixel in an image. The quantum of update, i.e., weight of each component, is determined by the saturation and the intensity of the pixel. The sum of the weights of the two contributions equals unity. Also, for the same saturation, the weight should vary with intensity. For a lower intensity value, the same value of saturation should give a lower weight on the "true color" component and vice versa. To incorporate dependency of the weight function on the saturation and intensity, we make it a function $w_H(S,I)$ of two variables, S and I, that should satisfy the following properties:

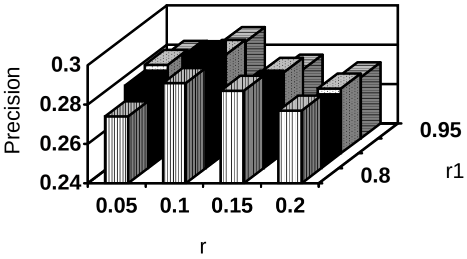
- $w_H(S,I) \in [0,1]$
- For $S_1 > S_2$, $w_H(S_1,I) > w_H(S_2,I)$
- For $I_1 > I_2$, $w_H(S,I_1) > w_H(S,I_2)$
- $w_H(S,I)$ changes slowly with S for high values of I
- $w_H(S,I)$ changes sharply with S for low values of I

To satisfy the above constraints, we consider the following function to determine the weight of the hue components.

$$w_H(S,I) = \begin{cases} s^{r(255/I)^{r_1}} & \text{for } I \neq 0 \\ 0 & \text{for } I = 0 \end{cases} \qquad \textbf{(1)}$$

Here r, $r_1 \in [0,1]$. We have done extensive studies on the possible combination of the two parameters of Eq. (1). The precision of retrieval in a content-based image retrieval application is plotted in Figure 2 for a number of r and r1 values. It is seen that a combination of r = 0.1 and $r_1 = 0.85$ has the best representation of the fuzzy nature of the human perception of the three dimensions in the HSV color space. We, therefore, chose these values of r and r_1 in our system. The histogram with soft decision from the HSV color space presented here is denoted by HSVSD in the rest of the chapter.

Figure 2. Variation of precision for the HSVSD histogram with the two parameters r and r_1



The weight of the intensity component $w_I(S,I)$ is then derived as:

$$w_I\,(S,I) = 1 - w_H\,(S,I) \qquad \textbf{(2)}$$

Thus, for a given pixel, after we determine its H, S, and I values, we calculate the "true color" weight and the "gray color" weight to update the histogram. The "true color" components in the histogram may be considered circularly arranged, since hue ranges from 0 to 2π, both the end points being red. The histogram, thus, may be represented as a logical combination of two independent vectors. The algorithm for generating the color histogram (Hist) can now be written as shown below.

```
For each pixel in the image

    Read the RGB value
    Convert RGB to Hue(H), Saturation(S) and Intensity Value(I)
    Determine $w_H(S,I)$ and $w_I(S,I)$ using Eqs.        (1)-(2)
    Update histogram as follows:
    Hist[Round(H.MULT_FCTR)] = Hist[Round(H.MULT_FCTR)] + $w_H(S,I)$
    Hist[$N_T$ + Round(I/DIV_FCTR)]= Hist[$N_T$ + Round(I/DIV_FCTR)] + $W_I(S,I)$
```

In the above algorithm MULT_FCTR and DIV_FCTR determine the quantization levels of the "true color" components and the "gray color" components, respectively. N_T is the total number of "true color" components and is given by:

$$N_T = \text{Round}\,(2\text{pMULT\_FCTR}) + 1.$$

Histogram Distance Measures for Shot Detection

We have evaluated the following measures for determining frame-to-frame distance during shot boundary detection:

a. *Bin-to-Bin difference (BTB).* Given two histograms h_1 and h_2, the bin-to-bin method is used to calculate frame-to-frame difference as follows:

$$\mathrm{fd}_{b2b}(h_1,h_2)= \frac{1}{N}\sum_i \quad |h_1[i]-h_2[i]| \tag{3}$$

where N is the number of pixels in each frame.

b. *Chi-square test histogram difference (CHI).* The histogram bin difference values are normalized to sharpen the frame differences being computed. The authors in Gargi et al. (2000) justify this to make the evaluation value more strongly reflect the difference of two frames. The distance is defined as follows.

$$\mathrm{fd}_{chi} = \quad \frac{1}{N^2}\sum_i \frac{|\,h_1[i]-h_2[i]\,|}{h_2[i]} \quad \text{for } h_2[i] \mathrel{!}= 0$$

$$\frac{1}{N^2}\sum_i \frac{|\,h_1[i]-h_2[i]\,|}{h_1[i]} \quad \text{for } h_2[i] \ = 0 \tag{4}$$

c. *Histogram Intersection (INT) distance.* The intersection and corresponding frame difference between two color histograms are defined as follows:

$$\text{intersection }(h_1,h_2)= \frac{\sum_i \min(h_1[i], h_2[i])}{N} \tag{5}$$

$$\mathrm{fd}_{int}(h_1,h_2)= 1- \text{intersection}(h_1,h_2) \tag{6}$$

d. We also use a Vector Cosine Angle Distance (VCAD) measure for calculating frame-to-frame difference. This distance is defined as follows.

$$\mathrm{fd}_{vcad} = \frac{h_1 \bullet h_2}{\|h_1\|\|h_2\|} \tag{7}$$

Here $h_1 \bullet h_2$ represents dot product of h_1 and h_2 while $\|h_1\|$ represents norm of the histogram h_1.

Local and Global Selection of Threshold for Shot Detection

In the histogram-based shot detection approach, if the histogram difference between two adjacent frames is greater than a threshold value, a shot transition is detected at the frame position. The problem of choosing an appropriate threshold is a key issue in such shot detection algorithms. If the threshold value is high, then the number of missed boundaries will be high and the number of false positive shots is low. If the threshold value is low, then the number of missed boundaries is low but the false positive shots are high. Heuristically chosen fixed global threshold is often found to be inappropriate because experiments have shown that the threshold for determining a shot boundary varies from one video to another video. Threshold shot boundary values are often dependent on the type of video, such as commercials, movies, news, sports, etc. Therefore, we feel that an adaptive threshold selection method is more reasonable than a fixed global threshold. Hence, in our histogram-based algorithm, we use an adaptive thresholding framework. We use two types of thresholds, namely, the global and the local threshold. In global threshold, only one threshold value is chosen for an entire video. On the other hand, in local threshold selection, we change the threshold depending on the nature of the adjacent frames in a portion of the video.

We use a global adaptive threshold for evaluating the performance of histogram-based shot detection using the HSV color histogram with soft decision.

The adaptive global threshold value is determined as follows:

Let d(m,m+1) denote the distance between the frames m and m+1 and N be the total number of frames in the video. We calculate the adaptive threshold by comparing N_{max} number of frame-to-frame distances where

$$N_{max} = kN;\ k \in (0,1)\ \text{k being a tunable parameter.} \qquad \textbf{(8)}$$

Let the N_{max} number of frame-to-frame distance values be written in the descending order as $d_1, d_2, \ldots, d_{Nmax}$. Then the adaptive threshold is calculated as:

$$Th_a = \frac{p}{N_{max}}\left(\sum_{i=1}^{N_{max}} d_i\right) \qquad \textbf{(9)}$$

Here p is a factor that depends on the histogram type and the inter-frame distance measure chosen. The value of p is different for Bin-to-bin, Chi-square, VCAD, and intersection distance measures for RGB and HSVSD histograms.

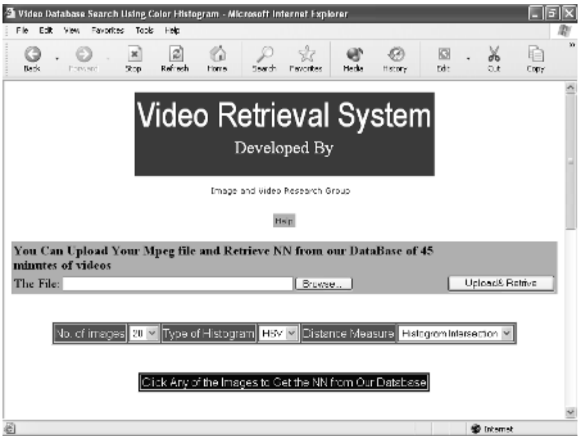
Web-based Video Retrieval System

In order to perform and demonstrate large-scale studies on content-based video retrieval, we have developed a Web-based multimedia application (http://www.imagedb.iitkgp.ernet.in/video/) as shown in Figure 3. In this section, we describe some of the important components of the application.

Video Shot Detection and Key Frame Extraction

We have performed extensive experiments with the proposed HSV-based color histogram with soft decision and an RGB histogram for video shot detection. We have used all the four distance measures discussed in the previous section for comparison. The HSV histogram shows the best performance with the histogram intersection distance

Figure 3. Web-based video retrieval system



Video Database Search Using Color Histogram - Microsoft Internet Explorer

Video Retrieval System

Developed By

Image and Video Research Group

You Can Upload Your Mpeg file and Retrieve NN from our DataBase of 45 minutes of videos

The File:

No. of images 20 Type of Histogram HSV Distance Measure Histogram Intersection

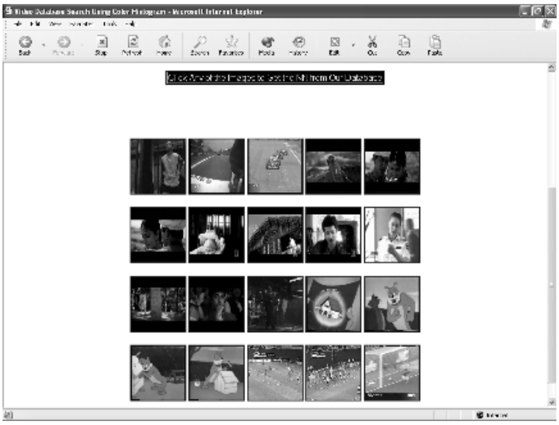
Click Any of the Images to Get the NN from Our Database

measure. In the Web-based video retrieval application, we, therefore, use an HSVSD histogram along with histogram intersection as the distance measure for shot detection. For all the available MPEG images, shots are detected using a combination of local and global thresholding. Key frames are then extracted from the detected shots. The key frames of all the shots for all the video files form a database of representative images in our video database.

Query Specification

A query in the application is specified by an example image as shown in Figure 4. Initially, a random set of 20 images is displayed. Refreshing the page on the Web browser displays a new set of 20 random images. The number of images to be retrieved and

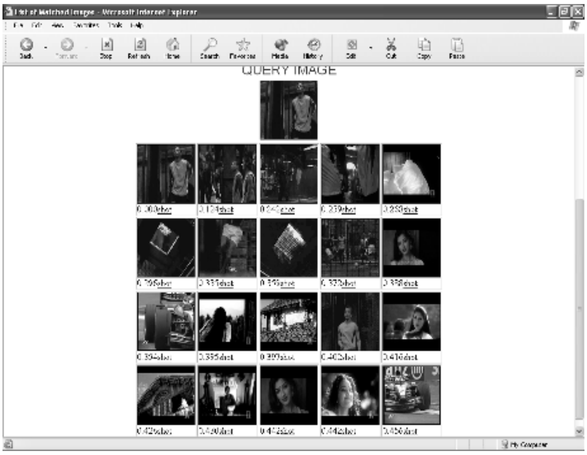
Figure 4. Display of random images for query specification



Video Database Search Using Color Histogram - Microsoft Internet Explorer

Click Any of the Images to Get the NN from Our Database

Figure 5. Result set display in the Web-based video retrieval system



displayed is selected as an input parameter. Thus, instead of 20, we can display 10 or 30 random images at a time. Users can select any of the displayed images as the example image by clicking on it.

Display of Result Set

The nearest neighbor result set is retrieved from the video database based on the clicked query image and is displayed as shown in Figure 5. The distance value from the query image is printed below each image. The retrieval process considers the parameter values selected in the options boxes for the number of nearest neighbors, type of histogram, and the distance metric.

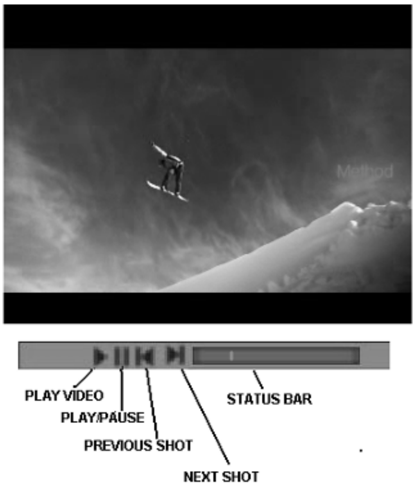
Histogram Display

One of the objectives of our research is to study the properties of different color histograms and how they affect shot boundary detection for a variety of distance metrics. To get an insight into this aspect, we made a provision for displaying the histograms. The "Histogram" hyper link on the result page displays all the retrieved histograms. On each of these result set histograms, we also display the query image histogram for effective comparison.

Shot Playback

Once the set of similar images is displayed to the user, the corresponding shot may be played by clicking on a link available at the bottom of each image. The shot is played on a custom video player that opens on a new HTML page. On this page, we provide buttons to play the next shot, the previous shot, or the entire video MPEG file, along with a play/pause button and a status bar. The video player applet playing a shot is shown in Figure 6.

Figure 6. Custom media player running retrieved shots in the Web-based video retrieval system



External Video Upload

Users are often interested in retrieving video images similar to their own input video frames. To facilitate this, we provide a utility to upload an external video file in our system. We perform shot detection and key frame extraction from the uploaded video. The key frames are displayed as possible example images. By clicking on any of them, we retrieve and display similar images to the user. The users can then play the corresponding shot or the entire video on the Web page. They can also retrieve further similar images.

The Web-based application can be freely accessed by interested readers to check the quality of shot detection, speed of retrieval, and similarity search.

Results

We compare the performance of our shot detection method with a standard RGB histogram approach for all the four distance measures described above. The tests are performed on 20 MPEG videos with the current total number of frames = 59,499 (approximately 33 minutes of video). The correct total number of shots in these videos is 940. The results are shown in Figures 7(a)-(d). In these figures, we have plotted the variation in precision for different values of recall for the two histograms using each of the four distance measures. Here Recall and Precision are defined as follows:

$$\text{Recall} = \frac{NoOfTrueShotsDetected}{TotalNoOfTrueShots}$$

$$\text{Precision} = \frac{NoOfTrueShotsDetected}{TotalNoOfShotsDetected}$$

It is observed that the performance of the proposed HSV-based shot detection algorithm is better than the RGB-based method. It is also observed that bin-to-bin and histogram- intersection methods have a similar performance. Chi-square-based distance has a performance worse than those two methods, while the VCAD-based method has the worst performance. As we try to achieve higher recall by increasing the decision threshold so that all the true shot boundaries are correctly detected, a higher number of false positives (transitions that are not true shot boundaries but are detected as shot boundaries) are also encountered. This results in a lower value of precision. Conversely, if the threshold for determining shot boundaries is increased so that we can reduce the false positives, some of the true shot boundaries are missed resulting in a lower value of recall.

In Figures 8(a)-(d), we plot the effect of variation in recall with precision for the four distance measures separately. It is seen that for all values of recall and precision combination the HSVSD histogram has a higher precision and recall than the RGB histogram. Further, we see that the difference in performance is more pronounced for the histogram intersection and the bin-to-bin distance.

DIRECTIONS FOR FUTURE RESEARCH

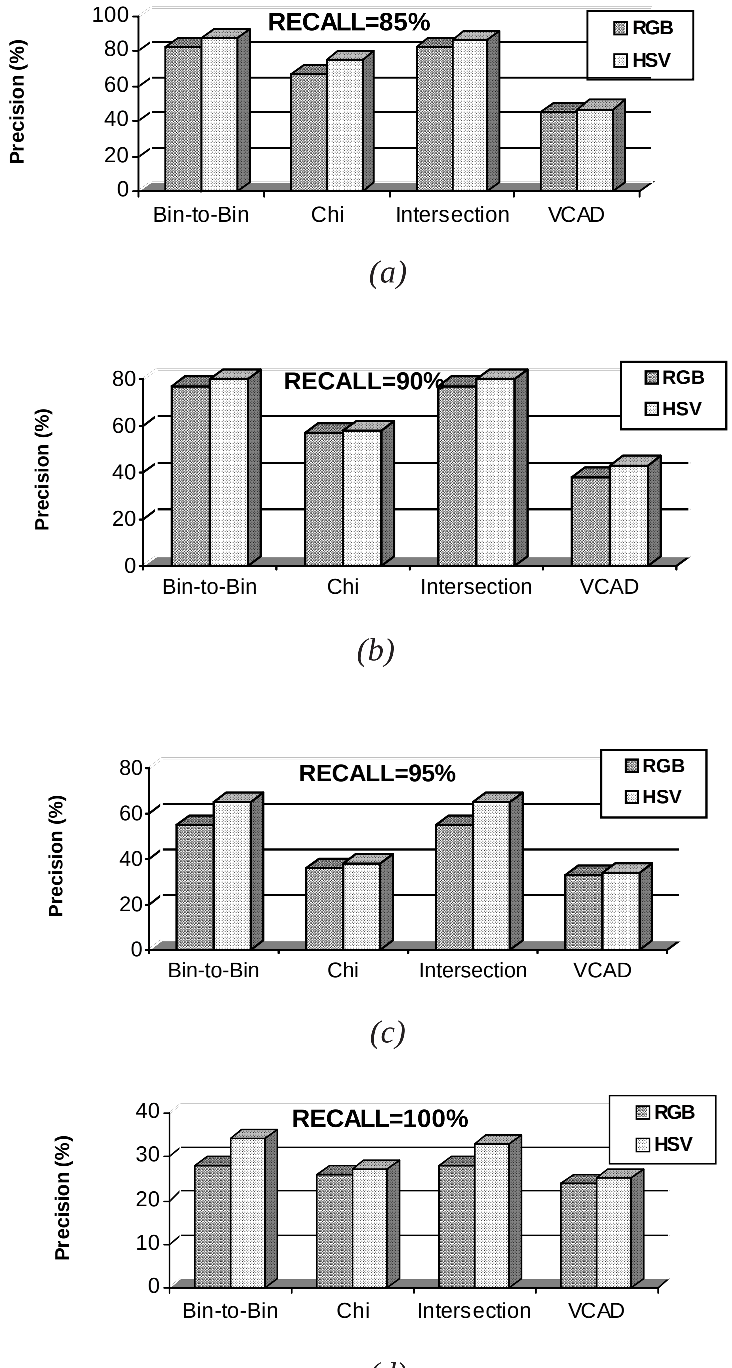
Video data management and information retrieval is an exciting and challenging research area. Researchers from many related fields are contributing to improve the current state-of-the-art. A lot of new areas of research in this field are also coming up. In this section, we suggest some of the research directions that can be considered as direct extensions of the work described in this chapter.

We have evaluated the performance of a shot-change detection method with a novel color histogram generated from the HSV color space using soft decision. The histogram represents visual perception of color more closely than the other existing methods. In order to do an objective comparison of performance, we used four distance measures and showed how the recall and precision vary for the proposed histogram and a histogram generated from the RGB color space. Since the proposed histogram had a better shot detection performance, we are extending our research to the compressed domain. In this approach, the histogram will be generated directly from the MPEG compressed domain to improve the speed of operation. Some of the research groups across the world have started looking into the area of such compressed domain processing.

Some of the other promising research directions in shot detection include the use of entropy (Cernekova, Nikou, & Pitas, 2002), use of linear prediction of motion (Bruno & Pellerin, 2002) and temporal slices (Ngo, Pong, & Zhang, 2002). With the new MPEG standards coming up that include information storage at different layers such as object, text, audio, etc., one can explore the possibility of a multi-modal approach to video shot detection. This approach would try to capture shot boundaries from each of the different modes and use a rule to combine them for final decision making.

Besides the detection of the hard cuts, another research area is exploring the methods of determining the edit cuts such as wipe, fade in, and fade out (Pei & Chou, 2002). Most of the video shot-detection methods do not perform well in the presence of such edit cuts, even though their hard-cut detection performance is quite good. There is a lot of scope for research in the field of such edit-cut detection.

Figure 7. Precision of the RGB and the HSV histograms for different frame-to-frame distance measures (recall values are (a) 85% (b) 90% (c) 95% and (d) 100%)



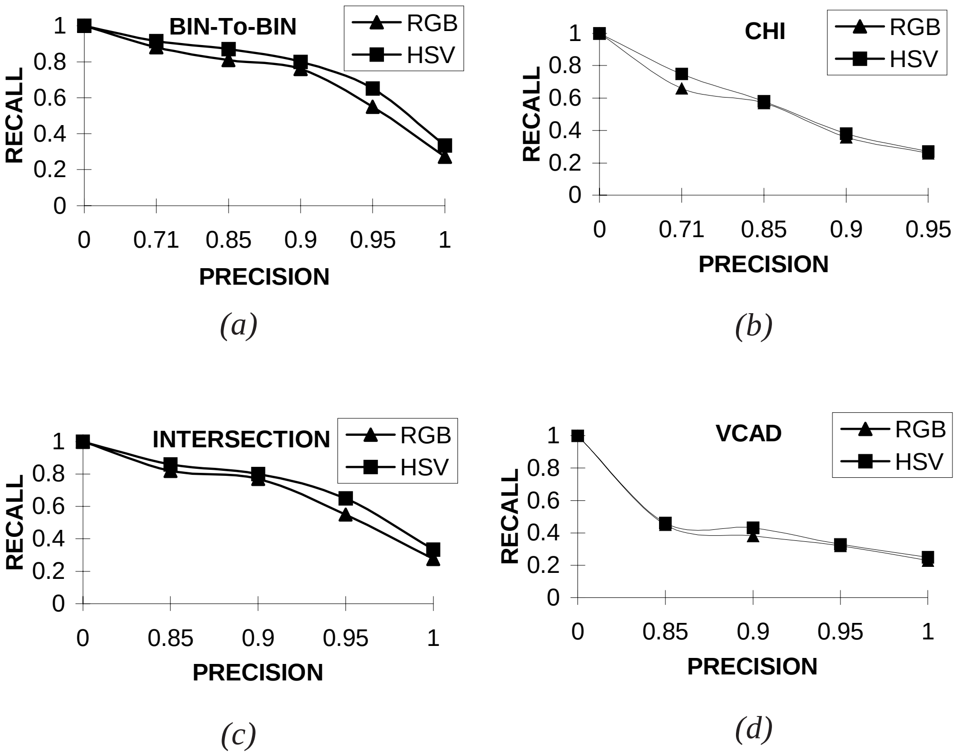
(a)

(b)

(c)

(d)

Figure 8. Variation of recall with precision for the RGB and the HSV histograms for different frame-to-frame distance measures ((a) Bin-to-Bin distance (b) Chi-square distance (c) Histogram intersection, and (d) Vector cosine angle distance)



With the development of efficient and robust algorithms for video processing, a number of new application areas are also emerging. This includes video shot detection in surveillance systems, news and sports video segmentation, medical video processing, and others. Model- based system development satisfying the requirements of such applications is expected to dominate the field of research on video processing.

CONCLUSIONS

In this chapter, we have discussed video shot boundary detection methods using histograms and various distance measures. A new color histogram has been proposed that is generated from the HSV color space using a novel soft-decision approach. The performance of the color histogram has been compared to a standard color histogram for four different distance measures. The new histogram has been found to have a better performance in identifying shot boundaries compared to the traditional histogram.

We have developed a Web-based video shot detection and retrieval application that is available free for use. One of the key features of this application is the ability of the users to load their own video clippings which will be processed by our application for shot boundary detection and subsequent retrieval of images similar to the key frames. The application has been developed using a modular approach so that once other shot boundary-detection algorithms are implemented, we can seamlessly integrate them into the existing system.

In order to make our Web-based application more useful, we are in the process of adding a large number of MPEG files to our system to have at least a few thousand minutes of video available for retrieval.

REFERENCES

Ahanger, G. & Little, T.D.C. (1996). A survey of technologies for parsing and indexing digital video. *Journal of Visual Communication and Image Representation, 7*(1), 28-43.

Boreczky, J.S., & Rowe, L.A. (1996). Comparison of video shot boundary detection techniques. *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases IV*, SPIE 2670, 170-179.

Bruno, E., & Pellerin, D. (2002). Video shot detection based on temporal linear prediction of motion. *Proceedings of the IEEE International Conference on Multimedia and Exposition (ICME),* Lausanne, Switzerland, 289-292.

Cernekova, Z., Nikou, C., & Pitas, I. (2002). Shot detection in video sequences using entropy-based metrics. *Proceedings of the IEEE International Conference on Image Processing*, Rochester, NY, 421-424.

Dailianas, A., Allen, R.B., & England, P. (1995). Comparisons of automatic video segmentation algorithms. *Proceedings of Integration Issues in Large Commercial Media Delivery Systems*, SPIE 2615, 2-16.

Deng, Y., Manjunath, B.S., Kenney, C., Moore, M.S., & Shin, H. (2001). An efficient color representation for image retrieval. *IEEE Transactions on Image Processing, 10*, 140-147.

Gargi, U., Kasturi, R. & Strayer, S.H. (2000). Performance characterization of video-shot-change detection methods. *IEEE Transactions on Circuits and Systems for Video Technology, 10*(1), 1-13.

Gargi, U., Kasturi, R., Strayer, S.H. & Antani, S. (1996). *An evaluation of color histogram based methods in video indexing*. Department of Computer Science & Engineering, Penn State University, University Park, PA, Tech. Rep. CSE-96-053.

Gevers, T., & Smeulders, A.W.M. (2000). PicToSeek: Combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing, 9*, 102-119.

Hampapur, A., Jain, R., & Weymouth, T. (1994). Digital video segmentation. *Proceedings of the ACM International Conference on Multimedia,* 357-364.

Jain, A., & Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition, 29*, 1233-1244.

Kuo, T.C.T. & Chen, A.L.P (2000). Content-based query processing for video databases. *IEEE Transactions on Multimedia, 2*(1), 240-254.

Lee, S.-W., Kim, Y.-M., & Choi, S.-W. (2000). Fast scene change detection using direct feature extraction from MPEG compressed videos. *IEEE Transactions on Multimedia*, *2*(4), 240-254.

Lienhart, R. (1998). Comparison of automatic shot boundary detection algorithms. *Proceedings of Storage and Retrieval for Image and Video Databases VII*, SPIE Vol. 3656, 290-301.

Nagasaka, A., & Tanaka, Y. (1992). Automatic video indexing and full-video search for object appearances. *Proceedings of the IFIP 2nd Working Conference Visual Database Systems*, 113-127.

Ngo, C.-W., Pong, T.-C., & Zhang, H.-J. (2002). On clustering and retrieval of video shots through temporal slices analysis. *IEEE Transactions on Multimedia*, *4*(4), 446-458.

Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P., Faloutsos, C., & Taubin, G. (1993). The QBIC project: Querying images by content using color, texture and shape. *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, 1908, 173-187.

Ortega, M., Rui, Y., Chakrabarti, K., Warshavsky, A., Mehrotra, S., & Huang, T.S. (1998). Supporting ranked Boolean similarity queries in MARS. *IEEE Transactions on Knowledge and Data Engineering*, *10*, 905-925.

Patel, N.V., & Sethi, I.K. (1997). Video shot detection and characterization for video databases. *SPIE Storage and Retrieval for Image and Video Databases*, 218-225.

Pei, S.-C., & Chou, Y.-Z. (2002). Effective wipe detection in MPEG compressed video using macroblock type information. *IEEE Transactions on Multimedia*, *4*(3), 309-319.

Shahraray, B. (1995). Scene change detection and content-based sampling of video sequences. *Proceedings of SPIE/IS&T Symp. Electronic Imaging Science and Technology: Digital Video Compression, Algorithms and Technologies*, 2419, 2-13.

Shapiro, L., & Stockman, G. (2001). *Computer vision*. Englewood Cliffs, NJ: Prentice Hall.

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 Transactions on Pattern Analysis and Machine Intelligence*, *22*, 1-32.

Sural, S., Qian, G., & Pramanik, S. (2002a). A histogram with perceptually smooth color transition for image retrieval. *Proceedings of the Fourth International Conference on Computer Vision, Pattern Recognition and Image Processing*, Durham, NC, 664-667.

Sural, S., Qian, G., & Pramanik, S. (2002b). Segmentation and histogram generation using the HSV color space for content-based image retrieval. *Proceedings of the IEEE International Conference on Image Processing*, Rochester, NY, 589-592.

Swain, M., & Ballard, D. (1991). Color indexing. *International Journal of Computer Vision*, *7*, 11-32.

Wang, J.Z., Li, J., & Wiederhold, G. (2001). SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 947-963.

Yeo, B.-L., & Liu, B. (1995). Rapid scene analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, *5*(6), 533-544.

Section V

Video Feature Extractions

Chapter XI

News Video Indexing and Abstraction by Specific Visual Cues: MSC and News Caption

Fan Jiang, Tsinghua University, China

Yu-Jin Zhang, Tsinghua University, China

ABSTRACT

This chapter addresses the tasks of providing the semantic structure and generating the abstraction of content in broadcast news. Based on extraction of two specific visual cues, Main Speaker Close-Up (MSC) and news caption, a hierarchy of news video index is automatically constructed for efficient access to multi-level contents. In addition, a unique MSC-based video abstraction is proposed to help satisfy the need for news preview and key-person highlighting. Experiments on news clips from MPEG-7 video content sets yield encouraging results that prove the efficiency of our video indexing and abstraction scheme.

INTRODUCTION

Along with the fast development and wide application of multimedia technology, digital video libraries and archives of immense size are becoming available over networks. How to efficiently access video content has become crucially important due to the so-called “Information Explosion.” However, because of the great length and rich content of video data, finding ways to ensure the quick grasping of a global picture of the data or effective retrieval of pertinent information is not an easy task. This challenge has led to the increasingly active research on the subject of Content-Based Video Retrieval (CBVR). So far, although considerable progress has been made in video analysis, representation, browsing and retrieval, which are fundamental techniques for accessing video content, a universal solution for high-level analysis is still very difficult to achieve. Researchers have been focusing on specific applications utilizing domain knowledge in video programs such as sports, movies, commercial advertisements, and news broadcasts. Since the temporal syntax of news video is generally very straight, some prior knowledge of broadcast news, as well as a number of specific visual cues would help identify the content structure.

The earlier related work focused on news video story parsing based on well-defined temporal structures in news video (Furht, Smoliar, & Zhang, 1995). With the progress in machine vision and image analysis techniques, several news video parsing tools and systems have been developed. Most of these proposed approaches allow automatic or semi-automatic parsing and annotation of news video for many kinds of applications, such as interactive news navigation, content-based retrieval and required news browsing. A few works along this line make use of the information from a single medium, such as visual cues (Rui, 1999; Yeung & Yeo, 1996), linguistic cues (Huang, Liu, & Rosenberg,1999), and text information (Greiff, Morgan, Fish, Richards, & Kundu, 2001; Sato, Kanade, Hughes, & Smith, 1999). Others consider integrating cues from different media to achieve more robust and precise content structuring and indexing results (Boykin & Merlino, 1999; Faudemay, Durand, Seyrat, & Tondre, 1998; Hanjalic, Kakes, Lagendijk, & Biemond, 2001; Jasinschi, Dimitrova, McGee, Agnihotri, Zimmerman, & Li, 2001; Qi,, Gu, Jiang, Chen, & Zhang, 2000; Raaijmakers, de Hartog, & Baan, 2002; Zhang & Kuo, 1999). Despite of some reported progress in the literature, lots of difficulties remain in the construction of semantic structure. In addition, a unified framework to represent pertinent information from video data concisely and efficiently is still lacking. Besides, a compact representation of a video sequence — a video abstract — can provide a quick overview of the content. While the need for news video abstraction is strong, how to acquire an efficient feasible abstraction remains far from being solved.

In this chapter, the above problems will be addressed. First, a hierarchical structure for news content indexing is proposed. Some novel techniques, including speaker close-up detection and caption localization in video stream, are then gathered together into this framework to facilitate efficient video browsing and retrieval. In contrast to some other structures, the one proposed in this chapter exploits only high-level visual cues such as anchors, captions, and news icons that are particular to the context of broadcast news. Not only are more semantic hints given by these specific visual cues than low-level ones (e.g., color and texture of image), but the proposed scheme also puts enough emphasis on the logical continuity of adjacent content units to accomplish meaningful results. Finally, based on the hierarchical structure and a table of constructed speaker close-ups, a multi-level abstraction in the key persons phase is provided to fulfill the needs for specific human-pertinent overview.

The rest of this chapter is organized as follows:

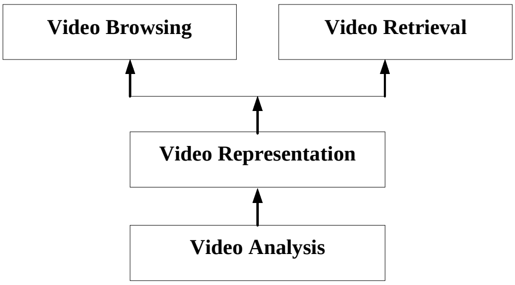
- Background
    - Shot Boundary Detection,
    - Anchorperson Detection,
    - Video OCR,
    - Video Representation and Abstracting;
- Main Thrust of the Chapter
    - Hierarchical Structure for News Content Indexing,
    - Main Speaker Close-Up (MSC) Shot Detection,
    - News Caption Detection,
    - Integrated Analysis Using Two Visual Cues,
    - Video Representation and Abstraction by Selective MSCs,
    - Experimental Results and Evaluation;
- Future Trends;
- Conclusion.

BACKGROUND

A general CBVR system consists of four essential modules: video analysis, video representation, video browsing, and video retrieval. Video analysis deals with the signal processing and analysis of the video, involving shot boundary detection, key frame extraction, etc. Video representation is concerned with the structure of the video. An example of a video representation is based on the hierarchical tree-structured model (Rui, Huang, & Mehrotra, 1998). Built on top of the video representation, video browsing deals with how to use the representation structure to help viewers browse the video content. Finally, video retrieval is concerned with retrieving interesting video objects. The relationship among these four modules is illustrated in Figure 1 (Rui & Huang, 2000).

In order to achieve these goals oriented by above directions, certain features including visual, acoustic, and textual features need to be extracted from video data and appropriately utilized to guide our determination at different stages of processing. As video's primary distinction from speech and text information, visual information understandably plays an important role in the analysis process. Statistical visual features,

Figure 1. Relationships among the four research areas



involving frame difference and camera motion, are traditionally used to solve primary problems, such as shot boundary detection (Furht et al., 1995) and key frame extraction (Zhang, Low, Smoliar, & Wu, 1995), without regard to actual content in video. Note that in the particular area of news video processing, knowledge about anchorpersons, news headlines, and news icons provides special visual cues. These cues are content-based and genre-specific, basically possessing semantic hints useful in content indexing and abstraction. In the following, some key topics of the related research are briefly discussed.

Shot Boundary Detection

It is not efficient to process a video clip as a whole. So it is beneficial to first decompose the video clip into shots (a shot denotes a single, uninterrupted camera take) and then work at the shot level. Consequently, the video parsing process usually starts with a step for shot boundary detection. Shot segmentation is now a well-known problem for which many solutions based on low-level features are available, such as histogram, pixel and other statistics-based methods, as well as recent work based on clustering and post filtering. In many cases, good results can be obtained (Gao & Tang, 2002; Garqi, Kasturi, & Strayer, 2000; O'Toole, Smeaton, Murphy, & Marlow, 1999).

In this chapter we assume that, prior to the news video indexing and abstracting process, a news sequence has already been segmented into video shots that reasonably serve as elementary units to index the whole video clip.

Anchorperson Detection

As addressed above, physical camera shots are minimum units that constitute the video structure. However, some semantic content units, such as conceptions of news item and news story, are needed for effective news browsing and retrieval. In this regard, high-level, content-based features have to be chosen and employed to enable reliable news item parsing. With the observation of generic news video programs, it can be found that each semantic independent news item is typically composed of an anchorperson shot followed by relevant news footage. In the majority of cases, anchorperson shots simply consist of a static background (usually in the studio) and the foreground person who reads the news. Therefore, detecting anchorperson shots from video clips plays a key role in news item parsing.

Most of the existing anchorperson-detection methods are based on the model-matching strategy (Avrithis, Tsapatsoulis, & Kollias, 2000; Faudemay et al., 1998; Furht et al., 1995; Hanjalic, Lagendijk, & Biemond, 1998, 1999). Some others use shot- clustering approaches (Ariki & Saito, 1996; Gao & Tang, 2002; O'Connor et al., 2001; Qi et al., 2000). In Lu and Zhang (2002), the detection of anchor segments is carried out by text-independent speaker-recognition techniques. Normally, *a priori* knowledge based on periodicity and repetition of the appearances of anchorperson, as well as some heuristics in the context, are exploited to make the detection process easier and faster (Ariki & Saito, 1996; Hanjalic, Langendijk, & Biemond, 1999; O'Connor et al., 2001). As shown by several detection results in those experiments, there are often a few false positives, within which some similar "talking heads" (e.g., anchors, reporters, interviewees, or lecturers) show up but definitely have nothing to do with anchors. They are not easily wiped off in the process for the great similarity, whatever thresholds or parameters are tuned. With these incorrect news item partitions, other components of video indexing become significantly less useful.

In this chapter, we aim to solve this problem in a three-pass approach. First, all kinds of focused "talking heads" in the center of the frame, defined here as the Main Speaker Close-Up (MSC) shots, are detected by motion analysis in the shot. Second, a clustering algorithm based on visual features is applied to construct a table of human groups. Finally, the anchor group is distinguished from others by some specific temporal features related to domain knowledge, the validity of which is proven in statistical experimental data. Since the whole process of detection is divided into small steps from approximation to precision, low complexity is required at each step. In addition, with a clear intention of extracting anchor shots from all the potential ones, it turns out to be more effective. Furthermore, those MSCs collected in the table give clues to an overview of the important persons involved in news programs.

Video OCR

To extract semantic content information from news is a challenging task. One of the most direct ways to achieve this purpose is to extract text (e.g., captions, movie credits, dialogues, etc.) from news video. Since text is an important information source and gives high-level semantic content clues if correctly recognized, it is useful for news story parsing and classification. As widely studied and used, extracting textual information directly from image sequences is often referred to as video OCR. There are mainly two steps in video OCR: text extraction and text recognition. Generally, image enhancement and segmentation methods are first employed to enable text area detection. Then a conventional OCR engine is applied to the detected area and recognized text information. The former step is crucial here and many research works on related problems have been reported. A lot of methods are based on edge detection and multi-frame integration approaches (Qi et al., 2000; Sato et al., 1998). In addition, natural language processing (NLP) technologies are used to perform automated categorization of news stories based on the text obtained. In this case, pre-defined categories are set empirically (Qi et al., 2000) or by statistical analysis from other media, e.g., auditory information (Faudemay et al., 1998).

In this chapter, instead of discussing the capture of all kinds of text appearing in news program generically, we just focus on one of the most familiar and distinctive textual information: news captions emerging at the bottom of frame. With a few spatial and temporal restrictions, a much simplified detection algorithm is designed. The advantage of this caption-detection algorithm is that enough useful information can be extracted conveniently to help modify main speaker clustering and index the human group table.

Video Representation and Abstraction

The target of all the video parsing approaches is to provide users with the ability to retrieve information from broadcast news programs in a semantically meaningful way at different levels, so constructing a hierarchical structure is probably the most effective approach. To generate this hierarchy, data processing in two directions has to be done. One direction is coarse to fine: hierarchically segmenting the given data into smaller retrievable data units; the other is fine to coarse: hierarchically grouping different units into larger yet meaningful categories (Huang et al., 1999). So far, many proposed frameworks (Faudemay, et al., 1998; Furht et al.,1995; Huang et al., 1999; O'Connor et al., 2001; Qi et al., 2000; Rui et al., 1998) are organized with several levels of video units, including terminologies like video shots, video scenes, video groups, video objects, and so on. By using these browsing structures, users not only have a global picture of the main idea, but also can choose to skip unconcerned units and locate video fragments of interest. Usually, the capability to play the video highlight back and forth on selective segments is also provided.

Among all the applications based on a structural representation, video abstraction is an important one that best fulfills users' needs on fast news browsing. Basically, effective browsing of a news program in a short time without losing the major content is a common requirement. An ideal abstraction would display only the video pertaining to a segment's content, suppressing irrelevant data. Examples of multimedia abstractions include short text titles (Qi et al., 2000; Sato et al., 1998), key frames (Zhang et al., 1995; Zhuang, Rui, Huang, & Mehrotra, 1998), key shots (Hanjalic & Zhang, 1999), and some integrated means (DeMenthon, Kobla, & Doermann, 1998; Pfeiffer, Lienhart, Fischer, & Effelsberg, 1996).

Most existing related works try to obtain a concise general expression of video content, in scope of either certain segments or the whole clip. However, our approach in this chapter takes a different perspective. We particularly focus on all the important talking humans captured by camera. Keeping in mind that these "VIPs in video" may be of special interest to and retrieved by users, we propose a hierarchical structure with specific annotation on shots of the VIPs and according video abstraction in the key persons phase.

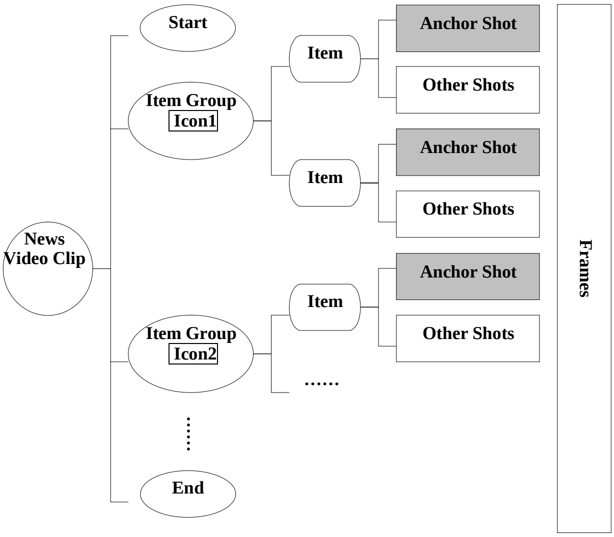
MAIN THRUST OF THE CHAPTER

Hierarchical Structure for News Content Indexing

From the many possible video representations, we chose the tree structure hierarchical representation in our work. Multi-level video units were created and grouped parsing the infrastructure of news clip, as shown in Figure 2. We have rules and assumptions based on specific visual cues of a generic news program to guide our partition and grouping process.

A common video clip is physically made up of frames, and our video structuring begins with identifying individual camera shots from frames. Video shots serve as elementary units to index the whole video clip. For the convenience of focusing attention on the indexing and abstracting applications, we assume that the shot boundary detection step has already been done using some other methods (Gao & Tang, 2002; Garqi et al., 2000; O'Toole et al., 1999). A higher level is composed of news items, which are naturally semantic content units for news video and contain several shots each. Mostly, one single item reports an independent event that happens around one place. Due to the assumption that every news item starts with an anchor shot, we can make the partitions between items by anchor shot detection. Next, the news icon that sometimes appears on the background of anchor frame is found to be a reliable indicator of a news topic, and is thus employed in our structuring. Typically, some content-related news items can be seen in continuous shots led by a common news icon. We call them, as a whole, a news

Figure 2. Tree-structured representation of a typical news video clip

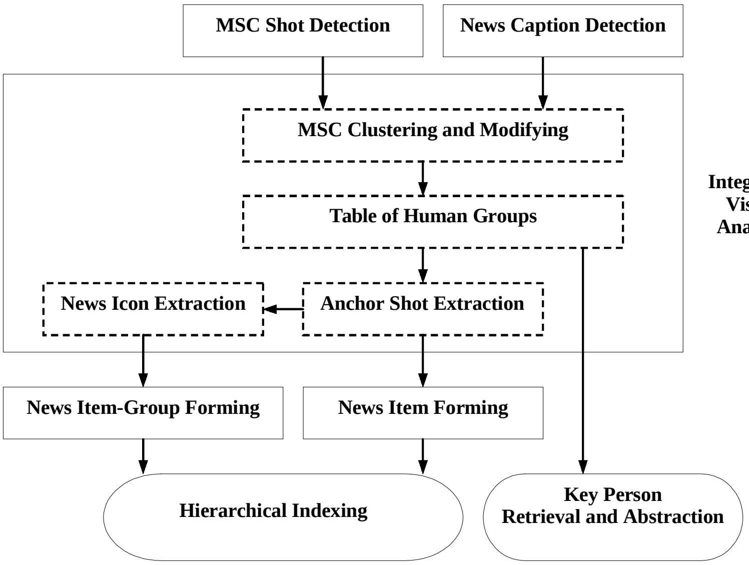


item-group, including consecutive news items in a same field or subject (i.e., focused on one person). News icon extraction and grouping helps to indicate boundaries of item-groups, and the extracted representative image gives a hint of the stories that follow. With the concepts described above, a hierarchical video structure with five levels is built up: frames, shots, news items (by anchor shot), news item-groups (by news icon), and video clip (Figure 2). In addition, the start and end of a news program are out of the main content structure, and they may be separated by the first and last anchor shot.

Tree structure hierarchical representation is commonly found in the literature (Faudemay et al., 1998; Furht et al., 1995; Huang et al., 1999; O'Connor et al., 2001; Qi, 2000; Rui, 1998). However, there exist two major problems in some of the works: First, a few medium-levels between shot-level and news-clip-level are defined in most cases, such as the "scene" or "group of shots" denoting a sequence of shots with similar visual or audio contents. They are normally not semantic groups, but aggregations resulting from grouping or clustering process (O'Connor et al., 2001; Qi et al., 2000), even with some rules considering time restriction, such as time-constrained clustering (Yeung & Yeo, 1996). Second, approaches extracting high-level, genre-specific features such as anchorpersons and news captions are provided (Ariki & Saito, 1996; Faudemay et al., 1998; Gao & Tang, 2002; Hanjalic, Lagendijk, & Biemond, 1999; Huang et al., 1999; Qi et al., 2000; Sato et al., 1998), for the unreliability of low-level features. But time-consuming computation is needed, and they infrequently bring satisfactory results due to the complexity and variance of news sources.

In order to overcome these drawbacks, our proposed structure is completely based on specific semantic cues of news video, such as anchors and news icons. Supported by a number of novel approaches for visual cues extraction, this content-based news structuring process is efficient and computationally easier. In addition, the organization of video units at different levels follows the rule of temporal continuity, with only strictly

Figure 3. News video indexing and abstraction scheme



time-consecutive grouping allowed. With meaningful multi-level content units, video indexing and abstraction is enabled, as shown in the experiments later.

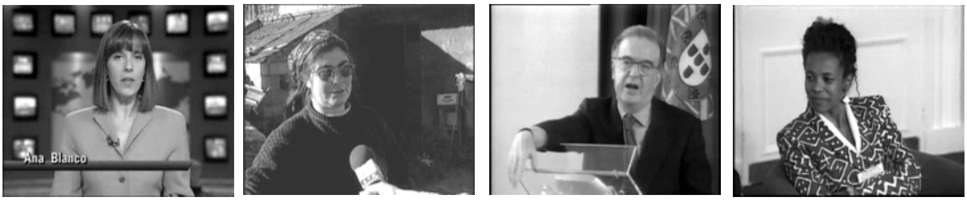
As illustrated in Figure 3, an overall framework of our indexing and abstraction scheme consists of several components. First, approaches for extraction of two visual cues are applied to the video: MSC shot detection and news caption detection. Then follows the main step: integrated visual analysis. In this process, all the MSC shots are automatically clustered and the results are refined according to news captions, building up a table of human groups. Next, with anchor shot and news icon extraction, two levels of our hierarchical structure are formed: news item and news item-group. Finally, the above results are gathered to enable news content indexing and abstraction in the key person phase. The following sections are for the components of this framework.

MAIN SPEAKER CLOSE-UP (MSC) SHOT DETECTION

Definition of MSC

As mentioned above, anchor shots are used as hypothesized starting points for news items, but anchorperson detection is probably interfered by some other kinds of likely talking heads, such as reporters, interviewees, and lecturers (Figure 4). One way to solve this problem is to add a few pre-processing (Albjol, Torres, & Delp, 2002) or post-

Figure 4. Examples of MSC



processing (Avrithis et al., 2000) methods to prevent the possible false alarms. However, in view of employing rather than rejecting the interferences, we can take proper steps to extract visual information from these “false alarms.” The special information gives clues to video-content structuring and facilitates anchor shot identification.

Since a dominant portion of news video is occupied by human activities, human images and especially those camera-focused talking heads play an important role. These heads always include the VIPs of the whole program. Observing a common news clips, we can easily notice addressing heads of presidents or ministers appearing in political news, speaking heads of pop stars or athletes in amusement or sports news, and teaching heads of professors or instructors in scientific news, etc. Definitely, these are key persons who would probably be of interest to users during news video browsing. Therefore, apart from being a good hint for the existence of anchorpersons, all the focused heads themselves serve as a good abstraction of news content in the key persons phase. For convenience, we might as well call the shot with such an impressive head the Main Speaker Close-Up (MSC) shot.

With regard to that consideration, our approach starts with detecting all the MSC shots in the video stream. Then, possible semantic analysis like person clustering and classification is made based on different visual features. After that, an outline table of main speakers is organized. Finally, anchor shots are distinguished from other MSCs using temporal distribution features. Actually, other groups in the table are employed later to facilitate human close-up indexing and retrieval.

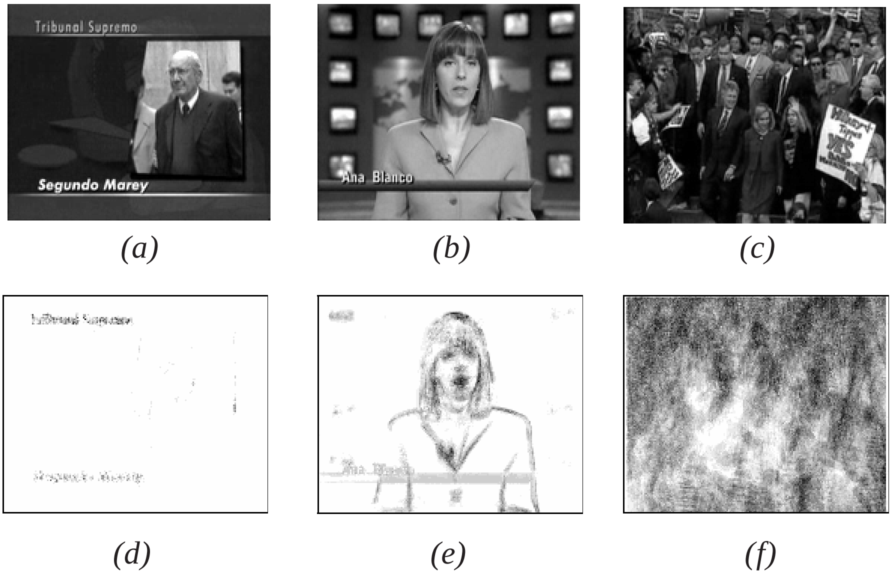
MSC Detection by MAMS Analysis and Head-Motion-Model Fitting

Identifying significant objects, such as human faces, text, or automobiles that appear in the video frames, is one of the key components for video characterization. The detection of a human subject is particularly important here in order to find MSC shots. As it is defined, MSC denotes a special scene with a single, focused talking head in center of frame (Figure 4), identification of which is possible by using a number of human face-detection algorithms. Generally, approaches of face detection in video are quite complicated, with some parameters needing to be manually tuned. Given that face detection in a still image is already difficult enough, face detection in video is too time-consuming for practical application. For example, in Wang and Chang (1997), Avrithis et al. (2000), and Tsapatsoulis, Avrithis, and Kollias, 2000), the typical techniques proposed involve color segmentation, skin color matching, and shape processing. In order to avoid these conventional complex steps, we propose a novel detection method based on Head-Motion-Model fitting.

In fact, we are not quite concerned about face details such as whether it has clear outline of cheeks, noses, or chins. What gives the most important clues at this point is whether or not it is a focused talking head in front of camera with the potential to be an anchorperson, rather than any meaningless object or background. In terms of this viewpoint, skin color and face shape are not the necessary features that must be extracted. Thus, differing from other methods based on complicated face color/shape, model training, and matching (Avrithis et al., 2000; Hanjalic et al., 1998), we manage to identify MSCs in a video stream by motion analysis in a shot, with a simplified Head-Motion-Model set. Before we begin to discuss the detail detection algorithm, some spatial and motional features of the MSC concept defined here have been gathered and illustrated below:

1. Basically, an MSC shot [Figure 5(b)] has relatively lower activity than common shots [Figure 5(c)] in a news video stream, but has stronger activity than stationary scene shots [Figure 5(a)];
2. Each MSC has a static camera perspective, and activity concentrates on a single dominant talking head in the center of frame, with a size and position within a specific range;

Figure 5. Comparison of shot activity between MSC shot and others



(a) (b) (c)

(d) (e) (f)

$$\begin{array}{ccccccc} \frac{MAMS}{x \cdot y} & : & 0.0022\,(10^{-3}) & < & 0.0384\,(10^{-2}) & < & 0.3473\,(10^{-1}) \\ & & (T_S = 10^{-3}) & & & & (TC = 10^{-1}) \end{array}$$

3. In MSC shots, the presence of a talking head is restricted in a fixed horizontal area during one shot, with three typical positions: Middle (M), Right (R), and Left (L), as shown in Figure 6(a) to (d).

Since we have defined MSC so broadly here with wide variations of size, color, and shape, detection algorithms based on such features do not work well. Instead, the spatial and motion features addressed above are used in our approach. According to the features (i) to (iii) mentioned above, motion analysis in a shot is sure to grasp the key points. In fact, we need a metric that not only measures frame dissimilarity [used for (i)], but also reflects the motion distribution in a shot [used for (ii)], Therefore, the popular image difference measurement — the mean absolute frame difference (MAFD — and an alternative histogram difference measurement by three color components (Furht et al., 1995) will not satisfy our needs. Innovatively, another shot activity measurement is proposed here, namely, the map of average motion in shot (MAMS). The MAMS of shot n is defined as:

$$MAMS_n(x,y) = \frac{1}{M}\sum_{\substack{i=1 \\ i=i+v}}^{M} \left| f_i(x,y) - f_{i+v}(x,y) \right| \qquad M = \frac{N}{v} \tag{1}$$

where N is the total number of frames in shot n, i is the frame index, v is the interval between two computed frames, and (x, y) are the spatial coordinates of the frame.

Actually, MAMS is a map that displays the intensity of activity in a shot and the distribution. This map is the same size as the original frame. However, we can also reduce our computation and storage load by just resizing original frames to a smaller dimension before calculation, of course with less precision. In our experiments, a compromising dimension of a half-size map is used and is illustrated in Figure 5(d) to (f) and Figure 6(d) to (f) (original frames also resized to give an equal look).

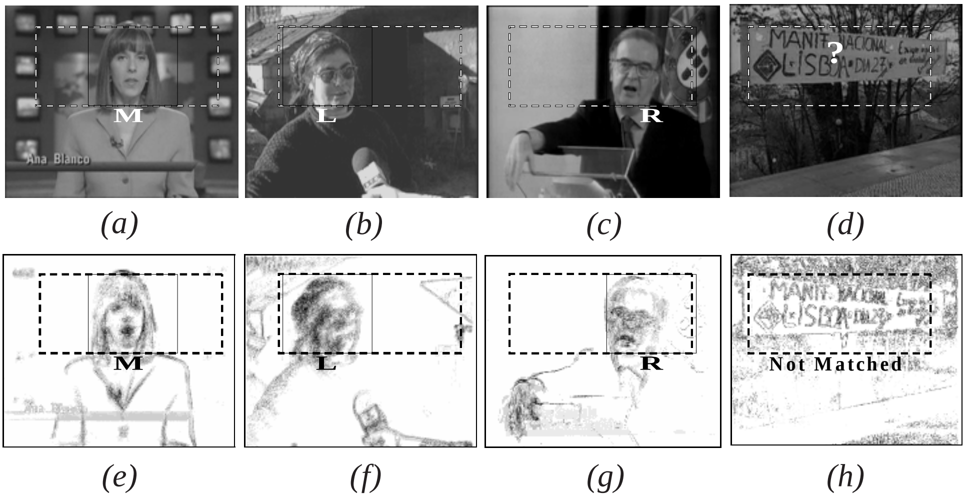
MAMS supports motion analysis both to the global screen and to some local parts. A measure similar to MAFD is computed by the average activity of pixel in shot, which is simply MAMS divided by the dimensions. Then, shot classification is made by this measurement, as shot activity of MSCs should be neither too low nor too high according to (i). Only those shots satisfying the condition:

$$Ts < \frac{MAMS_n}{x \cdot y} < Tc \tag{2}$$

are regarded as potential MSCs and move on to the following steps, where Ts and Tc stand for typical values in static scenes and common shots respectively (examples are shown in Figure 5).

After that, a Head-motion-model fitting step is taken based on four fixed boxes: Head Band (dashed squares in Figure 6), Middle Box, Right Box, Left Box (solid line squares in Figure 6), the position and size of which can be estimated by training or just empirical setting. Since activity in MAMS has a distinct concentration on the head of main speaker

Figure 6. Head-motion-model fitting based on MAMS



in MSC according to (ii), with nearly "blank area" in the remaining area of the Head Band, criteria can be set as:

$$\frac{\max\limits_{(L/M/R)}\left\{\sum\limits_{(x,y)\in(L/M/R)Box} MAMS_n(x,y)\right\}}{\sum\limits_{(x,y)\in HeadBand} MAMS_n(x,y)} > T_H \qquad \textbf{(3)}$$

which means that a ratio of MAMS in Boxes L, M, R to that in Head Band should be higher than a threshold T_H (it is empirically set to about 0.7 in our experiments). Before this judgment, the morphological open operation is also employed to MAMS to suppress possible noise. In this way, MSC shot can be determined with a dominant moving head locating remarkably in one of the three boxes at this height (Table 1).

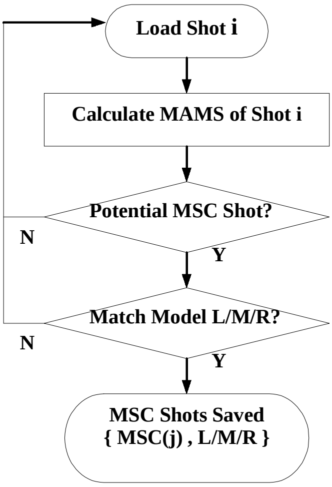
To sum up, our MSC-detection module of the overall framework (Figure 3) contains a two-pass filter scheme, as shown in Figure 7. A key concept throughout the process is the novel shot activity representation — MAMS. Unlike other researchers taking key frames as features of frame sequences (Faudemay et al., 1998; Gao & Tang, 2002; Hanjalic & Zhang, 1999), we use this "motion map" to preserve more temporal information, including activity intensity in all (as MAFD) and distribution in parts. With these advantages, our MSC-detection method is not quite dependent on scale and without relying heavily on lighting conditions, limited occlusion, and limited facial rotation. On the other hand, use of a fixed general-motion model has its drawbacks — less precision and sensitivity to noise, resulting in a few false alarms being found in experiments. In

Table 1. Head-Motion-Model fitting results of Figure 6

	(I)	(II)	(III)	(IV)
M	0.99	0.76	0.83	0.52
L	0.31	0.91	0.02	0.48
R	0.57	0.14	0.98	0.59
T_H (0.7)	M	L	R	None

Max {L, M, R} is marked in the shaded cells.

Figure 7. MSC shot-detection scheme



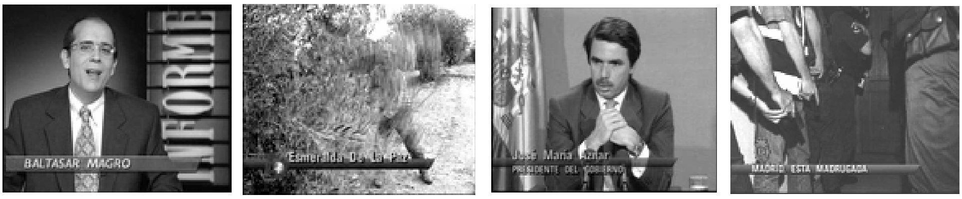
order to acquire better results, another visual cue, news caption, is taken into consideration as follows.

NEWS CAPTION DETECTION

The distinct textual information of a news program is the headline or caption appearing in video, as illustrated in Figure 8. According to the content and the function of the caption texts, most of them can be classified into two scenarios: annotation of a news story and name of a camera-focused person (e.g., anchorpersons, reporters, interviewees, etc.). Detection of both of them is an important way to help understand news contents and enable content-based parsing. In our proposed scheme, the news captions in the latter scenario play a more useful part, as they are related to MSCs that have already been extracted. The MSC clustering results can be modified and then indexed, based on the captions of person names.

Differing from normal printed text, news caption text has relatively low resolution and is often found on a complex background. It is difficult to get good detection results only using textual features, as found by Li, Doermann, and Kia (2000), Sato, Kanade, Hughes, Smith, and Satoh (1999), and Smith and Kanade (1997). In contrast, we use image

Figure 8. Examples of news captions



analysis technology to detect the frames that contain a "band-area" of captions, and then video OCR technology can be applied to these areas to extract text. In our process, spatial and temporal features rather than textual features are employed. Through a great deal of experimental analysis, we determined that the captions appearing in news program have three distinct features:

1. The appearances of captions are generally abrupt text events (i.e., resulting in prominent difference between frames with and without captions), with a rectangle band-area changing significantly compared to background;
2. News captions are located spatially in a fixed region, generally at the bottom 1/4 field of screen;
3. News captions are located temporally in the starting L frames of a video shot (L < 100) in most cases.

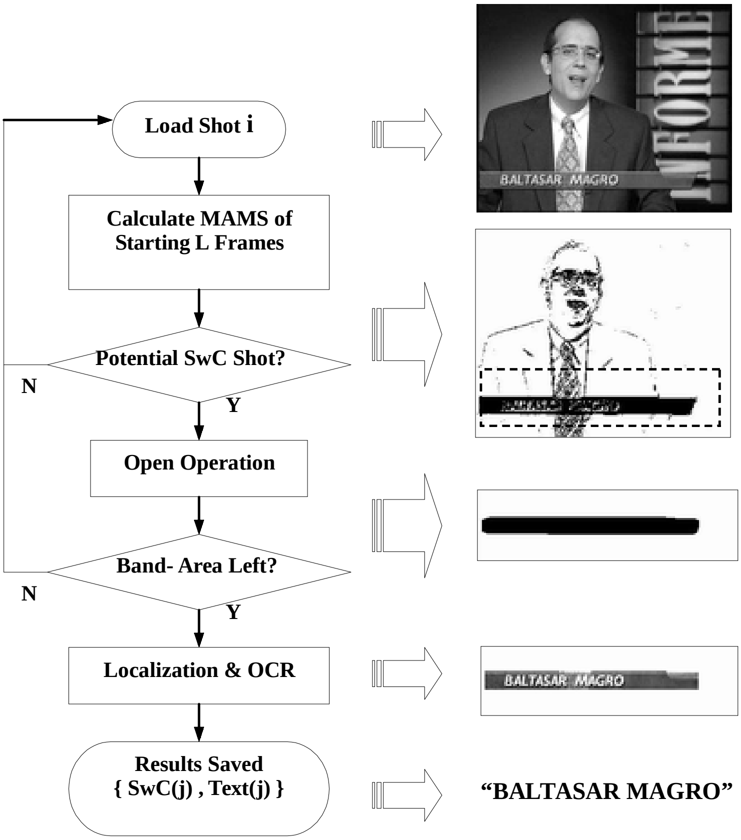
In order to detect frames containing the specific captions, we use the specific-region text-detecting algorithm. As shown in Figure 9, it mainly consists of three steps. First, MAMS is evaluated for the starting L (let L = 80 in experiment) frames of each shot. We can decide if it has the potential to be the Shot with Caption (SwC) by just counting the summation at bottom 1/4 field of screen. The SwC candidates should satisfy:

$$\sum_{(x,y)\in Bottom\frac{1}{4}Area}^{i<L} MAMS_n(x,y) > T_{Cap} \qquad \textbf{(4)}$$

where Tcap is an appropriate threshold selected by experiments, estimating the normal activity of abrupt text events. Second, considering that the scattering noise may be incorrectly taken as text events, we perform the morphological open operation, as it is used in MSC detection. Note that a bigger rectangular template is applied here to find rectangular band-area of similar dimension. After this operation, we get the caption area and pass it to the OCR module. Finally, all the SwCs and the corresponding caption texts are saved.

This is a much simplified detection algorithm compared to those used by Li et al. (2000), Sato et al. (1999), and Smith and Kanade (1997). It is not dependent on the clearness of text appearing in the video stream or the ability of text recognition. As long as a caption area emerges with the defined spatial and temporal features, it is captured

Figure 9. News caption detection: specific-region text-detecting algorithm



and located. This approach has robustness, especially in the case of gradual transition and heavy noises on the background.

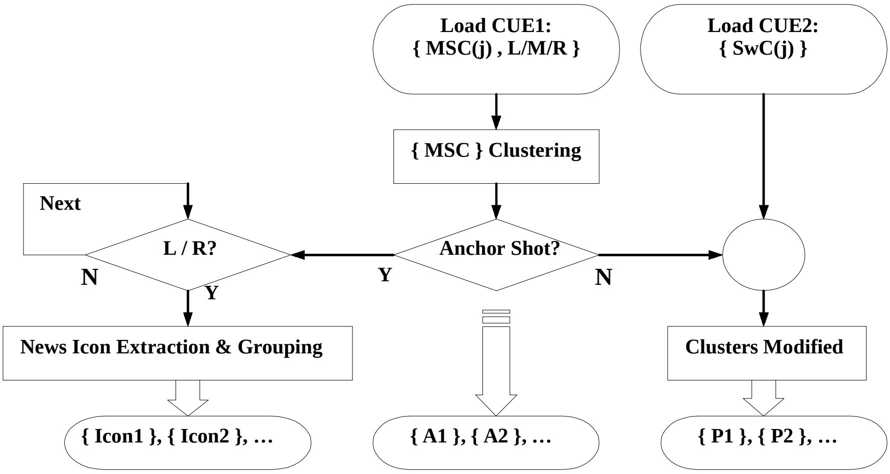
INTEGRATED ANALYSIS USING SPECIFIC VISUAL CUES

Based on the two extracted visual cues, MSC and news caption, as illustrated in the overall framework (Figure 3), an integrated analysis scheme is designed, with all steps shown in Figure 10. The following sections will describe them in detail.

MSC Clustering and Results Modified

As addressed before, the extracted MSCs serve as good abstraction of news content in the key persons phase. Actually, it is not effective for a user to browse all the MSCs one by one to search for somebody in particular. We need some MSC navigation

Figure 10. Integrated visual analysis scheme



and indexing schemes. Since the same "main speaker" usually has similar appearances (like clothes' color and body size) in many shots during one news program, hopefully human clustering is done to build up an index table. With the guidance of this table, the retrieval of key persons is supported in an immediate way.

Our MSC clustering algorithm uses color histogram intersection (CHI) as similarity metrics. Remembering that we had MSC-detection results before, the acquired head positions (L/M/R) actually suggest three corresponding body areas, which can be also called Left / Middle / Right body positions. Simply, we just set up three Body-Area-Models that have the shapes shown in Figure 11. The three bodies have equal size, and only pixels in these particular areas are included in CHI calculating.

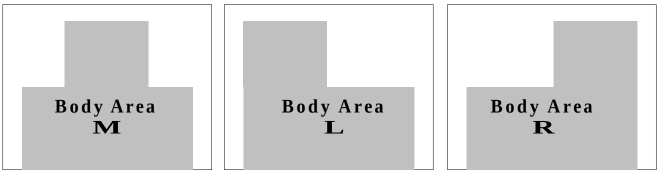
As each MSC shot would contains many frames, the CHI operation is applied on an average image of the whole shot. The similarity metric applied is

$$Sim\left\{\underset{L/M/R}{MSC1}, \underset{L/M/R}{MSC2}\right\} = \frac{\sum_{h=1}^{H} \min\left\{\overline{MSC1}_h(Y,U,V), \overline{MSC2}_h(Y,U,V)\right\}}{\sum_{h=1}^{H} \overline{MSC1}_h(Y,U,V)}, \qquad \textbf{(5)}$$

with

$$\overline{MSC} = \frac{1}{N}\sum_{i=1}^{N}\sum_{(x,y)\in L/M/R} f_i(x,y)(Y,U,V), \qquad \textbf{(6)}$$

Figure 11. Three Body-Area-Models (L/M/R)



where $\overline{MSC1}$ and $\overline{MSC2}$ are average images of two MSC shots, computed with pixels only in corresponding body areas, and $\overline{MSC}_h(Y,U,V)$ is the *h*-th bin of the color histogram (Y, U, V color space is used here). Once the similarity $Sim\{MSC1, MSC2\} \in [0,1]$ is calculated between every two MSC shots, clustering can be done by several common techniques (Jain & Dubes, 1988). The following scheme is just a simplified one, since the actual number of clusters is not known *a priori*. Any two MSC shots, MSC1 and MSC2, are clustered together whenever

$$Sim\{MSC1, MSC2\} > Tg \qquad \textbf{(7)}$$

where Tg is adjusted to about 0.8 in experiment. In this way, MSC shots are clustered into several groups, with each one containing similar human appearances.

After all, this cluster method is a quite simple one. As we do not let Tg be a strict value quite close to 1, close-ups of different persons are probably grouped together inappropriately. In order to construct an outline table of main speakers with a different person in each group, we have to further refine the clustering results. Thus, modifications are made based on the other extracted visual cue, the news caption. Normally, the news caption of one speaker's name (mentioned in section of "News Caption Detection") appears in one of the MSC shots, and often the first time a speaker shows up. With this rule, a three-step scheme is designed to modify MSC clustering results, as below:

Step1. Intersection

Intersect MSC shots {MSC} by Shots with Captions {SwC} to find MSC Shots with Captions {MSCwC}:

$$\{MSCwC\} = \{MSC\} \cap \{SwC\} \qquad \textbf{(8)}$$

Step2. Splitting

Split those groups with different captions (different names) in, and cluster again MSCs within one group (let Tg be a lower value this time);

Step3. Indexing and Tabling

After Step 2, refined groups are regarded as human groups containing close-ups of one key person each. Meanwhile, each group is indexed textually by a person's name (caption text), and visually by its representative close-up shot (MSCwC).

Up to now, all gathered MSC groups could be organized into an outline table of humans to facilitate key person indexing and retrieval, as illustrated later in video abstraction application.

Anchor Shot Identification and News Icon Extraction

In order to identify news item boundaries, we still have to separate anchor groups from other ones in this human table. Proper criteria are needed at this point. Some of the literature (Ariki & Saito, 1996; Faudemay et al., 1998; Gao & Tang, 2002; O'Connor et al., 2001; Qi et al., 2000) follows a certain medium-specific prior knowledge, such as the duration, proportion, distribution, and frequency of typical anchor shots, to filter out non-anchor shots from all the candidates detected in the video stream. However, the thresholds used there are usually empirically chosen. In our approach, we have investigated a large amount of video data about four potential features: shot length, number of shots, shot interval, and time range of all shots, with average statistics collected in Table 2. (The data acquired is based on more than six hours of news programs from MPEG-7 Video Content Set.)

As shown in Table 2, the last three features, which are most distinct ones, are used to distinguish anchor shots from other MSCs: more times of repetition, more disperse in time, and totally longer range of time. The anchor group identification approach is a greedy algorithm that loops through all the MSC groups and selects those that satisfy the following conditions:

- Total number of shots in the group is more than a predefined value (e.g., 5 shots per 20 minutes);
- The average shot interval should be more than a predefined value (e.g., 10 shots);
- The range of shots of the group (i.e., the distance between the first and the last shot in terms of number of shots) is higher than a predefined value (e.g., 50 shots).

Note that the thresholds used in this algorithm are chosen based on Table 2.

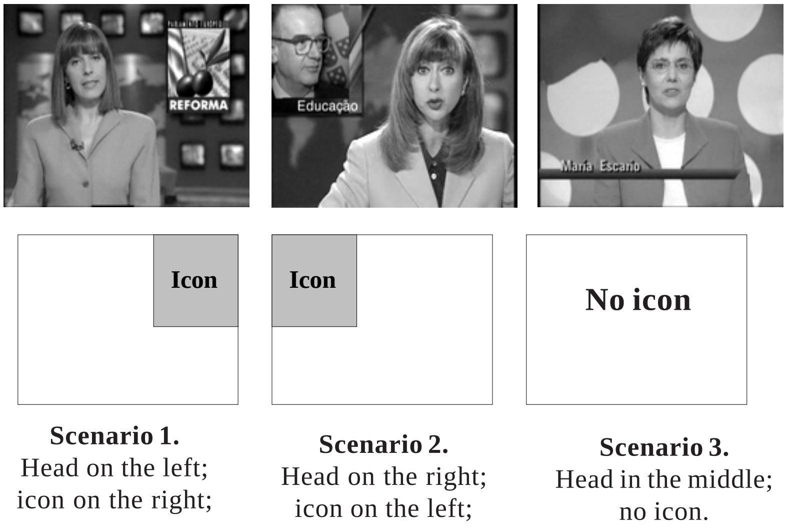
By this time, we have classified all MSC groups in table into two types: anchor groups ({A1}, {A2}, …) and other person groups ({P1}, {P2}, …) (Figure 10). Then, anchor shots are used to make partitions of news items, each anchor leading one item. The start and end of the news is segmented out by the first and last anchor shots. Therefore, the item-level parsing of news video is formed.

Another important visual hint is the news icon, or the topic box, that is a picture or symbol in the upper corner of the screen, mostly showing up in the anchor shot and describing a current or following topic. These news icons are found to be reliable

Table 2. Average statistics of anchor shots in contrast to other MSC shots

	Shot length	# of Shots (every 20 minutes)	Shot interval	Range of all shots
Anchor group	15.72sec	≈ 9 shots	≈ 12 shots	>60 shots
Any other MSC group	10.38sec	≈ 2 shots	≈ 5 shots	<10 shots

Figure 12. Possible anchor head positions and news icon extraction



Scenario 1.
Head on the left;
icon on the right;

Scenario 2.
Head on the right;
icon on the left;

Scenario 3.
Head in the middle;
no icon.

indicators of news content, and a common icon keeps its presence in anchor screens of continuous news items referring to a same field. Thus, consecutive news items can automatically be grouped into different topics (a new different icon coming out denotes a topic change), forming a higher level of semantic unit, the news item-group. In addition, news icons are extracted as visual annotation of the corresponding item-group. According to different anchor head positions, there are three possible scenarios, as shown in Figure 12.

We can empirically set the dimensions of the two news icon boxes and keep corresponding sub-images in the first two scenarios. Next, we calculate the visual similarity between consecutive news icons by the CHI metric in Equation (5) (of course, all pixels in the sub image are included). Two adjacent news items with icons are grouped together when

$$Sim\{Icon1, Icon2\} > Ti \qquad \textbf{(9)}$$

In contrast to Tg used in MSC clustering, here Ti is set to a strict value quite close to 1 (i.e., 0.95), for only the same news icon means the same news item-group. In particular, a news item without an icon (e.g., the leading anchorperson located in the middle of the screen) is regarded as an independent news item-group itself. Therefore, the item-group level parsing of the news video (Figure 2) is formed and illustrated by different news icon images.

VIDEO REPRESENTATION AND ABSTRACTION BY SELECTIVE MSCS

Once different semantic units are extracted, they are aligned in time with the media data so that proper representations of these events can be constructed in such a way that the visual presentation can convey the semantics efficiently. A commonly used presentation for semantic structure is the form of a table of contents (Rui, Huang, & Mehrotra, 1999). Since it is a concept that is familiar to most users, we adopted it in our representation. Additionally, in order to give users a sense of time, we also use a streamline representation in a metric of shot number. As can be seen in Figure 13, this proposed solution exploits essential characteristics in data itself at different levels.

Our presentation for the semantic structure of a news program can be visibly divided into two main parts, each serving a different function. On the left of the framework lies a news content index in a familiar hierarchical fashion, with news items aligned in time order from top to bottom and belonging to different news item-groups. In addition, news icons are provided as "thumbnail images," simultaneously appearing next to corresponding groups. To play back a particular item or item-group, a user simply clicks on the button of the desired segment in this hierarchical table. On the right of this interface is the streamline representation, where a time line (in shot) runs from left to right according to each item listed on the left. The most characteristic components of this line are those MSC shot positions that are pointed out, with types denoted by A (anchor) or P (any other person), and serial number denoted by 1, 2, 3 …. More visually illustrated, a graphic table of person group according to each MSC cluster is presented in Figure 14, with different names possibly extracted from captions annotating every person. A user can see

Figure 13. News video representation

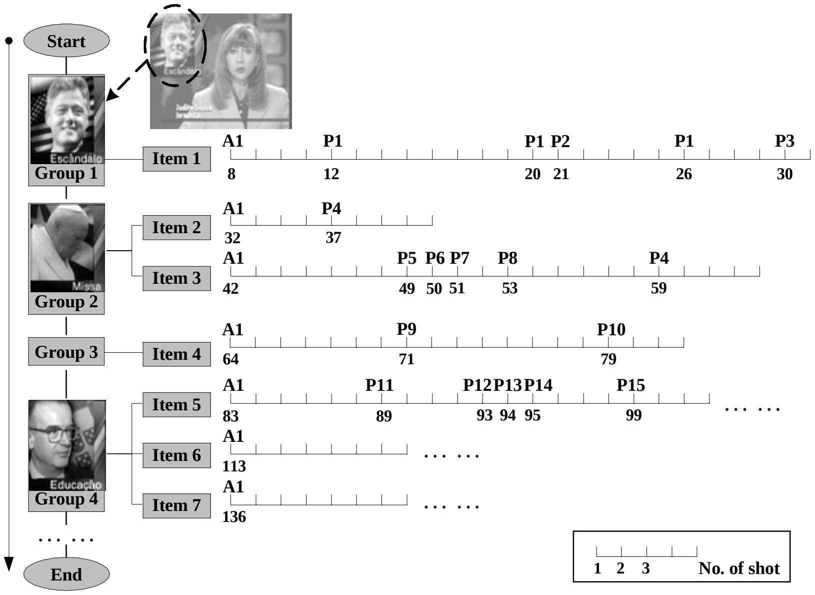
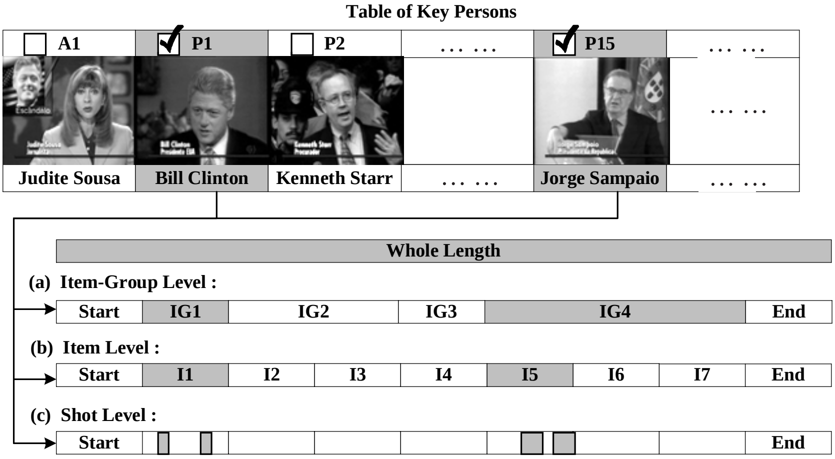


Figure 14. Multi-level video abstraction by selective MSCs



immediately who are the main speakers showing up in a certain news item or item-group, by just getting a glimpse of the MSC positions on the time line. Moreover, with a little effort referring to the corresponding person icons in table, a main idea is formed about who they are and whether some of them deserve attention. Then locating and playing back the particular MSC shots becomes as easy as picking up a known book from an orderly shelf. In general, compared with linear browsing or low-level scene cut browsing, our system allows a more effective non-linear access and a more direct content-based retrieval in the key persons phase.

This video structuring scheme also supports content-based abstraction. As indicated before, our abstraction strategy takes a different perspective from generating a normal overview of the video content. It is a kind of MSC-based abstraction with emphasis on key persons in the news program. Basically, we need to tackle queries from users such as how to find all the VIPs in a program or how to highlight specific scenes with particular persons. Usually, it is hard to answer these requests for it is not known where these persons are located in the video steam. However, in our approach, a table of such key persons has already been built, thus it naturally meets such needs. In this chapter, a hierarchical abstraction is proposed at three levels (from a low-physical level to a high-semantic level): shot level, item level, and item-group level. As illustrated in Figure 14, in this example, assuming a user is interested in two persons appearing in the news program: Bill Clinton and Jorge Sampaio. He may try to find both of them from the human table by name or representative images and check the corresponding boxes. Then, multi-level abstractions can be acquired: abstraction at shot level contains all the MSC shots of these two persons; abstraction at item level is made up of news items including those MSC shots; and abstraction at item-group level extends to every news item-group with news items contained at the item level. To obtain a summarizing preview sequence, frames belonging to all the video segments (at shot/item/item-group level) are concatenated together. Obviously, abstractions at three levels with different durations can adaptively fulfill users' needs, from independent close-ups to complete video clips. From this example, the effectiveness of key person abstraction for news programs is evident.

EXPERIMENTAL RESULTS AND EVALUATION

The methodology described above has been applied to three news video clips chosen from the MPEG-7 video content set for testing (Frame Rate = 25 frames/second, Frame Size = 288×352 pixel). The detailed information of these news videos is summarized in Table 3.

To evaluate the performance of the proposed scheme of MSC and anchor shot detection, we use the standard precision and recall criteria, shown in the following:

$$\text{precision} = \frac{\text{number of hits}}{\text{number of hits} + \text{number of false alarms}}$$

$$\text{recall} = \frac{\text{number of hits}}{\text{number of hits} + \text{number of misses}}$$

MSC Shot Detection Experiment

In the beginning, we evaluated the performance of the MSC shot-detection algorithm. This is the process of extracting the first specific visual cue. Table 4 shows the output of the proposed two steps in the scheme for the three news clips. For the total 337 video shots, there are 89 MSC shots (identified manually). In the first step, 134 potential MSC shots are detected by MAMS analysis, and then 38 candidates are excluded by Head-Motion-Model fitting. Overall, the proposed method detects 96 MSC shots, including 85 real ones and 11 false alarms. Thus, the precision is 88.54%, and the recall is 95.51%. The fairly high recall ensures the completeness of all the MSC shots, and

Table 3. Detailed information of experimental data

News video	Duration	Video shots	Source
Clip 1	16:35	91	Spanish TV, RTVE
Clip 2	13:32	90	Spanish TV, RTVE
Clip 3	17:58	156	Portuguese TV, RTP

Table 4. MSC shot detection results

News video	# of MSC shots	Candidates detected by MAMS analysis	Excluded by Head-Motion-Model fitting	Final results		
				Hits	Misses	False alarms
Clip 1	22	31	6	22	0	3
Clip 2	24	35	10	23	1	2
Clip 3	43	68	22	40	3	6

the precision also seems high enough for the later classification and modification process.

News Caption Detection Experiment

Next, we evaluate the performance of the news caption-detection algorithm. This is the process of extracting the second specific visual cue. Table 5 shows the output of the first two steps in the proposed scheme. We assume that the last step of character recognition is best accomplished by a typical OCR engine. There are a total of 72 shots with captions among the three news video clips. In the first step, 107 potential MSC shots are detected by MAMS analysis, and then 29 candidates are excluded by Band-Area detection (using morphological operation). Overall, the proposed method detects 78 MSC shots, including 69 real ones and nine false alarms. Thus, the precision is 88.46%, and the recall is 95.83%. Like the results achieved in MSC shot detection, nearly all the news captions are included as suggested by the high recall, and the precision is further improved by following intersection operation {SwC})”{MSC} (most of the false alarms are removed). This is the great advantage of integrating two different visual cues in our video parsing strategy.

MSC Shot Clustering and Anchor Shot Identification Experiments

The experimental results of MSC shot clustering are given in Table 6. First, based on the CHI metric restricted to three body areas (Equation (5)), the 89 MSC shots are clustered into 52 groups. Then, some clusters are further split into smaller ones using 72 shots with caption. After that, we come to the final result of about 92.45% recall with 81.67% precision. A table of human close-ups is constructed including most of the camera-focused speakers. However, a few false talking heads are combined in it, which is mainly caused by a fixed Head-Motion-Model because the MSC detection process fails to filter out some objects with likely head shapes in a motionless background. In spite of this deficiency, most of the false groups are without news captions and can be aligned at the end of human table. As a result, users are rarely confused or misled when browsing

Table 5. News caption detection results

News video	# of Shots with Caption	Candidates detected by MAMS analysis	Excluded by Band-Area detection	Final results		
				Hits	Misses	False alarms
Clip 1	21	33	10	20	1	3
Clip 2	19	28	8	19	0	1
Clip 3	32	46	11	30	2	5

Table 6. MSC shot clustering results

News video	# of persons in MSC	CHI clustering results {MSC}	Corrected by {SwC}	Final results		
				Hits	Misses	False alarms
Clip 1	9 (with 1 anchor)	9	1	9	0	1
Clip 2	15 (with 2 anchors)	16	2	14	1	4
Clip 3	29 (with 1 anchor)	27	5	26	3	6

Table 7. Anchor shot identification results

News video	# of anchor persons	# of anchor shots	Anchor identification results			L/M/R results
			Hits	Misses	False alarms	Errors
Clip 1	1	9	9	0	0	0
Clip 2	2	8	8	1	0	1
Clip 3	1	8	8	0	0	0

the human table (frequently by person names from start to end) to determine the key persons in the new video.

Additionally, anchor group identification results are shown in Table 7. On the basis of fine-grouped MSC shots, anchors are easily extracted by the three conditions listed before. A high precision of 100% and recall of 96.15% are achieved for the three news clips. Note that we have nearly done a perfect job with only one miss. This missing anchor shot is in another scenario, where two different anchorpersons appeared simultaneously in the screen. Since our MSC detection approach only concerns a single, dominant talking head in the frame, this two-person scenario was beyond its capability. It was expected to have more misses when doing experiments in such scenarios. However, compared to other existing anchorperson detection algorithms (Avrithis et al., 2000; Gao & Tang, 2002; Hanjalic et al., 1998; Hanjalic, Langendijk, & Biemond, 1999), our approach not only gives better performance in the one-anchor scenario, but is also computationally easier and requires little human intervention. Finally, the head position (Left / Middle / Right) is decided (with only one error according to the missed anchor shot) to make news icon extraction.

MSC-Based Video Indexing and Abstraction Experiment

Finally, based on the two specific visual cues extracted above, we can perform video parsing and indexing in the hierarchical structure of Figure 2. An example is given in Figure 13. The detailed statistics are shown in Table 8 for the three news clips. Note that all news icons in consecutive news items in Clip1 and Clip2 are different from each other, thus the same results are acquired at the news-item level and news item-group level in these two clips.

In addition, in order to test the video abstraction strategy presented in this chapter, we applied it to the third clip of our experimental data. As shown in Table 9, abstractions demanded in two situations are tested: abstraction with all MSCs, and abstraction with selective MSCs. The former one is a good global picture of all key persons showing up in the news program, and the latter one better fits the requirement of selective persons' highlighting. Hierarchical video abstraction at three levels provides enough options for different length. Table 9 shows details of these multi-level abstractions in the number of video units and in time duration (within brackets).

Table 8. News video parsing and structuring results

News video	Shot Level	Item Level	Item-Group Level
Clip 1	91 shots	8 items	8 item-groups
Clip 2	90 shots	7 items	7 item-groups
Clip 3	156 shots	7 items	4 item-groups

Table 9. MSC-based three-level abstraction results

	# of MSC groups	Shot Level	Item Level	Item-Group Level
Clip 3*	32	156 shots	7 items	4 item-groups
Abstraction with all MSCs	32	44 shots (07:21)	7 items (17:58)	4 item-groups (17:58)
Abstraction with selective MSCs	2	8 shots (00:35)	2 items (06:09)	2 item-groups (11:13)

** Whole length of Clip3 = (17:58)*

FUTURE TRENDS

The procedure and techniques presented in the above could be further improved and pursued in a number of promising directions:

1. The term Main Speaker Close-Up (MSC) defined in this chapter is actually a new concept, which has led to new applications on anchorperson detection and key person abstraction. However, its vital drawback is the limitation of only one dominant head appearing in screen, which excludes another important scenario: two speakers appearing simultaneously on both sides of the screen (e.g., two anchors, both interviewer and interviewee). To solve this problem, we need to broaden the MSC definition to a two-person version. Perhaps a more complicated head motion model is required, and more efforts are needed to tackle noises.
2. The MSC detection method fails to filter out some objects with likely head shapes in motionless background. This deficiency caused a few false alarms in the final results. How to adaptively set up head motion model and to use some possible post-processes should be studied in future.
3. We did not address the commercial breaks and the analysis of introductory or summary parts of the news video, since some of these additional parts are studied by other existing work (Hauptmann & Witbrock, 1998; Taniguchi, Akutsu, Tonomura, & Hamada, 1995). Nevertheless, these techniques could be fused into our later system to improve the robustness of parsing generic news programs.
4. Since the proposed scheme depends on a simply temporal structural model of news video, it has some restrictions. For instance, it cannot identify a change of news items within a single anchor shot sequence. In fact, it is quite difficult to overcome such a drawback using visual information alone. In situations where the simple temporal structure is not followed, speech signals might be combined for news item segmentation to develop a robust news-parsing system.

CONCLUSIONS

In summary, this chapter first reviewed related research works in content-based news video analysis, representation, and retrieval. Then, it proposed a hierarchical framework for news-video indexing and abstraction by two specific visual cues: the MSC shot and news caption. The first visual cue, Main Speaker Close-Up (MSC), is defined as camera-focused talking heads in center of the screen. All the MSCs are detected and clustered into groups. The second visual cue is the news caption appearing at the bottom of screen, probably annotating the current speakers' names. They are extracted to modify the MSC clustering. With these two visual cues, an outline table of MSCs is built up, each MSC group containing close-ups of one key person is indexed by its name if possible. Then, a number of criteria based on temporal features of the anchor shot are used to identify anchor groups from other MSC groups. In addition, news icons are extracted from anchor shots, if present. After that, the news data is segmented into multiple layers to meet different needs of indexing and retrieval: frames, shots, news items (by anchor shot), news item-groups (by news icon), and video clip. Finally, a unique MSC-based video abstraction method is proposed with the focus on key persons. In contrast to providing an overview of the video content in general cases, this abstraction especially meets the user's needs of human preview and retrieval. Experimental results show that the proposed scheme attains better performance than some similar research works and can significantly improves the effectiveness in retrieval. The most important advantage is the table of key persons that enhances the quality of video representation and gives a unique content abstraction in the human phase.

Further research directions in broadening the MSC definition to include a two-person scenario, in improving the robustness of MSC detection, in analyzing the commercial breaks, introductory or summary parts, and in using audio information to overcome drawbacks of visual analysis are indicated.

ACKNOWLEDGMENT

This work has been supported by the Grants NNSF-60172025 and TH-EE9906.

REFERENCES

Albiol, A., Torres, L., & Delp, E.J. (2002). Video preprocessing for audiovisual indexing. *Proceedings of ICASSP*, Vol.4, 3636-3639.

Ariki, Y., & Saito, Y. (1996). Extraction of TV news articles based on scene cut detection using DCT clustering. *Proceedings of ICIP*, Vol.3, 847-850.

Avrithis, Y., Tsapatsoulis, N., & Kollias, S. (2000). Broadcast news parsing using visual cues: A robust face detection approach. *Proceedings of IEEE International Conference on Multi-Media and Expo*, 1469-1472.

Boykin, S., & Merlino, A. (1999). Improving broadcast news segmentation processing. *Proceedings of International Conference on Multimedia Computing and Systems Proceedings*, Vol.1, 744-749.

DeMenthon, D., Kobla, V., & Doermann, D. (1998). Video summarization by curve simplification. *Proceedings of ACM Multimedia 98*, 211-18.

Faudemay, P., Durand, G., Seyrat, C., & Tondre, N. (1998). Indexing and retrieval of multimedia objects at different levels of granularity. *Proceedings of SPIE*, Vol. 3527, 112-121.

Furht, B., Smoliar, S.W., & Zhang, H.J. (1995). *Video and image processing in multimedia systems*. Norwell, MA: Kluwer.

Gao, X., & Tang, X. (2002). Unsupervised video-shot segmentation and model-free anchorperson detection for news video story parsing. *IEEE Trans. CSVT, 12*(9), 765-776.

Gargi, U., Kasturi, R., & Strayer, S.H. (2000). Performance characterization of video-shot-change detection methods. *IEEE Trans. CSVT*, *10*(1), 1-13.

Greiff, W., Morgan, A., Fish, R., Richards, M., & Kundu, A. (2001). Fine-grained hidden markov modeling for broadcast news story segmentation. *ACM Multimedia 2001*.

Hanjalic, A., Kakes, G., Lagendijk, R.L., & Biemond, J. (2001). DANCERS: Delft Advanced News Retrieval System. *Proceedings of SPIE*, Vol. 4315, 301-310.

Hanjalic, A., Lagendijk, R.L., & Biemond, J. (1998). Template-based detection of anchorperson shots in news programs. *IEEE ICIP*, *3*, 148-152.

Hanjalic, A., Lagendijk, R.L., & Biemond, J. (1999). Semi-automatic news analysis, indexing and classification system based on topics pre-selection. *Proceedings of SPIE*, Vol. 3656, 86-97.

Hanjalic, A., & Zhang, H.J. (1999). Integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Trans. CSVT*, *9*(8), 1280-1289.

Hauptmann, A.G., & Witbrock, M.J. (1998). Story segmentation and detection of commercials in broadcast news video. *Proceedings of the Forum on Research and Technology Advances in Digital Libraries, ADL*, 168-179.

Huang, Q., Liu, Z., & Rosenberg, A. (1999). Automated semantic structure reconstruction and representation generation for broadcast news. *Proceedings of SPIE*, Vol. 3656, 50-62.

Jain, A.K., & Dubes, R.C. (1988). *Algorithms for clustering data*. Englewood Cliffs, NJ: Prentice-Hall.

Jasinschi, R.S., Dimitrova, N., McGee, T., Agnihotri, L., Zimmerman, J., & Li, D. (2001). Integrated multimedia processing for topic segmentation and classification. *IEEE ICIP*, *3*, 366-369.

Li, H., Doermann, D., & Kia, O. (2000). Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, *9*(1), 147-156.

Lu, L., & Zhang, H.J. (2002). Speaker change detection and tracking in real-time news broadcasting analysis. *Proceedings of the ACM International Multimedia Conference and Exhibition*, 602-610.

O'Connor, N., Czirjek, C., Deasy, S., Marlow, S., Murphy, N., & Smeaton, A. (2001). News story segmentation in the Fischlar video indexing system. *IEEE ICIP*, *3*, 418-421.

O'Toole, C., Smeaton, A. Murphy, N., & Marlow, S. (1999). Evaluation of automatic shot boundary detection on a large video test suite. *Challenge of Image Retrieval. CIR99 - Second UK Conference on Image Retrieval*, 12.

Pfeiffer, S., Lienhart, R. Fischer, S., Effelsberg, W. (1996). Abstracting digital movies automatically. *Journal of Visual Communication and Image Representation*, *7*(4), 345-353.

Qi, W., Gu, L., Jiang, H., Chen, X.R., & Zhang, H.J. (2000). Integrating visual, audio and text analysis for news video. *IEEE ICIP*, *3*, 520-523.

Raaijmakers, S., den Hartog, J., & Baan, J. (2002). Multimodal topic segmentation and classification of news video. *Proceedings of 2002 IEEE International Conference on Multimedia and Expo*, Vol. 2, 33-6.

Rui, Y., & Huang, T.S. (2000). Unified framework for video browsing and retrieval. In A. Bovik (Ed.), *Handbook of image and video processing,* 705-715. San Diego, CA: Academic Press.

Rui, Y., Huang, T.S., & Mehrotra, S. (1998). Exploring video structure beyond the shots. *Proceedings of the IEEE Conference on Protocols for Multimedia Systems and Multimedia Networking*, pp. 237-240.

Rui, Y., Huang, T.S., & Mehrotra, S. (1999). Constructing table-of-content for videos. *Multimedia Systems*, *7*(5), 359-368.

Rui, Y., Zhou, S.X., & Huang, T.S. (1999). Efficient access to video content in a unified framework. *International Conference on Multimedia Computing and Systems Proceedings*, *2*, 735-740.

Sato, T., Kanade, T., Hughes, E.K., & Smith, M.A. (1998). Video OCR for digital news archive. *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database* (Cat. No.98EX125), 52-60.

Sato, T., Kanade, T., Hughes, E.K., Smith, M.A., & Satoh, S. (1999). Video OCR: Indexing digital news libraries by recognition of superimposed captions. *Multimedia Systems, 7*(5), 385-395.

Smith, M.A., & Kanade, T. (1997). Video skimming and characterization through the combination of image and language understanding techniques. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 775-781.

Taniguchi, Y., Akutsu, A., Tonomura, Y., & Hamada, H. (1995). Intuitive and efficient access interface to real-time incoming video based on automatic indexing. *Proceedings of the ACM International Multimedia Conference & Exhibition*, 25-33.

Tsapatsoulis, N., Avrithis, Y., & Kollias, S. (2000). Efficient face detection for multimedia applications. *IEEE ICIP*, *2*, 247-250.

Wang, H., & Chang, S.-F. (1997). Highly efficient system for automatic face region detection in MPEG video. *IEEE Trans. CSVT, 7*(4), 615-628.

Yeung, M.M., & Yeo, B.-L.. (1996). Time-constrained clustering for segmentation of video into story units. *Proceedings of the 13th International Conference on Pattern Recognition*, Vol. 3, 375-80.

Zhang, H.J., Low, C.Y., Smoliar, S.W., & Wu, J.H. (1995). Video parsing, retrieval and browsing: An integrated and content-based solution. *Proceedings of the ACM International Multimedia Conference & Exhibition*, 15-24.

Zhang, T., & Kuo, C.C.J. (1999). Video content parsing based on combined audio and visual information. *Proceedings of SPIE - The International Society for Optical Engineering*, Vol. 3846, pp. 78-79.

Zhuang, Y., Rui, Y., Huang, T.S., & Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. *IEEE International Conference on Image Processing*, 1, 866-870.

Section VI

Video Information Retrieval

Chapter XII

An Overview of Video Information Retrieval Techniques

Sagarmay Deb, University of Southern Queensland, Australia

Yanchun Zhang, Victoria University of Technology, Australia

ABSTRACT

Video information retrieval is currently a very important topic of research in the area of multimedia databases. Plenty of research work has been undertaken in the past decade to design efficient video information retrieval techniques from the video or multimedia databases. Although a large number of indexing and retrieval techniques have been developed, there are still no universally accepted feature extraction, indexing, and retrieval techniques available. In this chapter, we present an up-to-date overview of various video information retrieval systems. Since the volume of literature available in the field is enormous, only selected works are mentioned.

INTRODUCTION

Research on multimedia systems and video information retrieval has been given tremendous importance during the last decade. The reason behind this is the fact that video databases deal with text, audio, video, and image data that could provide us with enormous amounts of information and that have affected our lifestyle for the better. The generation of huge amounts of data has also raised the necessity of developing efficient mechanisms for access and retrieval of these data, but the technologies developed so far have not matured enough to handle these issues. This is because manual classification and annotation of topics as provided by Yahoo!, a text-only search engine, is not capable of handling such a large-scale volume of data, creating an information bottleneck for the users trying to obtain specific information. To automate the procedure requires efficient and meaningful image segmentation and the ability to get semantic meanings from the image, but this is proving very difficult because of the gap existing between low-level features like color, texture and shape and high-level features like table, chair, car, house and so on. Automatic content-based information retrieval seems to be the only answer to the problem of efficient access and retrieval of the vast amounts of video data that are generated everyday throughout the world.

Basically there are four steps involved in any automatic video information retrieval. They are (a) video shot boundary detection, (b) key frame selection, (c) feature extraction from selected key frames, and (d) content-based video information retrieval. These points are discussed in the chapter.

The first section gives an introduction to the area of discussion. Related works on video information retrieval techniques are then presented in the next session. The chapter ends with our concluding remarks.

RELATED WORKS ON VIDEO INFORMATION RETRIEVAL TECHNIQUES

Video Shot Boundary Detection

To model the unstructured video data, the first thing to do is to segment the data into meaningful sections. This is achieved through video shot boundary detection (Bescos, Menendez, Cisneros, Cabrera, & Martinez, 2000; Ford, Robson, Temple, & Gerlach, 2000; Jiang, Helal, Elmagarmid, & Joshi, 1998; Lin, Kankanhalli, & Chua, 2000; Yu & Wolf, 1999; Zhang, Kankanhalli, Smoliar, & Tan, 1993). Since video is a kind of time-based media, it is very important to break the video streams into basic temporal units of shots (Zhang, Low, Gong, & Smoliar, 1995). This temporal partitioning of video is generally called video segmentation or shot boundary detection (Hong & Wolf, 1998; Idris & Panchanathan, 1997). When the video data is segmented, the low-level structure can be extracted. This temporal segmentation is the first and the basic requirement in structured modeling of video (Chua & McCallum, 1996; Yoshinobu, Akihito, Yukinobu, & Gen, 1994). These shots stand as basic blocks, which together define the whole story and are processed accordingly by the computer (Davenport, Smith, & Pincever, 1991). To segment the video into shots, we need to locate the points where shots change from one type to another type based on the techniques used in the editing process (Hampapur, Jain, & Weymouth, 1994).

Data is stored in one of two ways, uncompressed or compressed. Basically two categories of video shot boundary detection techniques have been developed. One category is used for uncompressed data, whereas the other category is applicable to compressed data. Methods in the uncompressed domain have been broadly classified into five groups: template-matching, histogram-based, twin comparison, block-based and model-based, whereas methods in the compressed domain have been classified into three groups: using DCT coefficients of video-compression techniques in the frequency domain, making use of motion vectors and hybrid motion/DCT, which combines the first two categories (Farag & Abdel-Wahab, 2003). A new shot boundary algorithm had been proposed to deal with false shot detection due to flashing lights. Earlier, it was assumed that flashlight just occurred during one frame, but in practice, it could occur many times during a period; to rectify this problem, a “flash model” and a “cut model” have been used. A technique for determining the threshold that uses the local window-based method combined with a reliability verify process has also been developed (Zhang, Qi, & Zhang, 2001).

The motion feature vector is incorporated in the video shot boundary detection technique to detect flash and camera/object motion and automatically select thresholds for noise elimination by determining the type of video (Feng, Chandrashekhara, & Chua, 2003). An automatic video sequence segmentation algorithm to extract moving objects has been presented. The algorithm determines the local variation based on L*U*V* space and combines it with motion information to separate foreground from the background (Liu & Hou, 2001).

Video classification has been an important topic of research for many years. In a study on story segmentation for news video by Chaisorn, Chua, and Lee (2003), the video is analyzed at the shot and story unit (or scene) levels using a variety of features and techniques. At the shot level, Decision Tree technique has been employed to classify the shots into one of 13 predefined categories. At the scene/story level, Hidden Markov Models (HMM) analysis to locate story boundaries is performed. In their work, Wang and Gao (2001b) integrated audio and visual analysis to automatically segment news items, and found that this approach can overcome the weakness of only using the image analysis technique. The proposed approach identifies silence segments in accompanying audio and integrates them with shot segmentation results and anchor shot detection results to determine boundaries between news items. To automatically parse news video and extract news item, a fast anchor shot detection algorithm based on background chrominance and skin tone models was presented by these authors.

An automatic two-level approach to segment videos into semantically meaningful abstracted shots has been proposed by Chen and Lee (2001). In the first level, scene changes are detected using a GOP-based approach that assists in quickly segmenting a video sequence into shots. In the second level, each of the shots generated from the first level is analyzed by utilizing the information of camera operations and object motion that are computed directly from motion vectors of MPEG-2 video streams in compressed domain.

In another study (Shih, Tyan, & Liao, 2001), a general shot change detection algorithm has been proposed where two major stages, modeling and detection, are involved. In modeling stage, shot transition is modeled by calculating the change of color/intensity distribution corresponding to the shots before and after a transition. In detection stage, a video clip is considered as a continuous “frame flow,” and then the Reynolds Transport Theorem is applied to analyze the flow change within a predetermined control volume.

Key Frames Selection

Once the shot boundaries are defined, we have to develop efficient algorithms to extract the large amount of information found in segmented shots (Farag & Abdel-

Waheb, 2003). This is achieved by selecting representative or key frames for each shot (Marques & Furht, 2002). In selecting key frames to represent video shots, various features, such as motion, color, and region information are normally used (Kang, 1999; Zhang, Wu, Zhong, & Smoliar, 1997). Selection of key frames will greatly affect the amount of information that can be captured and the types of queries that are possible (Chen, Ozsu, & Oria, 2003). In a visual content-based approach , the first frame is selected as key frame. But in addition to this, other frames could also be selected as additional key frames if these have substantial content changes compared to the first key frame (Zhang, Kankanhalli, & Mulhem, 2003). There are several other key frame selection algorithms have been proposed (Defaux, 2000; Gunsel & Tekalp, 1998; Yeung & Yeo, 1996; Zhang et al., 1993) that use low-level visual features like color, shape, texture, luminance, and motion to select key frames. Since these algorithms do not reflect semantic changes within the shot such as change of a spatial relationship, manual and automatic interpretation techniques have been combined to identify appearance or disappearance of salient objects and changes in spatial relationships. A key frame is selected to represent the time within the shot where spatial relationships among salient objects contained in that video frame hold (Chen et al., 2003).

Then an algorithm based on clustering has been proposed where the algorithm assumes there are N frames within a shot divided into M clusters. It selects key frames from clusters having number of frames greater than N/M. The color histograms are used to obtain similarity between frames, and the frame that stands closest to the central point of the cluster is selected as the key frame (Zhuang, Rui, Huang, & Mehrotra, 1998).

Optical flow (motion) has been used to extract the key frame within a shot (Umamaheswaran, Huang, Palakal, & Suyut, 2002) where optical flow is the distribution of apparent velocities of movement of brightness patterns in an image, and is able to easily detect the change in motion in two successive frames in a video sequence. The frame showing the least motion is chosen as the key frame.

Feature Extractions

In an analysis and interpretation of object-based videos, an attempt has been made to address two very important issues: (a) the development of a comprehensive algorithm that extracts objects from low-level features, and (b) the modeling of object behavior to a semantic level (Luo, Hwang, & Wu, 2003). Also, for efficient video representation, a study by Chen et al. (2003) contributes by establishing links between image and video data and considers the relationship between videos and images. This proposed model expresses the semantics of video data content by means of salient objects and the relationships among them. Connections between video and image data are made through key frames, which are extracted from each shot. Based on these connections, techniques used to query image data may be used to query video data.

A semantic object-generation algorithm is proposed by Fan, Zhu, and Wu (2001) using collaborative integration of a new feature-based image segmentation technique and a seeded region aggregation procedure. The homogeneous image regions with closed boundaries are obtained by integrating the results of color edge detection and seeded region growing procedures. The object seeds are then distinguished from these homogeneous image regions. The semantic objects are finally generated by a seeded region aggregation procedure. The extracted semantic objects can then be tracked along the time axis. The video objects can then be used as the basic units for content-based video indexing.

With the enormous volume of digital information being generated in multimedia streams, results of queries are becoming very voluminous. That makes the manual classification/annotation in topic hierarchies through text, result in information bottleneck, and it is becoming unsuitable for addressing users' information needs. Creating and organizing a semantic description of unstructured data is very important to achieve efficient discovery and access of video data. But automatic extraction of semantic meaning out of video data is proving difficult because of the gap existing between low-level features like color, texture and shape and high-level semantic descriptions like table, chair, car, house and so on. A method has been proposed where a general hierarchical semantic concept tree is used to model and abstract the semantics of videos using sports videos (Zhou & Dao, 2001). The classification of video data has been achieved by extracting patterns in the temporal behavior of each variable and also in dynamics of relationship between variables and mapping these patterns to a high-level interpretation through the use of Dynamic Bayesian Network (DBA) framework (Mittal & Altman, 2003).

In order to achieve automatic summarization based on personal interest, an approach to automatic structuring and summarization of video captured by wearable camera is suggested by Aizawa, Ishijima, and Shiina (2001) where, in addition to summarization based on objective visual features of video and audio, subjective feelings of the person also have been taken into account. Another video summarization algorithm that works in the compressed domain, in particular for MPEG videos, has been presented. A feature called DC histogram that can be extracted from MPEG videos without full decompression has been used (Chew & Kankanhalli, 2001; Erol & Kossentini, 2001).

To detect movie events including two-speaker dialog scenes, multiple-speaker dialog scenes and hybrid scenes from the daily movies, visual information is employed to detect all possible shot sinks by using a window-based sweep algorithm. All shot sinks are then categorized into three classes using K-means algorithm. The accompanying audio cue will also be utilized for achieving more meaningful results. The authors claim that by integrating audio-visual information, meaningful events could be successfully detected and extracted (Li & Kuo, 2001).

The complex background of the video frame makes it difficult to detect text area from video frames. To extract text area, vertical edge information is used to detect candidate text area. The horizontal edge information is then used to eliminate some of the false candidates. Finally, shape suppression technique is applied to further refine the results (Chen & Zhang, 2001). A technique to segment object(s) from video sequence by color quantization has been presented by Oh (2002). According to the author, the scheme has been made cost effective by avoiding expensive computations and automatic by removing manual processing.

Content-based Video Information Retrieval

In content-based video information retrieval, we have to extract various low-level features such as color, text, shape and spatial locations from the query image and based upon those features, symmetry has to be searched for and established with the similar extractions from the objects of the video database. The video data model of the Digital Video Album (DVA) system has been introduced to support semantic content-based video indexing and vague query. Based on a predefined XML-schema, this model allows high-level semantic annotation to describe the content of video including objects and events. The annotation process makes use of low-level features processes during tracking of objects and detection of faces (Zhang et al., 2003).

To achieve location-based recollection of past events, three functional methods — image retrieval with motion information, video scene segmentation, and real-time video retrieval—have been proposed by Kawamura, Kono and Kidode (2001). The authors claim experimental results showed that these functions are effective to perform wearable information playing.

A method for searching and browsing news video based on multi-modal approach has been proposed (Kim, Kim, Chang, Kang, & Kim, 2001). Closed caption (CC) data has been used to index the contents of TV news article. To achieve time alignment between the CC texts and video data, a supervised speech recognition technique is employed.

In another paper (Jeong & Moon, 2001), an algorithm for content-based video retrieval using motion information has been proposed. In temporal scale invariant and spatial translation absolute retrieval using a trail model, the distance transformation is performed on each trail image in the database and then, from a given query trail, the pixel values along the query trail are added in each distance image to compute the average distance between the trails of query image and database image. In temporal scale absolute and spatial translation invariant retrieval using a trajectory model, a new coding scheme referred to as the Motion Retrieval Code is proposed by Jeong and Moon (2001) that they claim to be suitable for representing object motions in video.

To detect the human face in digital video, an automatic face detection system using a support vector machine (SVM) ensemble framework combining several SVMs in the scheme of majority voting has been proposed. It starts with skin block classification stage to classify the 8x8 block in each key frame of digital video into skin/non-skin blocks using the skin color and texture information. This stage can search and detect several face candidate regions. The next stage finds the true face position in a candidate region that includes a facial feature. This information can be used for automatic face recognition on MPEG-7 standard for video indexing (Je, Kim, Bang, Lee, & Choi, 2002). A video-based face recognition system by support vector machines is presented by Zhuang et al. (2002). The authors used Stereovision to coarsely segment the face area from its background and then multiple-related template matching method is used to locate and track the face area in the video to generate face samples of that particular person. Face recognition algorithms are based on Support Vector Machines of which both “1 vs. many” and “1 vs. 1” strategies are discussed.

CONCLUSION

An account of the research being carried out in the field of video information retrieval has been presented. The research in this field has come a long way during the last decade, but it has still a long way to go to provide users with tools to retrieve video information efficiently. One of the major problems in this area is the problem of computer perception, that is, bridging the gap that exists between low-level features like color, texture, shape, and spatial relationships and high-level features like table, chair, car, and so on. No proper solution to this problem has been achieved so far. Finding the correct symmetry between input image and image of the database using color, texture, shape, and spatial relationships also still remains to be resolved, although some progress has been achieved. XML-based image annotation of the images at the time of creation of the images in an automatic way has been suggested but not yet implemented. This method has the potentiality to solve the problem of computer perception being faced by researchers in this field. But this method is still in its infant stage and needs to be developed further to attain maturity. Automating the efficient video shot boundary detection techniques also remain a problem, although some progress has been made. Selection of appropriate key frames is another issue that needs more research attention. In short, there are difficult research issues still unresolved in the area that calls for more coordinated research efforts in the coming years.

REFERENCES

Aizawa, K., Ishijima, K., & Shiina, M. (2001). Automatic summarization of wearable video-indexing subjective interest. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*, *Beijing*, *China*, October 24-26, 16-23.

Bescos, J., Menendez, J.M., Cisneros, G., Cabrera, J., & Martinez, J.M. (2000). A unified approach to gradual shot transition detection. *Proceedings of International Conference on Image Processing*, Vol. III, 949-952.

Chaisorn, L., Chua, T., & Lee, C. (2003). A multi-model approach to story segmentation for news video. *World Wide Web, Internet and Web Information Systems*, 6, 187-208.

Chen, D., & Lee, S. (2001). Motion-based semantic event detection for video content description in MPEG-7. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 110-117.

Chen, L., Ozsu, M. T., & Oria, V. (2003). Modeling video data for content-based queries: Extending the DISIMA image data model. *Ninth International Conference on Multi-Media Modeling Proceedings*, January 8-10, Tamsui, Taiwan, 169-189.

Chen, X., & Zhang, H. (2001). Text area detection from video frames. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 222-228.

Chew, C.M., & Kankanhalli, M.S. (2001). Compressed domain summarization of digital video. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 490-497.

Chua, T., & McCallum, J.C. (1996). Multimedia modeling. In A. Kent & J.G. Williams (Eds.), *Encyclopedia of Computer Science and Technology*, Vol. 35, 253-287. New York: Marcel Dekker.

Davenport, G., Smith, T., & Pincever, N. (1991). Cinematic primitives for multimedia. *IEEE Computer Graphics & Applications*, *11*(4), 67-74.

Defaux, F. (2000). Key frame selection to represent a video. *Proceedings of IEEE International Conference on Image Processing*, 275-278.

Erol, B., & Kossentini, F. (2001). Color content matching of MPEG-4 video objects. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 891-896.

Fan, J., Zhu, X., & Wu, L. (2001). Seeded semantic object generation toward content-based video indexing. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 837-842.

Farag, W.E., & Abdel-Wahab, H. (2003). Video content-based retrieval techniques. In S. Deb (Ed.), *Multimedia Systems and Content-based Image Retrieval,* 114-154. Hershey, PA: Idea Group Publishing.

Feng, H., Chandrashekhara, A., & Chua, T. (2003). ATMRA: An Automatic Temporal Multi-resolution Analysis Framework for shot boundary detection. *Ninth International Conference on Multi-Media Modeling Proceedings*, January 8-10, Tamsui, Taiwan, 224-240.

Ford, R.M., Robson, C., Temple, D., & Gerlach, M. (2000). Metric for shot boundary detection in digital video sequences. *Multimedia Systems, 8*, 37-46.

Gunsel, B., & Tekalp, A.M. (1998). Content-based video abstraction. *Proceedings of IEEE International Conference on Image Processing*, 128-131.

Hampapur, A., Jain, R., & Weymouth, T.E. (1995). Production model based digital video segmentation. *Multimedia Tools and Applications, 1*(1), 9-46.

Hong, H. Y., & Wolf, W. (1998). Multi-resolution video segmentation using wavelet transformation. *Storage and Retrieval for Image and Video Database (SPIE)*, 176-187.

Idris, F., & Panchanathan, S. (1997, June). Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation, 8*(2), 146-166.

Je, H.M., Kim, D., Bang, S.Y., Lee, S., & Choi, Y. (2002). Human face detection in digital video using SVM ensemble. *Proceedings of 6th Joint Conference on Information Sciences*, March 8-12. Research Triangle Park, NC, 417-421.

Jeong, J.M., & Moon, Y.S. (2001). Efficient algorithms for motion based video retrieval. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 909-914.

Jiang, H., Helal, A., Elmagarmid, A.K., & Joshi, A. (1998). Scene change detection technique for video database systems. *Multimedia Systems, 6*, 186-195.

Kang, H. (1999). Key frame selection using region information and its temporal variations. *Proceedings IASTED Conf. IMSA'99*, Nassau, Bahamas.

Kawamura, T., Kono, Y., & Kidode, M. (2001). A novel video retrieval method to support a user's recollection of past events aiming for wearable information playing. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 24-31.

Kim, Y., Kim, J., Chang, H., Kang, K., & Kim, J. (2001). Content-based news video retrieval with closed captions and time alignment. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 879-884.

Li, Y., & Kuo, C.C.J. (2001). Movie event detection by using audio visual information. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 198-205.

Lin, Y., Kankanhalli, M.S., & Chua, T.S. (2000). Temporal multi-resolution analysis for video segmentation. *Proceedings of the International Conference of SPIE (Storage and Retrieval for Media Databases),* San Jose, CA, Vol. 3972, January, 494-505.

Liu, M., & Hou, C. (2001). Automatic segmentation and tracking of moving objects. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 214-221.

Luo, Y., Hwang, J., & Wu, T. (2003). Object-based video analysis and interpretation. In S. Deb (Ed.), *Multimedia systems and content-based image retrieval*, 182-199. Hershey, PA: Idea Group Publishing.

Marques, O., & Furht, B. (2002). Content-based visual information retrieval. In T. Shih (Ed.), *Distributed multimedia databases: Techniques and applications*, 37-57. Hershey, PA: Idea Group Publishing.

Mittal, A., & Altman, E. (2003). Contextual information extraction for video data. *Ninth International Conference on Multi-Media Modeling Proceedings*, January 8-10, Tamsui, Taiwan, 209-223.

Oh, J. (2002). Key object(s) extraction from video sequences using color quantization. *Proceedings of 6th Joint Conference on Information Sciences*, March 8-13, Research Triangle Park, NC, 923-926.

Shih, C.C., Tyan, H.R., & Liao, H.Y.M. (2001). Shot change detection based on the Reynolds Transport Theorem. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 819-824.

Umamaheswaran, D., Huang, J., Palakal, M., & Suyut, S. (2002). Video scene analysis using best basis wavelets and learning strategies. *Proceedings of 6th Joint Conference on Information Sciences*, March 8-12, Research Triangle Park, NC, 680-683.

Wang, W., & Gao, W. (2001a). Automatic segmentation of news items based on video and audio features. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 498-505.

Wang, W., & Gao, W. (2001b). A fast anchor shot detection algorithm on compressed video. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 873-878.

Yeung, M.M., & Yeo, B.L. (1996). Time-constrained clustering for segmentation of video into story units. *Proceedings of 13th International Conference on Pattern Recognition*, 375-380.

Yoshinobu, T., Akihito, A., Yukinobu, T., & Gen, S. (1994). Structured video computing. *IEEE Multimedia*, *1*, 34-43.

Yu, H., & Wolf, W. (1999, Jul/Aug). A hierarchical multi-resolution video shot transition detection scheme. *Computer Vision and Image Understanding*, *75*(1/2), 196-213.

Zhang, D., Qi, W., & Zhang, H.J. (2001). A new shot boundary detection algorithm. *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 63-70.

Zhang, H., Low, C.Y., Gong, Y., & Smoliar, S.W. (1995). Video parsing using compressed data. *Multimedia Tools and Applications*, *1*(1), 91-111.

Zhang, H., Wu, J., Zhong, D., & Smoliar, S. (1997). An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, *30*(4), 643-658.

Zhang, H.J., Kankanhalli, A., Smoliar, S., & Tan, S. (1993). Automatic partitioning of full-motion video. *Multimedia Systems*, *1*(1), 10-28.

Zhang, Q., Kankanhalli, M.S., & Mulhem, P. (2003). Semantic video annotation and vague query. *Ninth International Conference on Multi-Media Modeling Proceedings*, January 8-10, Tamsui, Taiwan, 190-208.

Zhou, W., & Dao, S.K. (2001). Combining hierarchical classifiers with video semantic indexing systems. *Proceedings of Advances in Multimedia Information Processing – PCM 2001, Second IEEE Pacific Rim Conference on Multimedia*, Beijing, China, October ,78-85.

Zhuang, L., Ai, H., & Xu, G. (2002). Video based face recognition by support vector machines. *Proceedings of 6th Joint Conference on Information Sciences*, March 8-13, Research Triangle Park, NC, 700-703.

Zhuang, Y., Rui, Y., Huang, T., & Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. *Proceedings of IEEE International Conference on Image Processing*, 866-870.

Chapter XIII

A Framework for Indexing Personal Videoconference

Jiqiang Song, Chinese University of Hong Kong, Hong Kong

Michael R. Lyu, Chinese University of Hong Kong, Hong Kong

Jenq-Neng Hwang, University of Washington, USA

ABSTRACT

The rapid technical advance of multimedia communication has enabled more and more people to enjoy videoconferences. Traditionally, the personal videoconference is either not recorded or only recorded as ordinary audio and video files that only allow linear access. Moreover, in addition to video and audio channels, other videoconferencing channels, including text chat, file transfer, and whiteboard, also contain valuable information. Therefore, it is not convenient to search or recall the content of videoconference from the archives. However, there exists little research on the management and automatic indexing of personal videoconferences. The existing methods for video indexing, lecture indexing, and meeting support systems cannot be applied to personal videoconference straightforwardly. This chapter discusses important issues unique to personal videoconference and proposes a comprehensive framework for indexing personal videoconference. The framework consists of three modules: videoconference archive acquisition module, videoconference archive indexing module, and indexed videoconference accessing module. This chapter will elaborate on the design principles and implementation methodologies of each module, as well as the *intra- and inter-module data flows and control flows. Finally, this chapter presents a subjective evaluation protocol for personal videoconference indexing.*

INTRODUCTION

Videoconference is an advanced type of meeting approach that employs real-time video communication technology to enable participants in different geographical locations to see and talk to each other. In fact, today's videoconference employs richer communication media than audio/video, such as text chat, file transfer, whiteboard, and shared applications. Therefore, it should be more precisely called a "multimedia conference." By convention, we still use the term "videoconference" in this chapter. A few years ago, videoconference was only an expensive option for big companies' business operations due to the requirement of special hardware and communication lines. With the growth of Internet bandwidth and the development of multimedia communication technologies in past years, videoconference has become more and more popular in business operations (Sprey, 1997). Furthermore, with affordable video and audio capture devices, the advanced low bit-rate coding, and pure-software videoconferencing tools, home users can also enjoy visual communications at 56Kbps or lower (Deshpande & Hwang, 2001). For example, video-based distance learning has benefited significantly from videoconferencing techniques.

A videoconference can be classified as either *personal videoconference* or *group videoconference* based on the number of participants at each geographical location. A videoconference is classified as a personal videoconference on the condition that there is only one participant at each location; otherwise, it is a group videoconference. A personal videoconference is usually held among several participants, and each participant sits in front of a computer equipped with a camera, a microphone, and a speaker (or earphone). A group videoconference is often held in a multimedia conference room, where more than one camera and microphones are installed. Existing research on videoconference indexing all focuses on group videoconferences, leading to meeting support systems. Since personal videoconferences are becoming more and more popular recently and the characteristics of personal videoconferences are different from that of group videoconferences, this chapter aims to propose a framework for indexing personal videoconferences.

A participant of a personal videoconference usually wishes to save the content of the conference for the later reference. However, current videoconferencing systems provide little support on this aspect. Even if the streaming video and audio information can be recorded as ordinary video and audio files, these files occupy too much space and do not support non-linear access, so it is not easy to recall the details of a videoconference without watching it again. Therefore, it is very difficult to manage and search those videoconference records, revealing the timely importance of the research on indexing personal videoconferences.

The rest of this chapter is organized as follows. The next section reviews the background of multimedia indexing. Then, we discuss the characteristics of personal videoconference indexing, followed by a proposed comprehensive framework for building a personal videoconference archive indexing system. A subjective evaluation protocol is then presented in the next section. Finally, we draw our conclusions.

BACKGROUND

When talking about videoconference archive indexing, one cannot ignore what has been researched in general video indexing because video — containing both visual and aural information — is the major medium during a videoconference. Before the wide deployment of videoconferencing, research on general video indexing had been conducted for several years (Madrane & Goldberg, 1994; Sawhney, 1993). Since the content of a video clip is hidden in its visual and aural information, which is not directly searchable like text, the main purpose of video indexing is to support content-based video retrieval (CBVR). CBVR is more complicated than the content-based image retrieval (CBIR) since a video clip is a time-varying image/audio sequence (Zhang, Wang, & Altunbasak, 1997). Therefore, the temporal semantic structure of a video clip also implies the video content. The success of video indexing research, not yet accomplished, will result in a digital video library featuring full-content and knowledge-based search and retrieval. A well-known effort in this field is the Informedia project (Wactlar Kanade, Smith, & Stevens, 1996) undertaken at Carnegie Mellon University.

Video indexing has been studied in multifarious ways, producing a lot of useful techniques for multimedia information retrieval and indexing. Early research mostly draws on single or dual modality of video content analysis. Liang, Venkatesh, and Kieronsak (1995) focused on the spatial representation of visual objects. Ardizzone, La Cascia, Di Gesu, & Valenti (1996) utilized both global visual features (e.g., color, texture, and motion) and local visual features (like object shape) to support the content-based retrieval. Chang, Zeng, Kamel, and Alonso (1996) integrated both image and speech analyses for news or sports video indexing. Ardizzone and La Cascia (1996), Wei, Li, and Gertner (1999), Zeng, Gao, and Zhao (2002), and Hsu and Teng (2002) proposed that motion was also an important clue for video indexing. To take advantage of video encoding features, some video indexing work was performed in the compressed domain (Chang, 1995). Later, multimodal information was explored for video indexing (Hauptmann & Wactlar, 1997). Li, Mohan, and Smith (1998) and Lebourgeois, Jolion, and Awart (1998) presented multimedia content descriptions of video indexation. Abundant multimodal video indexing methods have also been proposed (Albiol, Torres, & Delp, 2002; Lyu, Yau, & Sze, 2002; Tsekeridou & Pitas, 1998; Viswanathan, Beigi, Tritschler, & Maali, 2000). They extracted information in many aspects, including face, speaker, speech, keyword, etc.

On the other hand, high-level indexing aided by domain-specific knowledge has also attracted much interest. Dagtas, Al-Khatib, Ghafoor, and Khokhar (1999) proposed a trail-based model utilizing object motion information. Hidalgo and Salembier (2001) segmented and represented foreground key regions in video sequences. Maziere, Chassaing, Garrido, and Salembier (2000) and Hwang and Luo (2002) conducted object-based video analysis. Ben-Arie, Pandit, and Rajaram (2001) and Wang, Ma, Zhang, and Yang (2003) performed view-based human activity recognition and people similarity-based indexing, respectively.

Recently, researchers focus on understanding the semantic structure of video sequences, that is, story, scene, and shot. Segmenting video into shots is a fundamental task for this purpose. Segmentation is the partitioning of continuous media into homogenous segments. There exist many ways for shot segmentation, for example, by camera motion (Corridoni & Del Bimbo, 1996), using the human face (Chan, Lin, Tan, &

Kung, 1996), analyzing image basic features (Di Lecce et al., 1999), and counting the percentage of moving pixels against the background (Kang & Mersereau, 2002). It is also important to detect gradual shot transitions (Bescos, Menendez, Cisneros, Cabrera, & Martinez, 2000). The segmented shots are annotated for subsequent searches (Ito, Sato, & Fukumura, 2000; Wilcox & Boreczky, 1998) and for automatically summarizing the title (Jin & Hauptmann, 2001). Based on the semantic structure analysis, various semantic indexing models or schemes have been proposed (Del Bimbo, 2000; Gao, Ko, & De Silva, 2000; Gargi, Antani, & Kasturi, 1998; Iyengar, Nock, Neti, & Franz, 2002; Jasinschi et al., 2001a; Jasinschi et al., 2002; Luo & Hwang, 2003; Naphade & Huang, 2000; Naphade, Kristjansson, Frey, & Huang, 1998). Most of them employ probabilistic models, for example, Hidden Markov Models (HMMs) and/or Dynamic Bayesian networks (DBNs), to represent the video structure.

The extensive research on video indexing has produced the following valuable techniques to analyze the video and audio archives of a videoconference.

- *Key Frame Selection.* One basic means to remove the redundancy of video sequences is to represent them by key frames. Some methods work in the MPEG compressed domain (Calic & Lzquierdo, 2002; Kang, Kim, & Choi, 1999; Tse, Wei, & Panchanathan, 1995), whereas others work in the uncompressed domain (Diklic, Petkovic, & Danielson, 1998; Doulamis, Doulamis, Avrithis, & Kollias, 1998; Kim & Park, 2000).
- *Face Detection and Recognition.* Since a videoconference is always human-centric, human faces are principal objects in the video. Sato and Kanade (1997) associated human faces with corresponding closed-caption text to name the faces. Ariki, Suiyama, and Ishikawa (1998) indexed video by recognizing and tracking faces. Gu and Bone (1999) detected faces by spotting skin color regions. Tsapatsoulis, Avrithis, and Kollias, (2000) and Mikolajczyk, Choudhury, and Schmid (2001) proposed temporal face detection approaches for video sequences. Eickeler, Walhoff, Lurgel, and Rigoll (2001) and Acosta, Torres, Albiol, and Delp (2002) described content-based video indexing system using both face detection and face recognition.
- *Speech Recognition.* Speech of a videoconference contains the most information. In fact, there are many video-indexing methods only focusing on the audio track. Hauptmann (1995) analyzed the uses and limitations of speech recognition in video indexing. The audio track can also be segmented for acoustic indexing (Young, Brown, Foote, Jones, & Sparck-Jones, 1997). Barras, Lamel, and Gauvain (2001) proposed an approach to obtain the transcript of the compressed audio.
- *Speaker Identification.* Knowing who is speaking greatly enhances the videoconference indexing. Speaker identification can be performed in video or audio only, or using audio and video correlation (Cutler & Davis, 2000; Nam, Cetin, & Tewfik, 1997; Wilcox, Chen, Kimber, & Balasubramanian, 1994).
- *Keyword Spotting.* Other than recognizing every word in the speech to obtain the transcript of the speech for later textual analysis, one can use a domain-specific keyword collection to spot keywords in the speech for the indexing purpose (Dharanipragada & Roukos, 1996; Gelin & Wellekens, 1996).

- *Video Text Detection.* Since plain text is no doubt the most convenient medium for indexing and searching, one may try to extract as much as possible textual information from audio and video. Thus, video text detection has always been emphasized (Cai, Song, & Lyu, 2002; Hua, Yin, & Zhang, 2002; Jain & Yu, 1998; Kim, Kim, Jung, & Kim, 2000; Li & Doermann, 1998).
- *Topic Classification.* Locating the topic-switching point and summarizing the topic for a conference session are critical to deliver the accurate, both in time and topic, results to content-based searches. Jasinschi, Dimitrova, McGee, Agnihotri, and Zimmerman (2001b) integrate multimedia processing to segment and classify topics. Ngo, Pong, and Wang (2002) detect slide transitions for topic indexing

In addition to the research on video indexing, other research related to videoconference archive indexing includes lecture indexing systems and meeting support systems. Lecture indexing focuses on two points: (1) how to collect media archives of attractive lectures automatically, and (2) how to access these recorded archives. For the first point, the problem is subdivided into "what should be captured?" and "how it should be captured?" Since a lecture involves one or more speakers and an audience, who may interact with each other from time to time, both speakers and audience should be captured. Besides the verbal cues, non-verbal communication cues, e.g., body posture, gestures, and facial expressions, also play an important role during the interaction (Ju, Black, Minneman, & Kimber, 1997). Slides and handwritings on the whiteboard cannot be missed either. To capture the multimodal information, one or more cameras and microphones are necessary to construct an intelligent lecture room (Joukov & Chiueh, 2003; Kameda, Nishiguchi, & Minoh, 2003; Rogina & Schaaf, 2002; Stewart, Wolf, & Heminje, 2003). Kameda et al. (2003) even utilized ultrasonic equipment to capture the movement trajectory of the speaker. One principle of the capturing process is to keep it minimally intrusive. For the second point, these systems can be divided into two groups by providing static functionality or dynamic functionality, according to Stewart et al. (2003). The systems with static functionality assume that the media archives, once captured and produced, become frozen assets for later playback use. On the other hand, dynamic systems provide some extended functionality so the media assets can be edited and reused in addition to presentation.

Meeting support systems focus on real-life meetings or group videoconferences. These systems develop special techniques or devices to free participants from paper-based note-taking, to record meeting activities, and to provide post-meeting information sharing. These systems also follow the principle of "minimally intrusive" to record meetings. Chiu, Kapuskar, Reitmeier, and Wilcox (2000) described the multimedia conference room in FX Palo Alto Laboratory, which is well equipped with three room cameras, one videoconference camera, one document camera, ceiling microphones, rear projector screen, whiteboard, and wireless pen computers. The indices of a captured meeting can be classified into *direct indices* and *derived indices*. Direct indices are created online during the meeting capturing, whereas derived indices require further, usually offline, analyses.

Direct indices include meeting activities and participants' comments. Ginsberg and Ahuja (1995) explored various ways to visualize meeting activities, such as joining or leaving a meeting, using whiteboard, and more. There exist many tools to record participants' comments. NoteLook (Chiu et al., 1999) allows participants to take handwritten notes for indexing audio and video. LiteMinutes (Chiu, Boreczsky, Girgensohn, & Kimber, 2001) supports typing notes or minutes on a laptop, where each line of text acts as an index. TeamSpace (Geyer, Richter, & Abowd, 2003) emphasizes tracing those domain-specific artifacts that connect several meetings.

Creating derived indices focuses mostly on audio, video, and slides. There are many kinds of analyses that can be done on audio streams, for example, speech recognition, speaker identification, keyword spotting, and interaction pattern classification. The Jabber system (Kazman, Al-Halimi, Hunt, & Mantei, 1996; Kazman & Kominek, 1999) proposes four paradigms for indexing videoconferences by audio. A speech recognizer is used to obtain a partial transcript of the meeting, and topics or themes are identified using lexical chaining. In addition, participants' interaction patterns, for example, discussion or presentation, are classified. Kristjansson, Huang, Ramesh, and Juang (1999) also described a unified framework for indexing and gisting spoken interactions of people. Foote, Boreczsky, and Wilcox (1999) introduced an approach to find presentations in recorded meetings by correlating the slide intervals in video streams and the speaker-spotting results in audio streams. Gross et al. (2000) developed a system that did speaker identification, speech recognition, action item recognition, and auto summarization. The eMeeting system (Leung, Chen, Hendriks, Wang, & Sahe, 2002) provided a slide-centric recording and indexing system for interactive meetings.

The review of literature indicates that personal videoconference indexing has seldom been addressed, except for our preliminary study (Song, Lyu, Hwant, & Cai, 2003). Since personal videoconferencing is booming and its indexing bears different characteristics from existing video library indexing systems, lecture indexing systems, and meeting support systems, this chapter will propose a comprehensive framework for building a Personal Videoconference Archive Indexing System (PVAIS).

CHARACTERISTICS OF A PVAIS

This section describes the characteristics of a PVAIS in three aspects: archive acquisition, archive indexing, and indexed archive accessing and presenting.

In the archive acquisition process, the principle of "minimally intrusive" applies. The scenario of personal videoconferencing is that each participant sits in front of a computer, using pure-software videoconferencing client to communicate with each other. Different from video indexing, where only audio and video are available, a PVAIS considers six communication channels, including five medium channels and one conference control channel. The five medium channels are the audio channel, video channel, text chat channel, file transfer channel, and whiteboard channel. The conference control channel contains member information and conference coordination information. Different from lecture indexing and group videoconference indexing, where additional capturing equipment is used, a PVAIS is restricted to software-based capturing because there is no room for additional equipment between a participant and his/her computer.

The archive indexing process of a PVAIS should also be transparent to the user. Since a personal videoconference is not as well-organized as a broadcasting video program, the semantic video structure is not emphasized in a PVAIS. Instead, a PVAIS focuses on the conference events from the user's viewpoint. Such conference events include both media events (e.g., speaker changed, topic switched, text transmitted, whiteboard updated, and file transferred) and control events (e.g., member joined or left, channel created or closed). In addition to these conference events, a PVAIS also emphasizes the multimodal derived indices to provide content-based searches. As a personal indexing system, a PVAIS can automatically create and maintain a contact list for the user.

The access to the indexed videoconference archives is restricted to the authenticated users only. A PVAIS provides an interface for the user to search and review the indexed videoconference archives. The user also needs to manage and edit the indexed items via this interface. The user can conduct searches with various criteria based on content of interest or videoconferencing events through the interface. A PVAIS supports synchronized presentation of multimedia searching results. During the presentation, a PVAIS allows the user to pause at any time and add annotations or bookmarks.

FRAMEWORK FOR A PVAIS

Figure 1 shows the top-level view of the framework of a PVAIS, which consists of three separate processing modules with a linear processing order:

- *Videoconference Archive Acquisition Module*. This module works together with the personal videoconferencing terminal to extract and save the raw content archives from all communication channels *in real time*. This module can be either embedded in the terminal or separated from it.
- *Videoconference Archive Indexing Module*. This module works *offline* after the videoconference finishes. It takes the raw videoconference archives as input, analyzes the videoconferencing events, integrates information from multiple channels to produce derived indices, and finally outputs the indexed videoconference archives.
- *Indexed Videoconference Accessing Module*. This module serves as the interface to users. It provides the functions of managing indexed videoconferences, searching for content of interest among them, and presenting the selected videoconference.

This section further elaborates the design principles and implementation methodologies for each module. First, we briefly introduce the knowledge of videoconferencing system. Generally, a videoconferencing system could be implemented in various archi-

Figure 1. Top-level view of the framework of a PVAIS

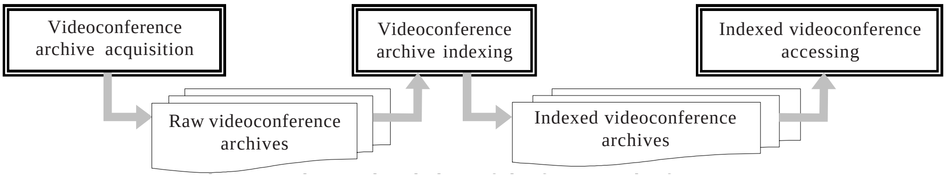
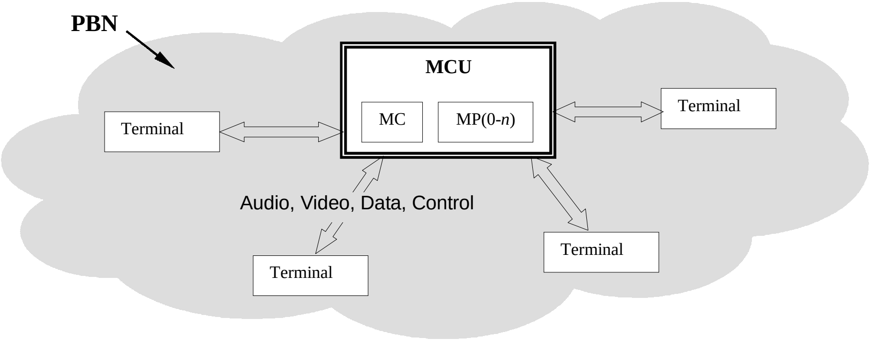

Videoconference archive acquisition
Videoconference archive indexing
Indexed videoconference accessing
Raw videoconference archives
Indexed videoconference archives

Figure 2. A centralized architecture of H.323 videoconference system



tectures, such as the Client/Server architecture for intranet-based videoconferencing and the H.323 architecture for Internet-based videoconferencing. To achieve the best compatibility, this framework should be designed to cooperate with the videoconferencing systems compliant to the most widely adopted ITU-T H.323 Recommendation (ITU-T, 2001), which is quite comprehensive for multimedia communication systems. Nevertheless, this framework can be easily tailored to fit other videoconferencing architectures.

Figure 2 illustrates a typical centralized architecture of H.323 videoconferencing system over a Packet Based Network (PBN). It consists of one Multipoint Control Unit (MCU) and several (at least two) *Terminals*. A terminal is a videoconferencing client in one location. Participants of a videoconference communicate with each other using their local terminals. An MCU contains one Multipoint Controller (MC) and *n* (*n*≥0) Multipoint Processors (MPs). The responsibility of an MC is to coordinate the videoconference, while that of an MP is to process audio/video streams when necessary, such as video switching and audio mixing. The communications between a terminal and the MCU may include audio, video, data, and control channels. A PVAIS, particularly the videoconference archive acquisition module, is only concerned with the terminal.

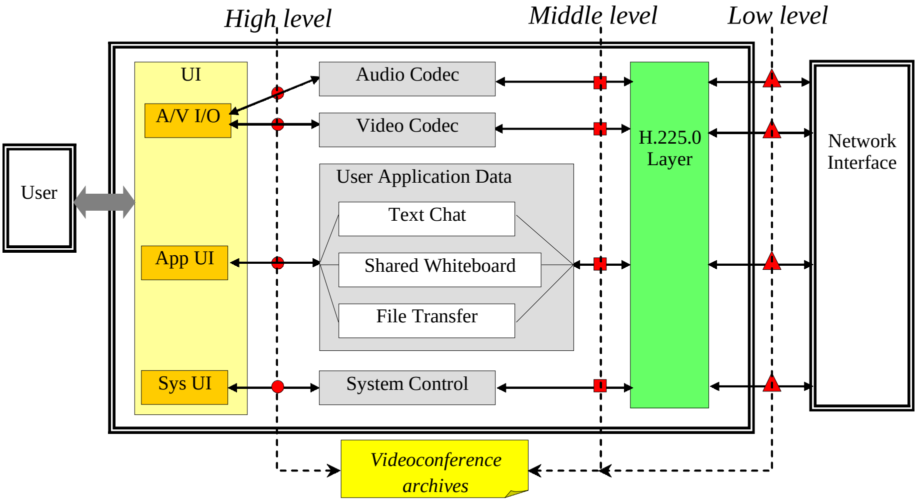
Videoconference Archive Acquisition Module

Since a personal videoconference terminal is pure software with only simple media capture and playback devices attached (like WebCam, microphone, and speaker), the videoconference archive acquisition module should also be pure software to be minimally intrusive. Thus, a thorough understanding of the structure of terminal is necessary.

As shown in Figure 3, the central double-bordered rectangle encloses the elements of a videoconferencing client, which consists of a User Interface (UI) and an H.323 terminal model. The UI part provides the interface for audio/video capture/playback equipments, for user applications, and for system controls. The H.323 terminal model contains audio/video codecs, user data path, system control unit, and H.225.0 layer (ITU-T, 2003a), which is for media stream packetization and synchronization.

There are four types of communication channels: audio, video, data, and control. The audio and video channels transmit incoming/outgoing video and audio information, respectively. The data channel carries information streams for user applications, such

Figure 3. Terminal structure and three interception levels



as text chat, shared whiteboard, and file transfer. The control channel transmits system controls information, including H.245 control (ITU-T, 2003b), call control, and Registration, Admission, and Status (RAS) control. Note that the transportation protocols for these channels are different. Since real-time audio and video transmission are extremely sensitive to delays and jitters, but insensitive to the occasional loss of one or two packets, the reliable Transportation Control Protocol (TCP) is not suitable to transmit audio and video due to the delays introduced during the connection-setup routine and the acknowledgment routines. Therefore, User Datagram Protocol (UDP) together with Real-time Transportation Protocol/Real-time Transportation Control Protocol (RTP/ RTCP) is used for audio and video channels. In contrast to audio and video, data and control information needs very reliable, accurate transmission, but they are not sensitive to a few delays. Thus, TCP is most suitable for data and control channels.

Where to Extract the Information

"Try to introduce the ***least*** delay into the terminal" is an important principle for extracting videoconference content in real time. This principle implies that the extraction process should not be too complex. This section describes three levels of extraction methods of the videoconference archive acquisition module: high-level, middle-level and low-level, as shown in Figure 3. Both the high-level extraction and the middle-level extraction are embedded in the videoconferencing client, whereas the low-level extraction is separated from the client.

The high-level extraction takes place between the UI and the information codecs. The interception points are marked with disks in Figure 3. For the audio channel and the video channel, the extraction will get uncompressed information. For the data channel and the control channel, the extraction can get the semantic operations from the UI directly.

The middle-level extraction is situated before the H.225.0 module — that is, before the information is packetized. The interception points are marked with squares in Figure 3. For the audio channel and the video channel, the extraction can utilize the features of codecs to obtain low-redundancy information, for example, taking I-frames from H.263-encoded video streams as candidate key frames. For the data channel and the control channel, the extraction should be aware of the structure of message stacks to retrieve the information.

The low-level extraction is separated from the videoconferencing client, situated before the network interface. The interception points are marked with triangles in Figure 3. The extraction process runs as a Daemon monitoring the IP transportation ports of the computer. When the communication of the videoconferencing client is detected, the extraction will unpacketize the bitstream to retrieve the information.

Comparing the three levels of extraction methods, we realize that from high-level extraction to low-level extraction, the implementation complexity increases dramatically. However, high complexity does not guarantee the commensurate enhancement to the efficiency. Therefore, to minimize the complexity of the videoconference archive acquisition module, it is recommended that the high-level or middle-level implementation methods is chosen if the existing videoconferencing client could be modified to or be replaced with an information-acquisition-enabled client.

How to Extract the Information

There are two important principles for storing videoconference content into archives:

- Try to reduce the information redundancy as much as possible.
- All recorded events must be labeled with a timestamp.

To index the content of a videoconference, the information in all the four channels should be extracted and stored. These channels could be further divided into logical channels according to the user's point of view, as follows: *video_in*, *video_out*, *audio_in*, *audio_out*, *text_chat*, *whiteboard*, *file_in*, *file_out*, and *control*. Some useful events in each channel are defined in Table 1.

Most of the above events do not take much time to detect except some events in video channels. Since video analysis is usually not fast enough, those events demanding complex analysis (e.g., face detection, slide classification, gesture, and head motion detection) should not be detected in real time. Usually, only the scene change event, resulting in key frames, is detected in real time. Other events will be recovered in the offline indexing by analyzing key frames.

Figure 4 shows the paradigm of storing the extracted information in each channel into the corresponding archive. The extraction processes of all logic channels begin simultaneously when the videoconference starts. The operations in each logic channel are depicted as follows.

The extraction processes for the *Video_in* channel and those for the *Video_out* channel are similar. The *scene change* events are detected in real time. Let *f(t)* denote the function of the video content feature, which varies with time *t*. Thus, the changes of video content will be detected in *f'(t)* as peaks, and the valleys right after each peak can be

Table 1. Event definition for logic channels

Logic Channel	Events
video_in	Scene change, face appeared, slide appeared, gesture made, head movement
video_out	Scene change, face appeared, slide appeared, gesture made, head movement
audio_in	Speech started, silence started, speaker changed
audio_out	Speech started, silence started
text_chat	Text sent, text received, chat type changed
whiteboard	Whiteboard updated
file_in	File sent
file_out	File received
control	Member joined, member left, channel created, channel closed

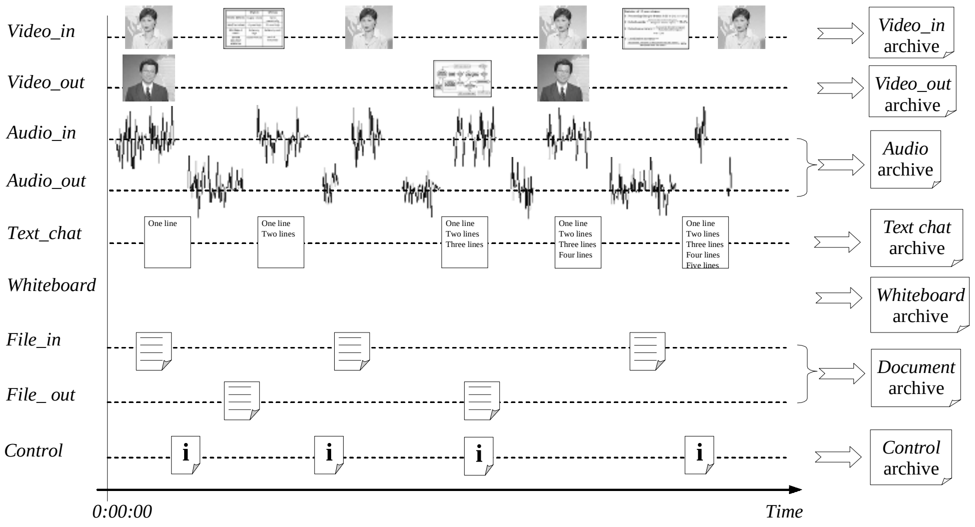
selected as key frames. Note that *f(t)* should consider not only the statistic distribution, but also the spatial distribution of colors to discriminate the change between slides, such as the definition in Dirfaux (2000). The resulting archive — the *Video_in* archive or *Video_out* archive — consists of one text-based index file and a number of key-frame pictures. The index file records the timestamp of each event and the location of the corresponding key frame picture. To preserve the details of the slide pictures, we should employ lossless compression methods, such as TIFF, to store the key-frame pictures.

The audio streams in the *Audio_in* channel and the *Audio_out* channel are first mixed into one stream. Then, silence detection is applied to the mixed stream. Thus, only the speech segments will be stored. The *speaker changed* events are detected in real time by comparing the current vocal feature with the last one. Usually, the vocal feature is modeled by a Gaussian Mixture Model (GMM). The *Audio* archive also includes a text-based index file that records the timestamp for each event and the location of the corresponding audio segment.

The *Text_chat* archive is just a text file containing the timestamp and the corresponding information of each event. For example, for a *text sent* event, the corresponding information includes the sender's username and the textual content. For a *chat type changed* event, the corresponding information is the new chat type, that is, either public chat or private chat.

The *Whiteboard* archive consists of a text-based index file and a number of whiteboard snapshot pictures. The index file records the timestamp for each *whiteboard updated* event and the location of the corresponding snapshot pictures. Since the *Whiteboard* channel contains handwritten texts and graphics, the update of this channel happens not at a point in time but in a period of time. To detect when the update begins and finishes, Song et al. (2003) proposed a method to monitor the *Whiteboard* by comparing sampling images. One can also obtain this information by monitoring the data transfer in this channel. The whiteboard snapshots can be saved as grayscale/binary images with lossless compression methods, such as TIFF/JBIG.

Figure 4. Paradigm of storing the videoconference archives



On each file exchange, the extraction process will copy the sent/received files to the directory storing the videoconference archives. The file exchange information in the *File_in* channel and the *File_out* channel is stored in one *Document* archive, which consists of a text-based index file and a number of exchanged files. The index file records the timestamp and the corresponding information of each event. For instance, for the *file sent* event, the corresponding information includes the recipient's user name and the location of the copy.

The *Control* archive is a text file containing the timestamp and the corresponding information of each event. For a *member joined/left* event, the corresponding information is the involved member's user name. For a *channel created/closed* event, the corresponding information is the involved channel type.

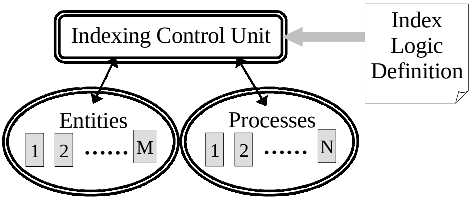
Videoconference Archive Indexing Module

The videoconference archive indexing module is automatically started after the videoconference finishes. This module should be transparent to users.

LEP Indexing Architecture

Since this module integrates many content indexing functionalities, its architecture should be carefully designed to ensure the development flexibility, the runtime stability, and the maintenance convenience. Therefore, the framework of a PVAIS employs an Logic of Entity and Process (LEP) indexing architecture, as shown in Figure 5, which consists of one *Indexing Control Unit*, one *Indexing Logic Definition*, 1~M *Entities*, and 1~N *Processes*.

Figure 5. LEP indexing architecture



Entities include raw videoconference archives produced by the videoconference archive acquisition module and indexed videoconference archives generated by performing indexing functions on raw videoconference archives.

A *Process* implements a content indexing function that takes one or more raw videoconference archives and/or indexed videoconference archives as input and outputs a new indexed archive or updates an existing indexed archive. A process is a stand-alone executable file, separating from the main control and other processes.

The *Indexing Logic Definition* (ILD) describes the function of each process and specifies the input entities and output entities of each process. Therefore, the relationship among all processes and entities — the indexing logic — is defined. The indexing logic determines the priority of each process, that is, the process generating the input entities of the current process should be accomplished before starting the current process. The ILD also defines the format of the resulting Extensible Markup Language (XML) index file.

The *Indexing Control Unit* (ICU) plays the role of main controller in the LEP indexing architecture. It reads the indexing logic from the ILD, checks the availability of each entity, coordinates the execution of each process, and finally generates the resulting XML index file. ICU is responsible for making the whole videoconference archive indexing module work smoothly and automatically. Once an entity become available, it should register to the ICU. When all the input entities of a process become available, the ICU will create a thread to start the process. When a process finishes, no matter whether it succeeds or fails, it will notify the ICU. Moreover, the ICU should check the status of every process from time to time to detect any abrupt termination caused by unexpected exceptions.

Since the ICU and the processes are separately executable files that are loosely coupled by the ILD in the LEP architecture, the *robustness*, *extensibility*, and *maintainability* of the videoconference archive indexing module are greatly enhanced. Therefore, adding, removing, or updating a process only affects the involved process and the ILD.

- *Robustness:* The failure of one function will not lead to the failure of the whole indexing module.
- *Extensibility:* To add new functions implemented as stand-alone processes, one only needs to edit the ILD to link the new processes in.
- *Maintainability:* To merely upgrade one function, just replace the involved process with the upgraded version, without updating the whole indexing module.

Only when the requirement of input entities has been changed is it necessary to edit the ILD. To remove one function, simply modify the ILD to drop this function.

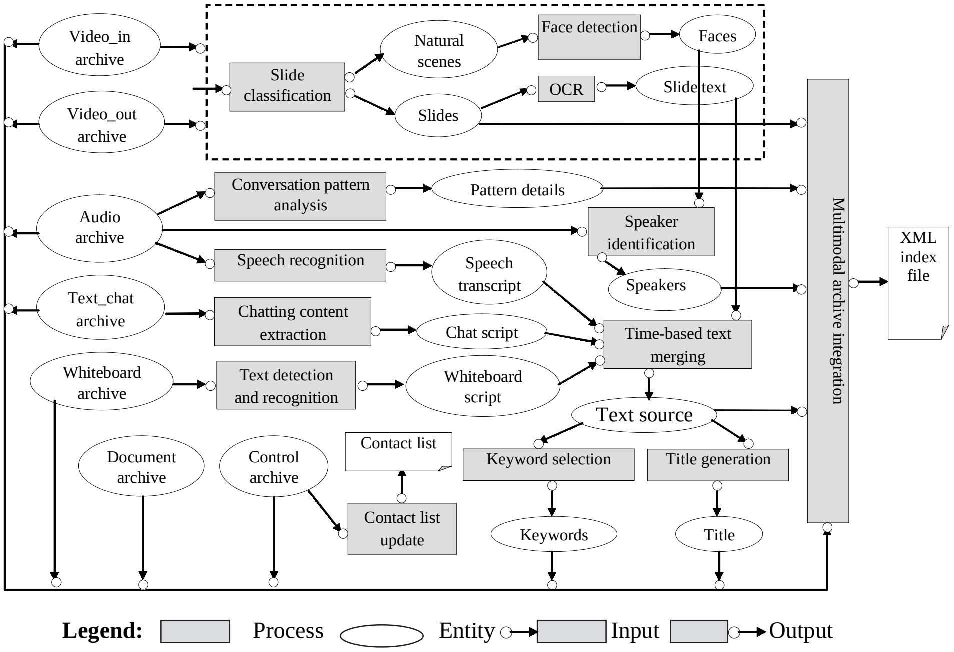
Indexing Logic Definition of a PVAIS

How to define the indexing logic is the core of the videoconference archive indexing module. The ILD of a PVAIS is illustrated in Figure 6, where entities are drawn as white ellipses and processes as gray rectangles. The ILD defines twelve processes totally, which are described in the processing order as follows.

At the very beginning, only the seven raw videoconference archives are available (see Figure 4). There are six processes whose input entities are ready. They are slide classification, conversation pattern analysis, speech recognition, chatting content extraction, text detection and recognition, and contact list update. These five processes can be started simultaneously.

Note that the *Video_in* archive and the *Video_out* archive will go through the same processes enclosed in the top dashed rectangle separately. The slide classification process takes a video archive (either *Video_in* archive or *Video_out* archive) as input, and classifies the key frames in the video archive into two classes: nature scenes and slides. Foote et al. (1999) detected slides by first downsizing the key frame to a 64×64 grayscale representation, then performing the DCT transform on the downsized frame, and finally feeding the 100 principle transform coefficients to a trained diagonal-covariance Gaussian model for the classification. For a simpler implementation, the slide classification can also be accomplished by analyzing both the maximum peak of color

Figure 6. Index logic definition of a PVAIS



histogram and the absolute difference in entropy between horizontal lines in a key frame picture (Leung et al., 2002).

The conversation pattern analysis process takes the *Audio* archive as input, analyzes speeches, silence and *speaker changed* events, and finally divides the whole timeline into segments according to conversation patterns, for example, presentation, discussion, and argument. Kazman et al. (1996) identify three salient measures to classify conversation patterns: (1) who is speaking and for how long, (2) the length of pause, and (3) the degree of overlap between speakers.

The speech recognition process also takes the *Audio* archive as input and produces the *Speech transcript* archive, which composes the majority of text source. Every word in the speech transcript is time-stamped. Since the accuracy of the speech recognizer will determine whether the speech transcript is useful or useless, this process is critical to the system performance. The speech recognizer can be implemented in two ways: (1) employ or improve existing algorithms for large-vocabulary, speaker-independent speech recognition (Barras et al., 2001; Hauptmann, 1995), or (2) utilize commercial speech recognition engines, such as IBM ViaVoice® and Microsoft SpeechAPI®.

The chatting content extraction process extracts the chatting texts and the corresponding time-stamps from the *Text_chat* archive to yield the *Chat script* archive. This process can be easily implemented since the structure of the *Text_chat* archive is known.

The text detection and recognition process takes the *Whiteboard* archive as input, detects text from the whiteboard snapshot pictures, recognizes the text (if any), and stores the recognized text and the corresponding time-stamps into the *Whiteboard script* archive. Since the whiteboard snapshot may contain both graphics and text, the first step is to segment text from graphics. This can be done by a connected-component based method (Fletcher & Kasturi, 1988). Then, an off-line, handwritten text-recognition method (Vinciarelli, Bengio, & Bunke, 2003) is applied to the segmented text image to obtain the text.

The contact list update process checks the *Member joined* events in the *Control* archive. If a member has not yet been recorded in the contact list, this process will add this member into the contact list. Note that the contact list is a global archive corresponding to the whole PVAIS, not belonging to a specific videoconference.

After the slide classification process finishes, its output entities (i.e., natural scenes and slides) become available. Then, the face detection process and the Optical Character Recognition (OCR) process can be started.

The face detection process takes the natural scenes as input and detects faces from them. The detected face regions and their corresponding time-stamps are stored as a new *faces* archive. When the *faces* archive becomes available, the speaker identification process can be started to identify speakers in every time period by integrating aural and visual information (Cutler & Davis, 2000). The implementation of these two processes involves face detection and recognition techniques, which are easily attainable in the literature. Eickeler et al. (2001) detect faces by Neural Network and then recognize faces using pseudo-2-D HMMs and a k-means clustering algorithm. Acosta et al. (2002) proposed a face detection scheme based on a skin detection approach followed by segmentation and region grouping. Their face recognition scheme is based on Principal Component Analysis (PCA). To diminish the limitation that any single face recognition algorithm cannot handle various cases well, Tang, Lyu, and King (2003) presented a face recognition committee machine that assembles the outputs of various face recognition algorithms to obtain a unified decision with improved accuracy.

The OCR process takes the *Slides* archive as input, binarizes the slide pictures, and recognizes text from them. The recognized text is stored as a *Slide text* archive. The major task of this process is to identify text regions in the slide pictures. Cai et al. (2002) proposed a robust approach to detect and localize text on complex background in video images. This algorithm utilizes invariant edge features to detect and enhance text areas, and then localizes text strings with various spatial layouts by a coarse-to-fine localization scheme.

Once the *Speech transcript* archive, the *Chat script* archive, the *Whiteboard script* archive, and the *Slide text* archive are all ready, the time-based text merging process will be started to merge these four archives into one *Text source* archive according to their associated time-stamp information. Therefore, the *Text source* archive integrates all the textual information obtainable from videoconference archives. When the *Text source* archive becomes available, the keyword selection process and the title generation process will be started.

The keyword selection process takes the *Text source* archive as input and produces keywords on two levels: global and local. The global keywords are selected in the scope of the whole videoconference, representing the overall subject of the conference. On the other hand, the local keywords are clustered in a limited time period; therefore, they only indicate the topic of this period. Providing two-level keywords enhances the flexibility in supporting the content-based retrieval. Global keywords enable the quick response when searching videoconferences, while local keywords are more powerful in seeking a point of interest in a videoconference. The Neural Network-based text clustering algorithm (Lagus & Kaski, 1999) can be employed to select both global keywords and local keywords.

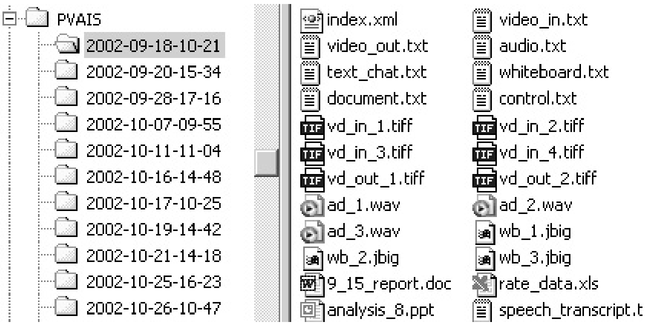
The title generation process also takes the *Text source* archive as input and generates a title for this videoconference by employing language-processing techniques. Jin and Hauptmann (2001) proposed a novel approach for title word selection. They treat this task as a variant of an Information Retrieval problem. A good representation vector for title words is determined by minimizing the difference between human-assigned titles and machine-generated titles over the training examples.

In addition to employing existing algorithms, one may also need to develop some specific multimedia processing techniques. In this case, one can find the fundamentals of digital image processing, digital video processing, and speech analysis in the books of Castleman (1996), Tekalp (1995), and Rabiner and Juang (1993), respectively.

Finally, when all the entities are ready, the multimodal archive integration process is started. It takes thirteen archives as input to construct an XML index file, which structurally organizes all index information from these archives and serves as an interface to the search engine.

All indexed videoconferences are stored in the home directory of a PVAIS. The indexed archives of the same videoconference are stored in the same subdirectory of the home directory, as shown in Figure 7. Using the starting date and time of the videoconference as the directory name is a good way to avoid conflicts.

Figure 7. Storage structure of PVAIS



PVAIS
- 2002-09-18-10-21
- 2002-09-20-15-34
- 2002-09-28-17-16
- 2002-10-07-09-55
- 2002-10-11-11-04
- 2002-10-16-14-48
- 2002-10-17-10-25
- 2002-10-19-14-42
- 2002-10-21-14-18
- 2002-10-25-16-23
- 2002-10-26-10-47

index.xml	video_in.txt
video_out.txt	audio.txt
text_chat.txt	whiteboard.txt
document.txt	control.txt
vd_in_1.tiff	vd_in_2.tiff
vd_in_3.tiff	vd_in_4.tiff
vd_out_1.tiff	vd_out_2.tiff
ad_1.wav	ad_2.wav
ad_3.wav	wb_1.jbig
wb_2.jbig	wb_3.jbig
9_15_report.doc	rate_data.xls
analysis_8.ppt	speech_transcript.t

Indexed Videoconference Accessing Module

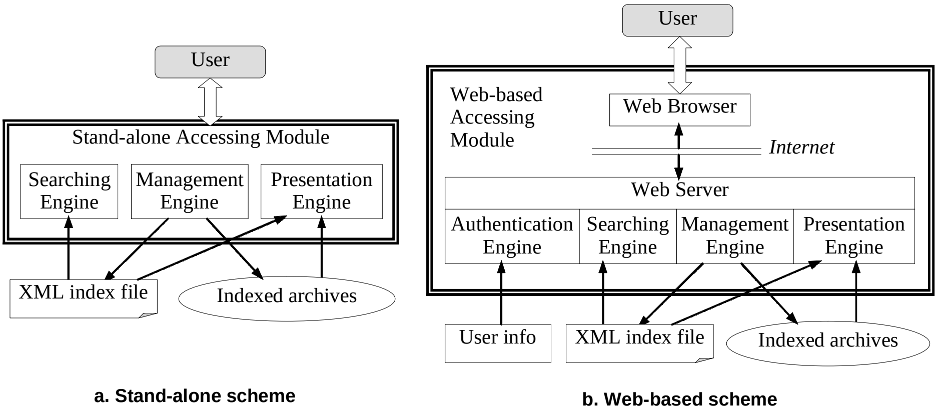
The indexed videoconference accessing module provides a user with an interface to manage, search, and review all indexed conferences. The search function allows the user to search the videoconference of interest by a variety of criteria. The management function can be divided into two levels: conference-oriented management and content-oriented management. Conference-oriented management functions apply to the whole conference, such as classifying the type of a conference (e.g., private or business) or deleting a conference. Content-oriented management functions only apply to specific content of a conference, such as editing keywords or title. The review function supports the synchronized presentation of a videoconference. During the presentation, it also enables the user to pause the presentation and attach annotations or bookmarks to the current time-stamp.

There are two types of implementation schemes for this module: the stand-alone scheme and the Web-based scheme, as shown in Figure 8. The former restricts the user to accessing the indexed videoconferences from the computer in which the indexed videoconferences are stored, while the latter allows the user to access the indexed videoconferences from any computer via the Internet.

According to the stand-alone scheme (Figure 8a), this module is implemented as a stand-alone application that integrates three engines: a searching engine, a management engine, and a presentation engine. These three engines handle the user's requests and return the results to the user. The searching engine searches the index files of all videoconferences for the content specified by the user, and then returns the videoconferences satisfying the user's searching criteria. The management engine maintains the indexed videoconferences according to the user's command. It will affect both the XML index file and the involved indexed archives. When the user selects a videoconference to review, the presentation engine reads synchronization information from the XML index file to control the display of multimedia information stored in the indexed archives.

In the Web-based scheme (Figure 8b), the user interface and the functionality engines are separated. The Web server is situated in the same computer in which the indexed videoconferences are stored. The user can access the indexed videoconferences

Figure 8. Two implementation schemes of the indexed videoconference accessing module



a. Stand-alone scheme

b. Web-based scheme

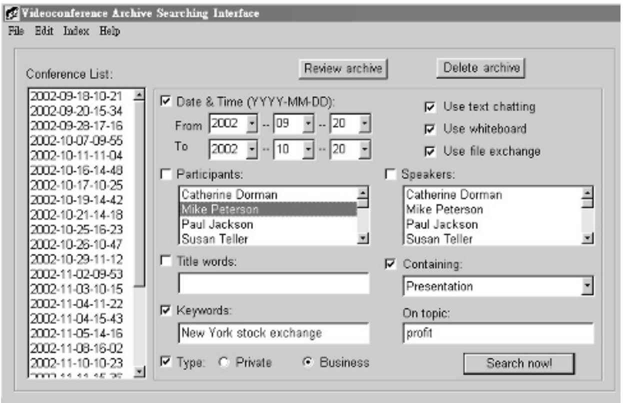
through any ordinary Web browser. The functionality engines are coupled with the Web server. In addition to the three engines discussed in the stand-alone scheme, the Web-based scheme requires an authentication engine to verify the user's identity because a PVAIS is a personal system. In this scheme, Synchronized Multimedia Integration Language (SMIL) can be employed to provide synchronized multimedia presentation in the Web browser.

The choice between the stand-alone scheme and the Web-based scheme depends on the user's habit. For those users who do videoconferencing on laptops and always take their laptops with them, the stand-alone scheme is suitable. For other users, especially those who wish to share their indexed videoconferences, the Web-based scheme is better. From the development point of view, the Web-based scheme is more advanced at the cost of demanding more complex development efforts. In fact, the Web-based scheme contains the stand-alone scheme; therefore, one can first implement a stand-alone accessing module as a prototype, and then extend it to the Web-based accessing module.

We built a prototype system of a PVAIS whose indexed videoconference accessing module is based on the stand-alone scheme. The searching interface and the review interface are shown in Figure 9 and Figure 10, respectively.

The searching interface (Figure 9) is the initial interface of the stand-alone application. Initially, the conference list contains all indexed videoconferences stored in the home directory of a PVAIS. The list of participants and the list of speakers are both loaded from the contact list. Other items are the searching criteria to be set by the user. For example, the user wants to find a business videoconference about "New York stock exchange." He only remembers the approximate date of the videoconference, but forgets who participated in the videoconference. However, he is sure that someone gave a presentation on "Profit" in the videoconference, and that speaker used text chat, whiteboard, and file exchange to communicate. Therefore, the user sets his searching criteria as shown in Figure 9. After the "Search now!" button is pressed, the searching engine will search the XML index files of all indexed videoconferences and then update

Figure 9. Searching interface of the accessing module of PVAIS



Conference List:
2002-09-18-10-21
2002-09-20-15-34
2002-09-28-17-16
2002-10-07-09-55
2002-10-11-11-04
2002-10-16-14-48
2002-10-17-10-25
2002-10-19-14-42
2002-10-21-14-18
2002-10-25-16-23
2002-10-26-10-47
2002-10-29-11-12
2002-11-02-09-53
2002-11-03-10-15
2002-11-04-11-22
2002-11-04-15-43
2002-11-05-14-16
2002-11-08-16-02
2002-11-10-10-23

☑ Date & Time (YYYY-MM-DD):
From 2002 -- 09 -- 20
To 2002 -- 10 -- 20

☑ Use text chatting
☑ Use whiteboard
☑ Use file exchange

☐ Participants: Catherine Dorman, Mike Peterson, Paul Jackson, Susan Teller

☐ Speakers: Catherine Dorman, Mike Peterson, Paul Jackson, Susan Teller

☐ Title words:

☑ Containing: Presentation

☑ Keywords: New York stock exchange

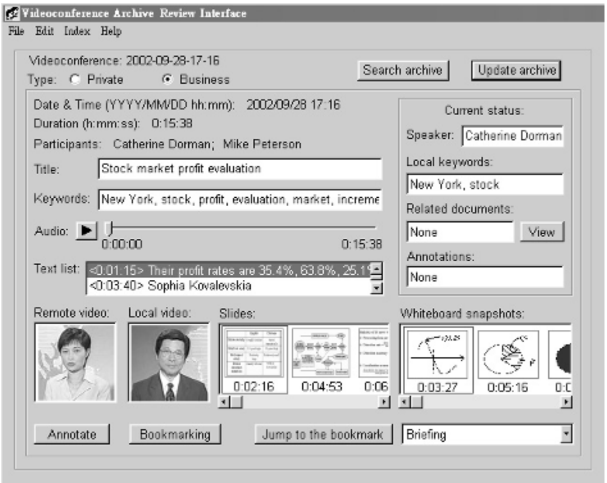
On topic: profit

☑ Type: ○ Private ● Business

the conference list with those satisfying the searching criteria. The user may select a videoconference from the list and press the “Review archive” button to review the videoconference (Figure 10). The other button, “Delete archive,” is used to delete a selected videoconference.

The review interface (Figure 10) provides the management functions and the synchronized presentation of an indexed videoconference. The user can set the conference type or edit those textual indexing items, such as title, keywords, and speaker name. On pressing the “Update archive” button, the updated information will be saved to the XML index file and archives. This interface displays rich information (both static and dynamic) of a videoconference. The static information includes date and time, duration, participants, title, and global keywords. The dynamic information is presented during the synchronized playback. When the user presses the ▶ button to play the audio archive

Figure 10. Review interface of the accessing module of PVAIS



Videoconference: 2002-09-28-17-16
Type: ○ Private ● Business
Date & Time (YYYY/MM/DD hh:mm): 2002/09/28 17:16
Duration (h:mm:ss): 0:15:38
Participants: Catherine Dorman; Mike Peterson
Title: Stock market profit evaluation
Keywords: New York, stock, profit, evaluation, market, increme
Audio: ▶ 0:00:00 0:15:38
Text list: <0:01:15> Their profit rates are 35.4%, 63.8%, 25.1
<0:03:40> Sophia Kovalevskia
Speaker: Catherine Dorman
Local keywords: New York, stock
Related documents: None
Annotations: None
Briefing

or drags the slide of audio timeline, the contents in the "Current status" frame, "Text list," "Remote video," "Local video," "Slides," and "Whiteboard snapshots" will be automatically scrolled and highlighted according to the current time-stamp. The user may also change the key frame in the "Remote video" and "Local video" windows or change the highlighted contents in the text list, the slide list, or the whiteboard snapshot list to change the current time-stamp. At any time during the playback, the user can press the "Annotation" button to write comments or press the "Bookmarking" button to insert a bookmark associated with the current time-stamp. These two operations will automatically pause the playback until the operations are finished. The existing annotations are shown in the "Current status" frame. The "Jump to the bookmark" button allows the user to quickly change to the time-stamp associated with the selected bookmark. The annotation and bookmark information will be saved after pressing the "Update archive" button. The function of the "Search archive" button is to finish the review and go back to the searching interface.

SUBJECTIVE EVALUATION PROTOCOL

This section discusses how to evaluate the performance of a videoconference indexing system. Since the ultimate goals of such systems are to augment people's memory and to accelerate the content-based searching, the evaluation should be conducted in terms of *recall capability* and *search capability*. The recall capability demonstrates how the system helps the user to recall the information in the videoconference, from the overall level to the detailed level. The search capability indicates how the system supports various means to let the user locate the content of interest quickly and correctly. Since it is still difficult to develop an objective evaluation protocol to represent these high-level criteria, this section defines a subjective evaluation protocol. Table 2 shows the aspects evaluated by this protocol. The upper limit of scoring for each aspect represents the weight of this aspect in its related capability.

The evaluation of recall capability considers eight aspects, as shown in the left half of Table 2. The "Subject" aspect checks whether the title and global keywords outline the videoconference. The "Details" aspect further examines whether the important details can be retrieved from the archives. The "Participant" aspect focuses on the

Table 2. Aspects of subjective evaluation

Recall capability			**Search capability**		
Aspect	**Symbol**	**Scoring**	**Aspect**	**Symbol**	**Scoring**
Subject	R_1	0 ~ 6	Time	S_1	0 ~ 6
Details	R_2	0 ~ 10	Topic	S_2	0 ~ 10
Participant	R_3	0 ~ 6	Participant	S_3	0 ~ 6
Key frame accuracy	R_4	0 ~ 8	Visual pattern	S_4	0 ~ 8
Speech fidelity	R_5	0 ~ 8	Aural pattern	S_5	0 ~ 8
Supporting tools	R_6	0 ~ 6	Textual pattern	S_6	0 ~ 10
Presentation	R_7	0 ~ 10	Conference event	S_7	0 ~ 6
Extensibility	R_8	0 ~ 6	Nonlinear access	S_8	0 ~ 10
Overall	$R = \Sigma R_i$	0 ~ 60	**Overall**	$S = \Sigma S_i$	0 ~ 64

completeness of participant's information, including joining or leaving. The "Key frame accuracy" aspect and the "Speech fidelity" aspect evaluate the effect of removing the redundancy in video and audio streams. The "Supporting tools" aspect checks whether the content in other communication tools, for example, text chat, whiteboard, and file exchange, is well indexed. The "Presentation" aspect pays attention to the presentation modes of all types of media and the capability of synchronized presentation. The "Extensibility" aspect checks whether the recall capability can be extended by user's interactions, such as editing, annotation, and bookmarking. The overall recall capability (*R*) sums up the scores of the above eight aspects.

The evaluation of search capability also considers eight aspects, as shown in the right half of Table 2. For the top seven aspects, we evaluate how the system supports searching by these aspects. For the "Nonlinear access" aspect, we check whether the system can automatically locate the point of interest in the searching result according to the user's searching criteria. The overall searching capability (*S*) sums up the scores of the above eight aspects.

After obtaining the scores for all evaluation aspects, the performance (denoted by *P*) of a videoconference indexing system is calculated as follows:

$$P = \alpha \cdot \frac{R}{60} + (1-\alpha) \cdot \frac{S}{64}$$

Where α is to adjust the weights of the recall capability and the searching capability. The default value of α is 0.5. Thus, *P* ranges from 0 to 1. The higher *P* is, the better performance the videoconference indexing system achieves.

CONCLUSIONS

This chapter focuses on the videoconference archive indexing, which bears different characteristics from existing research on video indexing, lecture indexing, and meeting support systems. We proposed a comprehensive personal videoconference indexing framework, i.e., PVAIS that consists of three modules: videoconference archive acquisition module, videoconference archive indexing module, and indexed videoconference accessing module. This chapter elaborated the design principles and implementation methodologies of each module, as well as the intra- and inter-module data and control flows. Based on the PVAIS framework, one can easily develop a videoconference indexing system according to his/her specific requirements. Finally, this chapter presented a subjective evaluation protocol for personal videoconference indexing.

ACKNOWLEDGMENT

The work described in this chapter was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4182/03E).

REFERENCES

Acosta, E., Torres, L., Albiol, A., & Delp, E. (2002). An automatic face detection and recognition system for video indexing applications. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, Vol. 4, IV-3644-IV-3647.

Albiol, A., Torres, L., & Delp, E.J. (2002). Video preprocessing for audiovisual indexing. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, Vol.4, IV-3636 -IV-3639.

Ardizzo, E., La Cascia, M., Di Gesu, V., & Valenti, C. (1996). Content-based indexing of image and video databases by global and shape features. In *Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96)*, Vol.3, 140-144.

Ardizzone, E., & La Cascia, M. (1996). Video indexing using optical flow field. In *Proceedings of the International. Conference on Image Processing (ICIP'96)*, Vol. 3, 831-834.

Ariki, Y., Suiyama, Y., & Ishikawa, N. (1998). Face indexing on video data-extraction, recognition, tracking and modeling. In *Proceedings of the Third IEEE International. Conference on Automatic Face and Gesture Recognition*, 62-69.

Barras, C., Lamel, L., & Gauvain, J.-L. (2001). Automatic transcription of compressed broadcast audio. In *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Vol. 1, 265-268.

Ben-Arie, J., Pandit, P., & Rajaram, S. (2001). View-based human activity recognition by indexing and sequencing. In *Proceedings of the International. Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Vol. 2, II-78 -II-83.

Bescos, J., Menendez, J.M., Cisneros, G., Cabrera, J., & Martinez, J.M. (2000). A unified approach to gradual shot transition detection. In *Proceedings of the Intl. Conf. on Image Processing (ICIP'00)*, Vol. 3, 949-952.

Cai, M., Song, J., & Lyu, M.R. (2002). A new approach for video text detection. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'02)*, Vol. 1, 117-120.

Calic, J., & Lzquierdo, E. (2002). A multiresolution technique for video indexing and retrieval. *Proceedings of the International Conference on Image Processing (ICIP'02)*, Vol. 1, 952-955.

Castleman, K.R. (1996). *Digital image processing*. Englewood Cliffs, NJ: Prentice Hall.

Chan, Y., Lin, S.-H., Tan, Y.-P., & Kung, S.Y. (1996). Video shot classification using human faces. In *Proceedings of the International. Conference on Image Processing (ICIP'96)*, Vol. 3, 843-846.

Chang, S.-F. (1995). Compressed-domain techniques for image/video indexing and manipulation. In *Proceedings of the International. Conference on Image Processing (ICIP'95)*, Vol. 1, 314-317.

Chang, Y.-L., Zeng, W., Kamel, I., & Alonso, R. (1996). Integrated image and speech analysis for content-based video indexing. In *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems (ICMCS'96)*, 306-313.

Chiu, P., Boreczky, J., Girgensohn, A., & Kimber, D. (2001). LiteMinutes: An Internet-based system for multimedia meeting minutes. In *Proceedings of the 10th World Wide Web Conference*, 140-149.

Chiu, P., Kapuskar, A., Reitmeier, S., & Wilcox, L. (1999). NoteLook: Taking notes in meeting with digital video and ink. In *Proceedings of the ACM International Conference on Multimedia*, Vol. 1, 149-158.

Chiu, P., Kapuskar, A., Reitmeier, S., & Wilcox, L. (2000). Room with a rear view. Meeting capture in a multimedia conference room. *IEEE Multimedia*, *7*(4), 48-54.

Christel, M., Kanade, T., Mauldin, M., Reddy, R., Stevens, S., & Wactlar, H. (1996). Techniques for the creation and exploration of digital video libraries. In Chapter 8 of B. Furht (Ed.), *Multimedia Tools and Applications (Vol. 2)*. Boston, MA: Kluwer Academic Publishers.

Corridoni, J.M., & Del Bimbo, A. (1996). Structured digital video indexing. In *Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96)*, Vol.3, 125-129.

Cutler, R., & Davis, L. (2000). Look who's talking: Speaker detection using video and audio correlation. In *Proceedings of the International. Conference on Multimedia and Expo (ICME'00)*, Vol. 3, 1589-1592.

Dagtas, S., Al-Khatib, W., Ghafoor, A., & Khokhar, A. (1999). Trail-based approach for video data indexing and retrieval. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Vol. 2, 235-239.

Del Bimbo, A. (2000). Expressive semantics for automatic annotation and retrieval of video streams. In *Proceedings of the International Conference on Multimedia and Expo (ICME'00)*, Vol. 2, 671-674.

Deshpande, S.G., & Hwang, J.-N. (2001). A real-time interactive virtual classroom multimedia distance learning system. *IEEE Trans. on Multimedia*, *3*(4), 432-444.

Dharanipragada, S., & Roukos, S. (1996). A fast vocabulary independent algorithm for spotting words in speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'96)*, Vol. 1, 233-236.

Diklic, D., Petkovic, D., & Danielson, R. (1998). Automatic extraction of representative key frames based on scene content. In *Proceedings of the 32nd Asilomar Conference on Signals, Systems & Computers*, Vol. 1, 877-881.

Di Lecce, V., Dimauro, G., Guerriero, A., Impedovo, S., Pirlo, G., & Salzo, A. (1999). Image basic features indexing techniques for video skimming. In *Proceedings of the International Conference on Image Analysis and Processing*, 715-720.

Dirfaux, F. (2000). Key frame selection to represent a video. In *Proceedings of the International Conference on Image Processing (ICIP'00)*, Vol. 2, 275-278.

Doulamis, N.D., Doulamis, A.D., Avrithis, Y.S., & Kollias, S.D. (1998). Video content representation using optimal extraction of frames and scenes. In *Proceedings of the International Conference on Image Processing (ICIP'98)*, Vol. 1, 875-879.

Eickeler, S., Wallhoff, F., Lurgel, U., & Rigoll, G. (2001). Content based indexing of images and video using face detection and recognition methods. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Vol. 3, 1505-1508.

Fletcher, L.A., & Kasturi, R. (1988). A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *10*(6), 910-918.

Foote, J., Boreczsky, J., & Wilcox, L. (1999). Finding presentations in recorded meetings using audio and video features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99)*, Vol. 6, 3029-3032.

Gao, Q., Ko, C.C., & De Silva, L.C. (2000) A universal scheme for content-based video representation and indexing. In *Proceedings of the 2000 IEEE Asia-Pacific Conference on Circuits and Systems*, 469-472.

Gargi, U., Antani, S., & Kasturi, R. (1998). VADIS: A Video Analysis, Display and Indexing System. In *Proc. Intl. Conf. on Computer Vision and Pattern Recognition (CVPR'98)*, 965-965.

Gelin, P., & Wellekens, C.J. (1996). Keyword spotting enhancement for video soundtrack indexing. In *Proceedings of the 4th International Conference on Spoken Language*, Vol. 2, 586-589.

Geyer, W., Richter, H., & Abowd, G.D. (2003). Making multimedia meeting records more meaningful. In *Proceedings of the International Conference on Multimedia and Expo (ICME'03)*, Vol. 2, 669-672.

Ginsberg, A., & Ahuja, S. (1995). Automating envisionment of virtual meeting room histories. In *Proceedings of the ACM International. Conference on Multimedia*, 65-75.

Gross, R., Bett, M., Yu, H., Zhu, X., Pan, Y., Yang, J., & Waibel, A. (2000). Towards a multimodal meeting record. In *Proceedings of the International. Conference on Multimedia and Expo (ICME'00)*, Vol. 3, 1593-1596.

Gu, L., & Bone, D. (1999). Skin colour region detection in MPEG video sequences. In *Proceedings of the International Conference on Image Analysis and Processing*, 898-903.

Hauptmann, A.G. (1995). Speech recognition in the Informedia Digital Video Library: Uses and limitations. In *Proceedings of the 7th International Conference on Tools with Artificial Intelligence*, 288-294.

Hauptmann, A.G., & Wactlar, H.D. (1997). Indexing and search of multimodal information. In *Proceedings of the IEEE International. Conference on Acoustics, Speech, and Signal Processing (ICASSP'97)*, Vol. 1, 195-198.

Hidalgo, J.R., & Salembier, P. (2001). Robust segmentation and representation of foreground key regions in video sequences. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Vol. 3, 1565-1568.

Hsu, C.-T., & Teng, S.-J. (2002). Motion trajectory based video indexing and retrieval. In *Proceedings of the International Conference on Image Processing (ICIP'00)*, Vol. 1, 605-608.

Hua, X.-S., Yin, P., & Zhang, H.-J. (2002). Efficient video text recognition using multiple frame integration. In *Proceedings of the International Conference on Image Processing (ICIP'02)*, Vol. 2, II-397 -II-400.

Hwang, J.-N., & Luo, Y. (2002). Automatic object-based video analysis and interpretation: A step toward systematic video understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, Vol. 4, IV-4084 -IV-4087.

Ito, H., Sato, M., & Fukumura, T. (2000). Annotation and indexing in the video management system (VOM). In *Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 2, 834-839.

ITU-T Recommendation H.225.0 (2003a). Call signaling protocols and media stream packetization for packet-based multimedia communication systems. Retrieved July 2003: *http://www.itu.int/rec/recommendation.asp*

ITU-T Recommendation H.245 (2003b). Control protocol for multimedia communication. Retrieved August 2001: *http://www.itu.int/rec/recommendation.asp*

ITU-T Recommendation H.323 Draft v4. (2001). Packet-based multimedia communications systems. Retrieved February 2001: *http://www.itu.int/rec/recommendation.asp*

Iyengar, G., Nock, H., Neti, C., & Franz, M. (2002) Semantic indexing of multimedia using audio, text and visual cues. In *Proceedings of the International Conference on Multimedia and Expo (ICME'02)*, Vol. 2, 369-372.

Jain, A.K., & Yu, B. (1998). Automatic text location in images and video frames. In *Proceedings of the 14th International Conference on Pattern Recognition (ICPR'98)*, Vol. 2, 1497-1499.

Jasinschi, R.S., Dimitrova, N., McGee, T., Agnihotri, L., & Zimmerman, J. (2001a). Video scouting: An architecture and system for the integration of multimedia information in personal TV applications. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Vol. 3, 1405-1408.

Jasinschi, R.S., Dimitrova, N., McGee, T., Agnihotri, L., Zimmerman, J., & Li, D. (2001b). Integrated multimedia processing for topic segmentation and classification. In *Proceedings of the International Conference on Image Processing (ICIP'01)*, Vol. 3, 366-369.

Jasinschi, R.S., Dimitrova, N., McGee, T., Agnihotri, L., Zimmerman, J., Li, D., & Louie, J. (2002). A probabilistic layered framework for integrating multimedia content and context information. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, Vol. 2, 2057-2060.

Jin, R., & Hauptmann, A. (2001). Learning to select good title words: A new approach based on reversed information retrieval. In *Proceedings of the International. Conference on Machine Learning (ICML'01)*, 242-249.

Joukov, N., & Chiueh T.-C. (2003). Lectern II: A multimedia lecture capturing and editing system. In *Proceedings of the International Conference on Multimedia and Expo (ICME'03)*, Vol. 2, 681-684.

Ju, S.X., Black, M.J., Minneman, S., & Kimber, D. (1997). Analysis of gesture and action in technical talks for video indexing. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 595-601.

Kameda, Y., Nishiguchi, S., & Minoh, M. (2003). Carmul: Concurrent automatic recording for multimedia lecture. In *Proceedings of the International. Conference on Multimedia and Expo (ICME'03)*, Vol. 2, 677-680.

Kang, E.K., Kim, S.J., & Choi, J.S. (1999). Video retrieval based on key frame extraction in compressed domain. In *Proceedings of the International Conference on Image Processing (ICIP'99)*, Vol. 3, 260-264.

Kang, J., & Mersereau, R.M. (2002). An effective method for video segmentation and sub-shot characterization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, Vol. 4, IV-3652-IV-3655.

Kazman, R., Al-Halimi, R., Hunt, W., & Mantei, M. (1996), Four paradigms for indexing video conferences. *IEEE Multimedia*, *3*(1), 63 -73.

Kazman, R., & Kominek, J. (1999). Supporting the retrieval process in multimedia information systems. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*, Vol. 6, 229-238.

Kim, E.Y., Kim, K.I., Jung, K., & Kim, H.J. (2000). A video indexing system using character recognition. In *Digest of Technical Papers of International Conference on Consumer Electronics*, 358-359.

Kim, S.H., & Park, R.-H. (2000). A novel approach to scene change detection using a cross entropy. In *Proceedings of the International Conference on Image Processing (ICIP'00)*, Vol. 3, 937-940.

Kristjansson, T., Huang, T.S., Ramesh, P., & Juang, B.H. (1999). A unified structure-based framework for indexing and gisting of meetings. In *Proceedings of the International Conference on Multimedia Computing and Systems*, Vol. 2, 572-577.

Lagus, K., & Kaski, S. (1999). Keyword selection method for characterizing text document maps. In *Proceedings of the International Conference on Artificial Neural Networks*, Vol. 1, 371-376.

Lebourgeois, F., Jolion, J.-M., & Awart, P.C. (1998). Towards a description for video indexation. In *Proceedings of the 14th Intl. Conf. on Pattern Recognition (ICPR'98)*, Vol. 1, 912-915.

Leung, W.H., Chen, T., Hendriks, F., Wang, X., & Shae, Z.-Y. (2002). eMeeting: A multimedia application for interactive meeting and seminar. In *Proceedings of the Global Telecommunications Conference*, Vol. 3, 2994-2998.

Li, C.-S., Mohan, R., & Smith, J.R. (1998). Multimedia content description in the InfoPyramid. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'98)*, Vol. 6, 3789-3792.

Li, H., & Doermann, D. (1998). Automatic identification of text in digital video key frames. In *Proceedings of the 14th International Conference on Pattern Recognition (ICPR'98)*, Vol. 1, 129-132.

Liang, R.Z., Venkatesh, S., & Kieronska, D. (1995). Video indexing by spatial representation. In *Proceedings of the 3rd Australian and New Zealand Conference on Intelligent Information Systems (ANZIIS'95)*, 99-104.

Luo, Y., & Hwang, J.N. (2003). Video sequence modeling by dynamic Bayesian networks: A systematic approach from fine-to-coarse grains. In *Proceedings of the International Conference on Image Processing (ICIP'03)*, II615-II618, Barcelona, Spain, September.

Lyu, M.R., Yau, E., & Sze, K.S. (2002). A multilingual, multimodal digital video library system. In *Proceedings of the Joint Conference on Digital Libraries (JCDL 2002)*, 145-153.

Madrane, N., & Goldberg, M. (1994). Towards automatic annotation of video documents. In *Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing*, Vol. 1, 773-776.

Maziere, M., Chassaing, F., Garrido, L., & Salembier, P. (2000). Segmentation and tracking of video objects for a content-based video indexing context. In *Proceedings of the International. Conference on Multimedia and Expo (ICME'00)*, Vol. 2, 1191-1194.

Mikolajczyk, K., Choudhury, R., & Schmid, C. (2001). Face detection in a video sequence - A temporal approach. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Vol. 2, II-96-II-101.

Nam, J., Cetin, E., & Tewfik, A.H. (1997). Speaker identification and video analysis for hierarchical video shot classification. In *Proceedings of the International Conference on Image Processing (ICIP'97)*, Vol. 2, 550-553.

Naphade, M.R., & Huang, T.S. (2000). Inferring semantic concepts for video indexing and retrieval. In *Proceedings of the International Conference on Image Processing (ICIP'00)*, Vol. 3, 766-769.

Naphade, M.R., Kristjansson, T., Frey, B., & Huang, T.S. (1998). Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems. In *Proceedings of the International Conference on Image Processing (ICIP'98)*, Vol. 3, 536-540.

Ngo, C.-W., Pong, T.-C., & Huang, T.S. (2002). Detection of slide transition for topic indexing. In *Proceedings of the International Conference on Multimedia and Expo (ICME'02)*, Vol. 2, 533-536.

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. Englewood Cliffs, NJ: Prentice-Hall.

Rogina, I., & Schaaf, T. (2002) Lecture and presentation tracking in an intelligent meeting room. In *Proceedings of the 4th Intl. Conf. on Multimodal Interfaces*, 47-52.

Sato, S., & Kanade, T. (1997). NAME-IT: Association of face and name in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 368-373.

Sawhney, H.S. (1993). Motion video annotation and analysis: An overview. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, Vol. 1, 85-89.

Song, J., Lyu, M., Hwang, J.-N., & Cai, M. (2003). PVCAIS: A personal videoconference archive indexing system. In *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME'03)*, 117-120.

Sprey, J.A. (1997). Videoconferencing as a communication tool. *IEEE Trans. on Professional Communication*, *40*(1), 41-47.

Stewart, A., Wolf, P., & Heminje, M. (2003). Media and metadata management for capture and access systems in electronic lecturing environments. In *Proceedings of the International. Conference on Multimedia and Expo (ICME'03)*, Vol. 2, 685-688.

Sun, M.T., Wu, T.-D., & Hwang, J.-N. (1998). Dynamic bit allocation in video combining for multipoint conferencing. *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, *45*(5), 644-648.

Tang, H.-M., Lyu, M.R., & King, I. (2003). Face recognition committee machine. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, 837-840.

Tekalp, A.M. (1995). *Digital video processing*. Upper Saddle River, NJ: Prentice Hall.

Tsapatsoulis, N., Avrithis, Y., & Kollias, S. (2000). Efficient face detection for multimedia applications. In *Proceedings of the International Conference on Image Processing (ICIP'00)*, Vol. 2, 247-250.

Tse, K., Wei, J., & Panchanathan, S. (1995). A scene change detection algorithm for MPEG compressed video sequences. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, Vol. 2, 827-830.

Tsekeridou, S., & Pitas, I. (1998). Speaker dependent video indexing based on audio-visual interaction. In *Proceedings of the International Conference on Image Processing (ICIP'98)*, Vol. 1, 358-362.

Vinciarelli, A., Bengio, S., & Bunke, H. (2003). Off-line recognition of large vocabulary cursive handwritten text. In *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR'03)*, 1101-1105.

Viswanathan, M., Beigi, H.S.M., Tritschler, A., & Maali, F. (2000). Information access using speech, speaker and face recognition. In *Proceedings of the International Conference on Multimedia and Expo (ICME'00)*, Vol. 1, 493-496.

Wactlar, H., Kanade, T., Smith, M., & Stevens, S. (1996). Intelligent access to digital video: The Informedia Project. *IEEE Computer: Digital Library Initiative special issue, 29*(5), 46-52.

Wang, P., Ma, Y.-F., Zhang, H.-J., & Yang, S. (2003). A people similarity based approach to video indexing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Vol. 3, III-693-III-696.

Wei, J., Li, Z.-N., & Gertner, I. (1999). A novel motion-based active video indexing method. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Vol. 2, 60-465.

Wilcox, L., & Boreczky, J. (1998). Annotation and segmentation for multimedia indexing and retrieval. In *Proceedings of the 31st International Conference on System Sciences*, Vol. 2, 259-266.

Wilcox, L., Chen, F., Kimber, D., & Balasubramanian, V. (1994). Segmentation of speech using speaker identification. In *Proceedings of the IEEE International. Conference on Acoustics, Speech, and Signal Processing (ICASSP'94)*, Vol. 1, 161-164.

Young, S.J., Brown, M.G., Foote, J.T., Jones, G.J.F., & Sparck-Jones, K. (1997). Acoustic indexing for multimedia retrieval and browsing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97)*, Vol. 1, 199-202.

Zhang, H.J., Wang, J.Y.A., & Altunbasak, Y. (1997). Content-based video retrieval and compression: A unified solution. In *Proceedings of the International Conference on Image Processing (ICIP'97)*, Vol. 1, 13-16.

Zeng, W., Gao, W., & Zhao, D. (2002). Video indexing by motion activity maps. In *Proceedings of the International Conference on Image Processing (ICIP'02)*, Vol. 1, 912-915.

Chapter XIV

Video Abstraction

Jung Hwan Oh, The University of Texas at Arlington, USA

Quan Wen, The University of Texas at Arlington, USA

Sae Hwang, The University of Texas at Arlington, USA

Jeongkyu Lee, The University of Texas at Arlington, USA

ABSTRACT

This chapter introduces Video Abstraction, which is a short representation of an original video, and widely used in video cataloging, indexing, and retrieving. It provides a general view of video abstraction and presents different methods to produce various video abstracts. Also, it discusses a new approach to obtain a video abstract called video digest that uses the closed-caption information available in most videos. The method is efficient in segmenting long videos and producing various lengths of video abstracts automatically. The authors hope that this chapter not only gives newcomers a general and broad view of video abstraction, but also benefits the experienced researchers and professionals by presenting a comprehensive survey on state- of-the-art video abstraction and video digest methods.

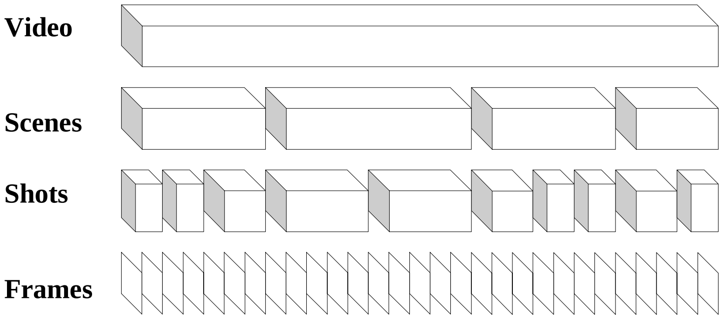
INTRODUCTION AND BACKGROUND

The volume of digital video data has been increasing significantly in recent years due to the wide use of multimedia applications in the areas of education, entertainment, business, and medicine. To handle this huge amount of data efficiently, many techniques about video segmentation, indexing, and abstraction have emerged to catalog, index, and retrieve the stored digital videos. The topic of this chapter is *video abstraction*, a short representation of an original video that helps to enable the fast browsing and retrieving of the represented contents. A general view of video abstraction, its related works, and a new approach to generate it will be presented in this chapter. Digital video data refers to the video picture and audio information stored by the computer using digital format. In this chapter, the terms "digital video," "video," "film," and "movie" all refer to a digital video unless specified with clarifications.

Before discussing the details of video abstraction, we provide readers with a fundamental view on video. Video consists of a collection of video frames, where each frame is a picture image. When a video is being played, each frame is being displayed sequentially with a certain frame rate. The typical frame rates are 30 and 25 frames/second as seen in the various video formats (NTSC, PAL, etc.). An hour of video has 108,000 or 90,000 frames if it has a 30 or 25 frames/second rate, respectively. No matter what kind of video format is used, this is a huge amount of data, and it is inefficient to handle a video by using all the frames it has. To address this problem, video is divided into segments, and more important and interesting segments are selected for a shorter form — a video abstraction. With granularity from small to large, the segmentation results can be *frame*, *shot*, *scene*, and *video*. Shot is a sequence of frames recorded in a single-camera operation, and scene is a collection of consecutive shots that have semantic similarity in object, person, space, and time. Figure 1 illustrates the relationship among them. Video abstraction methods will use these notions of video structure.

There are two types of video abstraction, *video summary* and *video skimming* (Li, Zhang, & Tretter, 2001). Video summary, also called a *still abstract,* is a set of salient images (*key frames*) selected or reconstructed from an original video sequence. Video skimming, also called a *moving abstract,* is a collection of image sequences along with the corresponding audios from an original video sequence. Video skimming is also called a *preview* of an original video, and can be classified into two sub-types: *highlight* and *summary sequence*. A highlight contains the most interesting and attractive parts of a video, while a summary sequence renders the impression of the content of an entire video. Among all types of video abstractions, summary sequence conveys the highest semantic meaning of the content of an original video. We will discuss the details of video summary and video skimming in the next two sections of the chapter. In a later section, we briefly describe the future work, and give our concluding remarks in the last section.

Figure 1. Structure of video



VIDEO SUMMARY

As mentioned in the Introduction, video summary is a set of salient images (key frames) selected or reconstructed from an original video sequence. Therefore, selecting salient images (key frames) from all the frames of an original video is very important to get a video summary. Several different methods using shot boundaries, visually perceptual features, feature spaces, and/or clusters will be discussed in the following subsections.

Shot Boundary-based Key Frame Selection

In the shot boundary-based key frame selection, a video is segmented into a number of shots and one or more key frames are selected from each shot. Together, these selected key frames form a video summary. Therefore, the main concern of this approach is how to detect shot boundaries. As mentioned in the previous section, a shot is defined as a collection of frames recorded from a single-camera operation. The principle methodology of shot-boundary detection is to extract one or more features from the frames in a video sequence, and then the difference between two consecutive frames is computed using the features. In case the difference is more than a certain threshold value, a shot boundary is declared.

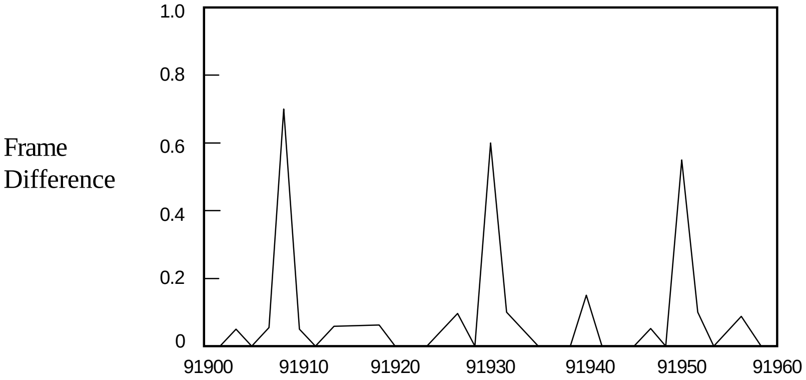
Many techniques have been developed to detect a shot boundary automatically. These schemes mainly differ in the way the inter-frame difference is computed. The difference can be determined by comparing the corresponding pixels of two consecutive frames (Ardizzone & Cascia, 1997; Gunsel, Ferman, & Tekalp, 1996; Swanberg, Shu, & Jain, 1993). Color or grayscale histograms can be also used (Abdel-Modttaleb & Dimitrova, 1996; Lienhart, Pfeiffer, & Effelsberg, 1996; Truong, Dorai, & Venkatesh, 2000; Yu & Wolf, 1997). Alternatively, a technique based on changes in the edges has also been developed (Zabih, Miller, & Mai, 1995). Other schemes use domain knowledge (Lienhart & Pfeiffer, 1997; Low, Tian, & Zhang, 1996), such as predefined models, objects, regions, etc. Hybrids of the above techniques have also been investigated (Adjeroh & Lee, 1997; Chang, Chen, Meng, Sundaram, & Zhong, 1997; Jiang, Helal, Elmagarmid, & Joshi, 1998; Oh & Hua, 2000; Oh, Hua, & Liang, 2000; Sun, Kankanhalli, Zhu, & Wu, 1998; Taskiran & Delp, 1998; Wactlar, Christel, Gong, & Hauptmann, 1999). Figure 2 shows the frame differences between two consecutive frames computed using the edge change ratio (Zabih et al., 1995) in a certain range (Frame #91900 to Frame #91960) of a video. As seen in the figure, three shot boundaries, between Frame #91906 and Frame #91907, between Frame #91930 and Frame #91931, and between Frame #91950 and Frame #91951 can be detected.

Once shot detection is completed, key frames are selected from each shot. For example, the first, the middle, or the last frame of each shot can be selected as key frames (Hammoud & Mohr, 2000). If a significant change occurs within a shot, more than one key frame can be selected for the shot (Dufaux, 2000).

Perceptual Feature-based Key Frame Selection

In the perceptual feature-based key frame selection, the first frame is selected initially as the most recent key frame, then the following frames are compared using the

Figure 2. Example of frame differences by edge change ratio



visually perceptual features. The examples of those features include color, motion, edge, shape, and spatial relationship (Zhang, 1997). If the difference between the current frame and the most recent key frame exceeds a predefined threshold, the current frame is selected as a key frame. We discuss three methods that use different features as follows.

Color-Based Selection

Color is one of the most important features for video frames; it can distinguish an image from others since there is little possibility that two images of totally different objects have very similar colors. Color histogram is a popular method to describe the color feature in a frame due to its simplicity and accuracy. It selects *N* color bins to represent the entire color space of a video and counts how many pixels belong to each color bin of each frame. Zhang (1997) first quantizes the color space into 64 super-cells. Then, a 64-bin color histogram is calculated for each frame where each bin is assigned the normalized count of the number of pixels. The distance ($D_{his}(I,Q)$) between two color histograms, *I* and *Q*, each consisting of *N* bins, is quantified by the following metric:

$$D_{his}(I,Q) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} a_{ij}(I_i - Q_i)(I_j - Q_j) \qquad \textbf{(1)}$$

where the matrix a_{ij} represents the similarity between the colors corresponding to bins *i* and *j*, respectively. This matrix needs to be determined from human visual perception studies. If a_{ij} is an identity matrix, this equation measures the Euclidean distance between two color histograms.

After the first key frame is decided manually, the color histograms of consecutive frames are compared with that of the last selected key frame using Equation (1). If the distance is larger than a predefined threshold, the current frame is decided as the last key frame. The user can change the threshold value to control the amount of key frames. A larger threshold will produce less key frames. On the contrary, a lower threshold will produce more key frames.

Motion-Based Selection

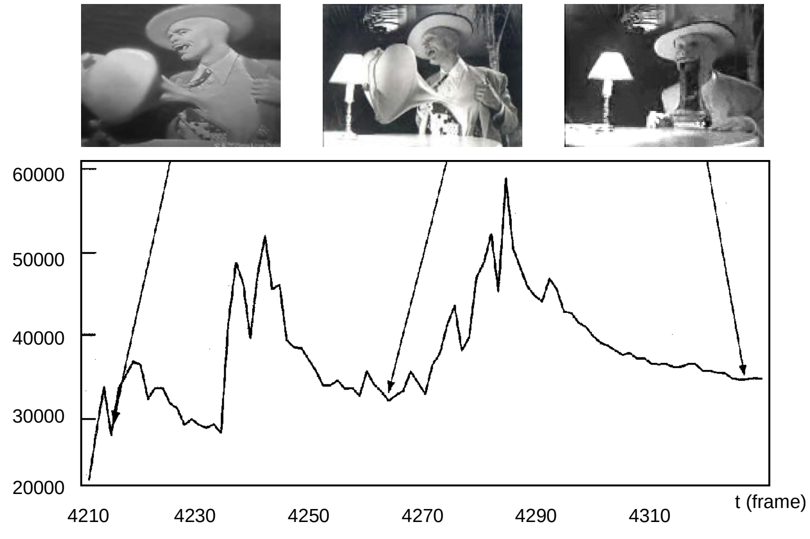
A color histogram is insensitive to camera and object motion (Wolf, 1996; Zhang, 1997). In film production, a director often pans and zooms the camera from one location to another to show the connection between two events. Similarly, several distinct and important gestures by a person will appear in one shot. Therefore, color-based key frame selection may not be enough to render the visual contents of a shot. Wolf (1996) uses motions to identify key frames. In the algorithm for key frame identification, a motion metric, *M(t)* based on optical flow is computed for frame *t* with a size of *r´c* using the following formula:

$$M(t)=\sum_{i=1}^{r}\sum_{j=1}^{c}\left|o_x(i,j,t)\right|+\left|o_y(i,j,t)\right| \qquad \textbf{(2)}$$

where $o_x(i, j, t)$ is the *x* component of optical flow of a pixel positioned *i* and *j* in frame *t* and similarly $o_y(i, j, t)$ for the *y* component. Then, the metric is analyzed as a function of time to select key frames at the minima of motion.

The analysis begins at *t*=0, and identifies two local maxima, m_1 and m_2 using Equation (2) such that the difference between the two values (m_1 and m_2) is larger than a predefined threshold. A frame with a value of the local minimum of *M(t)* between these two local maxima is selected as a key frame. The current m_2 is selected as m_1, and the algorithm continues to find the next m_2 in temporal order. Figure 3 shows the values of

Figure 3. Values of M(t)s and key frames from a shot in the movie, The Mask



*M(t)*s for the frames in a shot and a couple of key frames selected by the algorithm. The *M(t)* curve clearly shows the local maxima and minima of motion in the shot.

Object-based Selection

Object-based key frame selection methods can be found in the literature (Ferman, Gunsel, & Tekalp1997; Kim & Huang, 2001). Figure 4 illustrates the integrated scheme for object-based key frame extraction (KFE) (Kim & Huang, 2001), which can be briefly described as follows.

First, it computes the difference of the number of regions between the last key frame and the current frame. When the difference exceeds a certain threshold value, the current frame is considered as a new key frame assuming a different event occurs.

In case the difference is less than a certain threshold value, two 7-dimensional feature vectors (x_k and x_{last}) for the current frame and the last key frame are generated using the seven Hu moments (Nagasaka & Tanaka, 1991; Zhang, 1997), which are known as reasonable shape descriptors. Then, the distance, $D(F_{last}, F_k)$ is computed between x_k and x_{last} by using the city block distance measure (Zhang, 1997). Because the city block distance, which is also called the "Manhattan metric," is the sum of the distances among all variables, it can measure spatial closeness, which helps to decide whether the current frame can be a new key frame. If this difference exceeds a given threshold value, the current frame is selected as a new key frame in the same event.

Feature Vector Space-based Key Frame Selection

The feature vector space-based key frame selection (DeMenthon, Kobla, & Doermann, 1998; Zhao, Qi, Li, Yang, & Zhang, 2000) considers that the frames in a video sequence are characterized by not just one but multiple features. Each frame can be represented by a vector with multiple features, which is a point in multi-dimensional feature space. And the entire feature vectors of the frames in a video sequence can form a curve in the feature space. Key frames are selected based on the property of the curve such as sharp corners or transformations. These perceptually significant points in the curve can be obtained by the multidimensional curve splitting algorithm, which was proposed by Ramer (1972).

Figure 4. Block diagram of integrated system for object-based key frame extraction

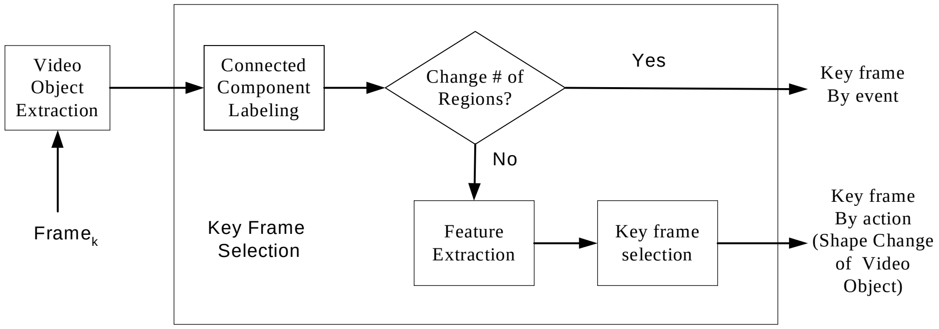
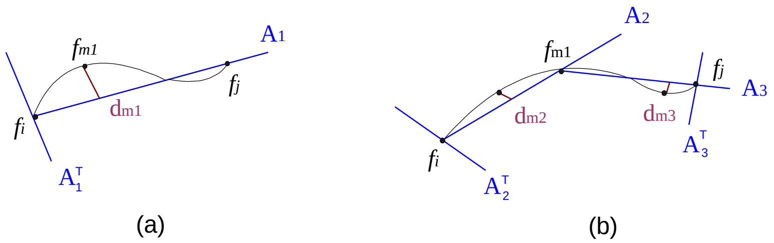


Figure 5. Curve-splitting algorithm



(a)

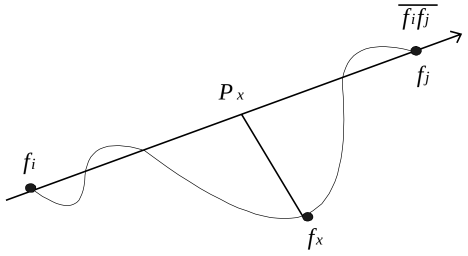
(b)

For illustrative purposes, we consider the two dimensional space (Figure 5). f_i and f_j in Figure 5(a) and (b) represent two feature vectors of the first frame and the last frame in a video, respectively. The curve represents the feature vectors of the entire frames in a video. A new Cartesian coordinate system will be built by the axis, A_1 pointing from f_i to f_j and the orthogonal axis, A_1^T. The maximum distance, d_{ml} between the curve and the axis, A_1 is calculated, and compared with the predefined threshold (x). If d_{m1} is larger than x, the curve will be split into two curve pieces f_if_{m1} and $f_{m1}f_j$, and the same procedure is applied recursively to each of the two curve segments (Figure 5b) until the maximum distance is smaller than the threshold, x.

In the multi-dimensional feature space, we denote a feature vector, f_x of each frame in a video sequence as the following: $f_x = \{ f_{1x}, f_{2x}, \ldots, f_{nx} \}$, where n is the dimension of the feature space. As shown in Figure 6, the distance between the feature point, f_x and the feature line, f_if_j is calculated by the followings: $\mathrm{Dist}(f_x, f_if_j) = | f_x - p_x|$, where $p_x = f_i + \mathrm{m}(f_j - f_i)$ and $\mathrm{m} = (f_x - f_i)(f_j - f_i) / (f_j - f_i)(f_j - f_i)$.

The difference between frames F_i and F_j can be measured as the Euclidean distance in the feature space. The shape and the dimensionality of the feature curve in a video sequence can be formed based on the feature vectors to characterize each individual frame. Therefore, choice of proper features is an important factor for the feature vector space-based key frame selection.

Figure 6. Multi-dimensional feature space



Cluster-based Key Frame Selection

If the number of key frames for each shot is limited to one, it may not represent the content of a shot very well since the complexity of each shot is hardly reflected by one frame. Several key frame-selection techniques based on clustering have been proposed (Hanjalic & Zhang, 1999; Sun et al., 1998; Uchihashi, 1999; Wolf, 1996; Zhuang, Rui, Huang, & Mehrotra, 1998) that select a proper number of key frames from a shot.

In a brief explanation of the cluster-based key frame selection, a given shot, *s* has *N* number of frames, and these *N* number of frames, $\{f_1, f_2, \ldots, f_N\}$ are clustered into *M* number of clusters, $\{C_1, C_2, \ldots, C_M\}$. This clustering is based on the similarity measures among frames, where the similarity of two frames is defined as the similarity of their features, such as color, texture, shape, or a combination of the above. Initially, the first frame, f_1 is selected as the centroid of the first cluster. Then, the similarity values are measured between the next frame f_i and the centroids of existing clusters C_k ($k = 1, 2, \ldots, M$), such that the maximum value and its corresponding cluster, C_j are determined. If this maximum value is less than a certain threshold value, it means frame f_i is not close enough to be added into any existing cluster, then a new cluster is formed for frame f_i. Otherwise, frame f_i is put into the corresponding cluster, C_j. The above process is repeated until the last frame f_N is assigned into a cluster. This is a simple clustering algorithm, but more sophisticated algorithms (e.g., K-mean [Ngo, Pong, & Zhang, 2001]) can be used. After the clusters are constructed, the representative frames are extracted as key frames from the clusters.

Zhuang et al. (1998) use the color histogram of a frame as the feature and select the frame that is closest to the centroid of a cluster as a key frame. They also consider the cluster size such that if the size of a cluster is smaller than a predefined value, those smaller clusters are merged into a larger one using a pruning technique. Sun et al. (1998) perform an iterative partitional-clustering procedure for key frame selection. First, a difference is computed between the first and last frames in each shot. If the difference is less than a threshold value, only the first and last frames are selected as key frames. If the difference exceeds a threshold value and the size of the cluster is smaller than the tolerable maximum size, all frames in the cluster are taken as key frames. Even if the difference is larger than the threshold but the size of the cluster is larger than the tolerable size, the cluster is divided into sub-clusters with the same size, and the partitional-clustering procedure for each sub-cluster is iterated. Hammoud and Mohr (2000) extract multiple key-frames to represent a cluster. First, they select a key frame that is the closest frame to the centroid of a cluster. The similarity between the key frame and each frame in a cluster is calculated. If this similarity is larger than a predefined similarity threshold, the frame is added to a set of key frames. A temporal filter is applied on the set of all selected key frames in order to eliminate the overlapping cases among the constructed clusters of frames.

Other Methods

There are other methods for selecting key frames besides the key frame extraction methods mentioned above. The most intuitive way is to select key frames by sampling at fixed or random distances among frames. The others include face and skin-color detection-based (Dufaux, 2000), statistic-based (Yfantis, 2001), and time-constrained-based (Girgensohn & Boreczky, 2000) methods. Ideas combining several of the above methods are also very common in practice.

VIDEO SKIMMING

As mentioned at the beginning of this chapter, video abstraction is classified into two types: *video summary* and *video skimming*. We have discussed methods in getting video summary in the previous section. In this section, the methods for producing video skimming will be explored. Video skimming consists of a collection of image sequences along with the related audios from an original video. It possesses a higher level of semantic meaning of an original video than the video summary does. We will discuss the video skimming in the following two subsections according to its classification: *highlight* and *summary sequence*.

Highlight

A highlight has the most interesting parts of a video. It is similar to a trailer of a movie, showing the most attractive scenes without revealing the ending of a film. Thus, highlight is used in a film domain frequently. A general method to produce highlights is discussed here. The basic idea of producing a highlight is to extract the most interesting and exciting scenes that contain important people, sounds, and actions, then concatenate them together (Kang, 2001a; Pfeiffer, Lienhart, Fischer, & Effelsberg, 1996). It is illustrated in Figure 7.

Pfeiffer et al. (1996) used visual features to produce a highlight of a feature film and stated that a good cinema trailer must have the following five features: (1) important objects/people, (2) action, (3) mood, (4) dialog, and (5) a disguised ending. These features mean that a highlight should include important objects and people appearing in an original film, many actions to attract viewers, the basic mood of a movie, and dialogs containing important information. Finally, the highlight needs to hide the ending of a movie.

In the *VAbstract* system (Pfeiffer et al., 1996), a scene is considered as the basic entity for a highlight. Therefore, the scene boundary detection is performed first using existing techniques (Kang, 2001b; Sundaram & Chang, 2000; Wang & Chua, 2002; Zabih et al., 1995). Then, it finds the high-contrast scenes to fulfill the trailer Feature 1, the high-motion scenes to fulfill Feature 2, the scenes with basic color composition similar to the average color composition of the whole movie to fulfill Feature 3, the scenes with dialog of various speeches to fulfill Feature 4, and deletes any scene from the last part of an original video to fulfill Feature 5. Finally, all the selected scenes are concatenated together in temporal order to form a movie trailer. Figure 8 shows the abstracting algorithm in the *VAbstract* system.

Figure 7. Diagram of producing a video highlight

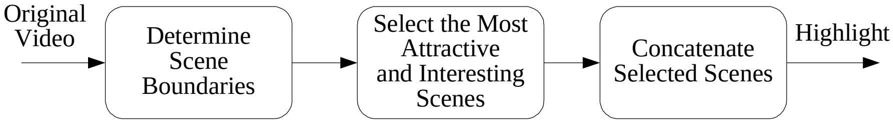
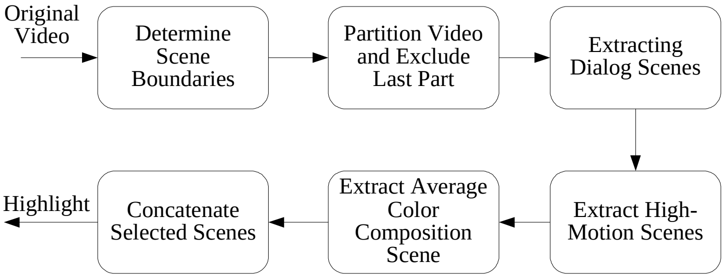

Original Video
Determine Scene Boundaries
Select the Most Attractive and Interesting Scenes
Concatenate Selected Scenes
Highlight

Figure 8. VAbstract abstracting algorithm


Original Video
Determine Scene Boundaries
Partition Video and Exclude Last Part
Extracting Dialog Scenes
Extract High-Motion Scenes
Extract Average Color Composition Scene
Concatenate Selected Scenes
Highlight

We will now discuss the main steps in *VAbstract* system, which are scene boundary detection, extraction of dialog scene, extraction of high-motion scene, and extraction of average color. More details can be found in Pfeiffer et al. (1996).

- *Scene Boundary Detection:* Scene change can be determined by the combination of video- and audio-cut detections. Video-cut detection finds sharp transition, namely *cut* between frames. The results of this video-cut detection are shots. To group the relevant shots into a scene, audio-cut detection is used. A video cut can be detected by using color histogram. If the color histogram difference between two consecutive frames exceeds a threshold, then a cut is determined. The details of the audio-cut detection method can be found in Gerum (1996).
- *Extraction of Dialog Scene:* A heuristic method is used to detect dialog scenes. It is based on the finding that a dialog is characterized by the existence of two "a"s with significantly different fundamental frequencies, which indicates that those two "a"s are spoken by two different people. Therefore, the audio track is first transformed to a short-term frequency spectrum and then normalized to compare with the spectrum of a spoken "a." Because "a" is spoken as a long sound and occurs frequently in most conversations, this heuristic method is easy to implement and effective in practice.
- *Extraction of High-Motion Scene:* Motion in a scene often includes camera motion, object motion, or both. The related motion-detection methods can be found in Section 2.2.2, and in Oh and Sankuratri (2002). A scene with a high degree of motion will be included in the highlight.
- *Extraction of Average Color Scene:* A video's mood is embodied by the colors of each frame. The scenes in the highlight should have the color compositions similar to the entire video. Here, the color composition has physical color properties such as luminance, hue, and saturation. It computes the average color composition of the entire video and finds scenes whose color compositions are similar to the average.

Summary Sequence

Being different from *highlight*, which focuses on the most interesting parts of a video, *summary sequence* renders the impressions of the content of an entire video. It conveys the highest level of semantic meaning of an original video among all the video abstraction categories. Some representative methods, such as time compression-based, model-based, and text- and speech-recognition- based methods are discussed in the following subsections.

Time Compression-Based Method

The methods to obtain summary sequence are diverse. Text- and speech-recognition- based and model-based methods are two major categories. Also, there are other methods such as the speed-up method to generate a video skimming. Omoigui He, Gupta, Grudin, and Sanocki, (1999) use a time-compression technology to speed up watching a video. This time-compression method consists of two aspects: audio compression and video compression. In this section, we will describe them briefly.

Audio compression can be obtained in a very intuitive way. Suppose we divide the entire audio clip into the equal-sized segments with a length of 100 milliseconds (ms) each. If we delete a 25 ms portion from each segment and concatenate all the remaining 75 ms portions, the total length of the entire audio will be reduced to three-quarters of the original one. The drawback of this simple method is that there is some sound distortion, although the intelligibility of the audio (speech) is un-affected. Other ways to improve the quality of audio time-compression are selective sampling (Neuburg, 1978), sampling with dichotic presentation (Orr, 1971), and Short-Time Fourier Transform (Griffin & Lim, 1984).

As to video compression, it simply drops the frames according to the compression ratio of the audio. If we use the audio time-compression example mentioned above, in which the audio is compressed to three-quarters of an original video, then one frame will be dropped for every four video frames.

Model-Based Method

Some types of videos have fixed structures that can facilitate the process of extracting important segments. Sports and news programs may fall into this category. A number of video- skimming methods based on the modeling for these types of videos have been reported (Babaguchi, 2000; Li & Sezan, 2002). The basic idea of the model-based method is to use the special structure of the video to select its most important scenes. These special structures include fixed scene structures, dominant locations, and backgrounds. Figure 9 shows the basic idea.

A model-based approach in Li and Sezan (2002) depicts how to model videos based on their domain knowledge. We will describe the details of the method in this section. Li and Sezan summarize the American football broadcast video by defining a set of plays appearing in an entire game. The idea is that first, the start of a play is detected by using field color, field lines, camera motions, team jersey colors, and player line-ups. Second,

Figure 9. Model-based method for summary sequence

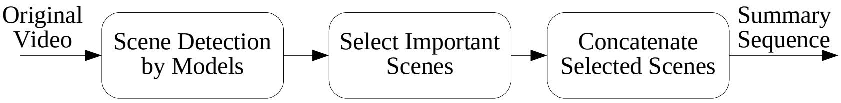
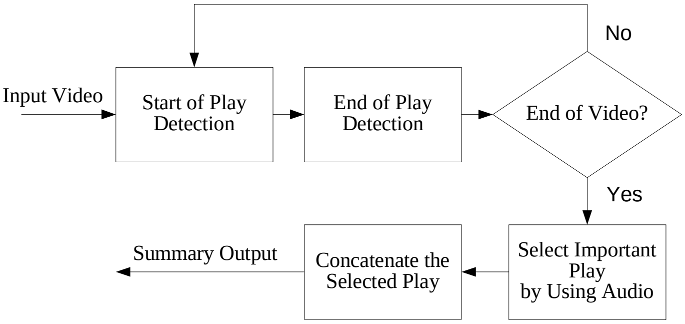


Figure 10. Diagram of a football video summarizer



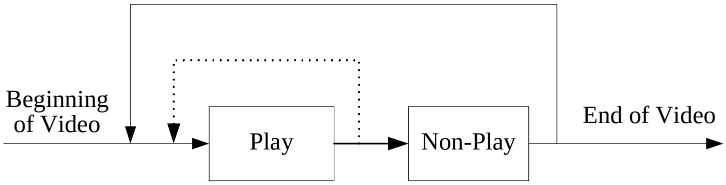
the end of the play is detected by finding camera breaks after the play starts. At last, the waveform of the audio is used to find the most exciting plays, and a summary of the game is constructed by combining them. Figure 10 gives a general view of the whole process.

The method comes out from the observation that generally a football broadcast lasts about three hours but the actual game time is only 60 minutes. There are many unexciting video segments in the broadcast when, for example, the ball is "dead" or there is no play. Therefore, the method defines a "play" as the period between the time when the offensive team has the ball and is about to attempt an advance and the time when the ball is dead. The whole football video is modeled as a series of "plays" interleaved with non-plays. Figure 11 illustrates the structure model of the football video.

Text- and Speech-Recognition-based Method

As mentioned in the previous section, the model-based approach can only be applied to certain types of videos such as sports or news programs that have a certain type of fixed structures. However, most of the other videos do not have these structures. In other words, we need a different approach to apply to general videos that do not have any structure for modeling. To address this problem, the audio information — especially speech in a video — is widely used. This speech information can be obtained from caption data or speech recognition. Caption data usually helps people with hearing problems watch TV programs. Now it is broadly used by the main TV channels and many educational audio and video materials. There are two types of captions, namely, open

Figure 11. Structure model of football video (the inner loop (in dotted line) means that there may have no non-play between plays)



caption and closed caption. Open-caption data is stored and displayed as a part of video frames. Closed-caption data is stored separately from each video frame and displayed as an overlap on video frames. When there is no caption data provided, speech recognition technologies are used to obtain the corresponding text information. Some brands of commercial speech recognition software are available with good performance, such as *Dragon Naturally Speaking* by Scan Soft and *Via Voice* by IBM.

In this section, first, the general idea of existing methods using the text from caption data or speech recognition to get summary sequence will be presented. Then, a new method called *video digest* is discussed.

General Idea of Existing Methods

The general idea of text- and speech-recognition based methods (Agnihoti, 2001; Alexander, 1997; Christel, Smith, Taylor, & Winkler1998; Fujimura, Honda, & Uehara, 2002; He, Sanoki, Gupta, & Grudin, 1999; Li, 2001; Smith & Kanade, 1997; Taskiran, Amir, Ponceleon, & Delp, 2002) is simple, and falls into four steps:

Step 1. Segment a video into a number of shots (or scenes) according to its visual and/or audio (not speech) contents.

Step 2. Obtain text information from the video by capturing caption data or using speech recognition. One (i.e., Signal to Noise Ratio (SNR) technique) of the natural language processing (NLP) techniques is used to get the dominant words or phrases from the text information.

Step 3. Find the shots (or scenes) including the dominant words or phrases obtained in *Step 2*.

Step 4. Concatenate the corresponding shots (or scenes) obtained in *Step 3* together in temporal order.

The main drawbacks of the existing approach are as follows. First, the segmentation results do not always reflect the semantic decomposition of a video, so the generated summary is not always optimal. Therefore, we need a different segmentation technique that considers the semantic of a video. Second, the dominant words or phrases may not be distributed uniformly throughout a video so that the generated summary may miss certain parts of the video. Thus, we need to get different units (e.g., sentences) instead of dominant words or phrases. Third, the existing approach produces only one version (with a fixed length) of a summary from a video due to the lack of its flexibility. However, it is desirable to have several versions (with various lengths) of summaries to satisfy numerous applications with diverse requirements. Fourth, some existing approaches are dependent upon a number of specific symbols (e.g., ">", ">>", or ">>>") in caption or domain knowledge so that they cannot be applied generally. We need a new approach independent of any specific symbol or domain knowledge. To address the four issues above, we introduce a new approach for video-summary sequence as follows.

Top-down Approach for Video Segmentation

The first task for video-summary sequence is to partition a video into a number of segments. The existing methods for video- summary sequence adapt one of the existing shot- boundary detection (SBD) techniques to get the segments that are shots (or scenes). As mentioned previously, these SBD techniques are bottom-up, in which a sequence of frames is extracted from a video and two consecutive frames are compared to find a boundary. Since these shots are very short, a number of related shots are grouped into a scene. However, it is still an open problem to find optimal scene boundaries by grouping related shots automatically, as mentioned in the literatures (Corridoni, Bimbo, Lucarella, & Wenxue, 1996; Jiang & Elmagarmid, 1998; Rui, Huang, & Mehratra, 1999; Zhong, Zhang, & Chang, 1997). To address this, we segment a video based on top-down fashion. In our technique, a video is segmented into a number of paragraphs using the time gaps that do not have any audio. We call this segment "paragraph" since it is based on the entire text information in a video. Figure 12 shows a sample of a closed-caption script for a documentary, "The Great War." The time-stamps (that have a time format of hour: minute: second: 1/100 second, and are the relative times from the beginning of video) in the first column indicate the starting times of the audios (i.e., speech, music, etc.) in the second column. However, a blank line is occasionally followed by a time-stamp. For example, the fourth time-stamp (0:1:40:75) is followed by a blank line, and the fifth time-stamp (0:1:42:78) has a sentence. In other words, a no-audio time gap lasts around two seconds (00:1:40:75 ~ 00:1:42:78) between [Dramatic Music] and a sentence "IT COLORED EVERYTHING …".

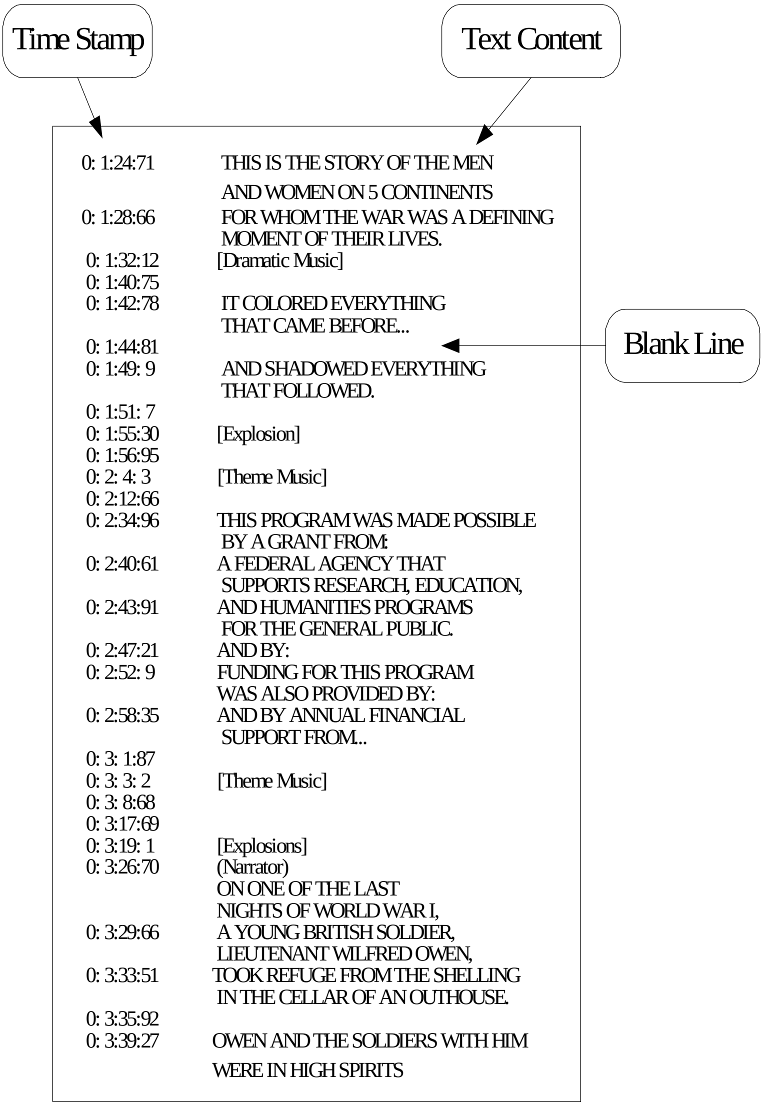
We use these no-audio time gaps to segment a video, but, we only consider the gaps between sentences, music, or sound effects. The gaps in the middle of sentences, music, or sound effects are not used for the segmentation. Figure 13 shows the no-audio time gaps between audios in Figure 12. The long gap between two audios implies semantic segmentation of the original script. In Figure 13, the two largest gaps, #5 and #7, with durations larger than 10 seconds divide the script into three paragraphs. Each paragraph talks about different topics. The first paragraph, which is before Gap #5, tells about the influence of the World War; the second paragraph, which is between Gap #5 and #7, states the contributors of the video; the third paragraph, which is after Gap #7, begins to state a story about a man. In general, an entire video is segmented into a number of paragraphs based on the predefined threshold (e.g., 10 seconds) about the no-audio time gap. If a paragraph is too long, it is re-segmented into subparagraphs.

Summary-Sequence Generation

After a video is segmented into the paragraphs (or subparagraphs), we extract not the words or phrases but the dominant sentences by using one of the Natural Language Processing tools. We can get a number of different versions of summaries that have various lengths by controlling the number of dominant sentences per paragraph (or subparagraph). Since every sentence has its beginning time-stamp and ending time-stamp (which is the beginning of the next one), it is convenient to extract audio and video corresponding to a target sentence. Suppose the sentence, "THIS IS THE STORY OF THE MEN AND WOMEN ON FIVE CONTINENTS FOR WHOM THE WAR WAS A DEFINING MOMENT OF THEIR LIVES" in Figure 12, appears in the summarized text. Then, we use its beginning time-stamp "0:1:24:71" and ending time-stamp "0:1:32:12" to allocate the corresponding audio and video with a length of 7.41 seconds.

The result of our video-skimming approach is called *video digest* to distinguish it from the others. Here are the steps of our approach to get video digest:

Figure 12. Sample of closed-caption script for "The Great War"


Time Stamp
Text Content
Blank Line

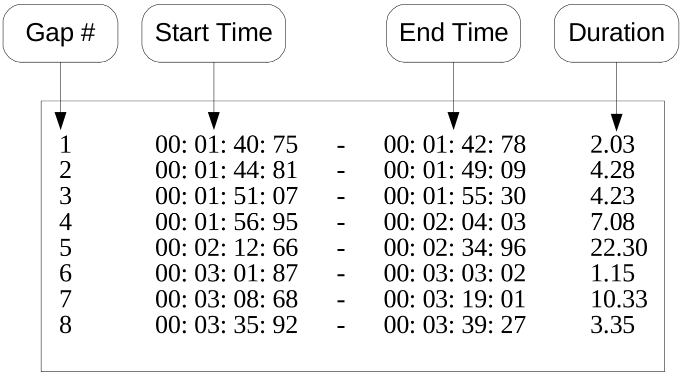
```
0: 1:24:71      THIS IS THE STORY OF THE MEN
                AND WOMEN ON 5 CONTINENTS
0: 1:28:66      FOR WHOM THE WAR WAS A DEFINING
                MOMENT OF THEIR LIVES.
 0: 1:32:12     [Dramatic Music]
 0: 1:40:75
 0: 1:42:78     IT COLORED EVERYTHING
                THAT CAME BEFORE...
 0: 1:44:81
 0: 1:49: 9     AND SHADOWED EVERYTHING
                THAT FOLLOWED.
 0: 1:51: 7
 0: 1:55:30     [Explosion]
 0: 1:56:95
 0: 2: 4: 3     [Theme Music]
 0: 2:12:66
 0: 2:34:96     THIS PROGRAM WAS MADE POSSIBLE
                 BY A GRANT FROM:
 0: 2:40:61     A FEDERAL AGENCY THAT
                 SUPPORTS RESEARCH, EDUCATION,
 0: 2:43:91     AND HUMANITIES PROGRAMS
                 FOR THE GENERAL PUBLIC.
 0: 2:47:21     AND BY:
 0: 2:52: 9     FUNDING FOR THIS PROGRAM
                WAS ALSO PROVIDED BY:
 0: 2:58:35     AND BY ANNUAL FINANCIAL
                 SUPPORT FROM...
 0: 3: 1:87
 0: 3: 3: 2     [Theme Music]
 0: 3: 8:68
 0: 3:17:69
 0: 3:19: 1     [Explosions]
 0: 3:26:70     (Narrator)
                ON ONE OF THE LAST
                NIGHTS OF WORLD WAR I,
 0: 3:29:66     A YOUNG BRITISH SOLDIER,
                LIEUTENANT WILFRED OWEN,
 0: 3:33:51    TOOK REFUGE FROM THE SHELLING
                IN THE CELLAR OF AN OUTHOUSE.
 0: 3:35:92
 0: 3:39:27    OWEN AND THE SOLDIERS WITH HIM

               WERE IN HIGH SPIRITS
```

Step 1. Extract the closed-caption script (including time-stamp) as seen in Figure 12 from a video using a caption-decoder device. If caption data is not available, speech recognition can be used to get the same text information.

Step 2. Compute the no-audio time gaps followed by the blank lines as shown in Figure 13, then segment the entire text into paragraphs (or subparagraphs) based on these gaps, using a certain threshold.

Figure 13. No-audio time gaps in Figure 12



Gap #	Start Time		End Time	Duration
1	00: 01: 40: 75	-	00: 01: 42: 78	2.03
2	00: 01: 44: 81	-	00: 01: 49: 09	4.28
3	00: 01: 51: 07	-	00: 01: 55: 30	4.23
4	00: 01: 56: 95	-	00: 02: 04: 03	7.08
5	00: 02: 12: 66	-	00: 02: 34: 96	22.30
6	00: 03: 01: 87	-	00: 03: 03: 02	1.15
7	00: 03: 08: 68	-	00: 03: 19: 01	10.33
8	00: 03: 35: 92	-	00: 03: 39: 27	3.35

Step 3. Extract a number of dominant sentences from each paragraph (or subparagraph). We can control the length of the summary by controlling the number of dominant sentences.

Step 4. Extract videos and audios corresponding to the dominant sentences and concatenate them together in the temporal order.

The advantages of our approach are:

- Our segmentation results reflect the semantic decomposition of a video so that the generated summary is optimal.
- Instead of words or phrases, we introduce a more effective and efficient unit — a sentence — for video segmentation and summary generation.
- Our approach can build several versions (with various lengths) of summaries to satisfy numerous applications with diverse requirements.
- The proposed approach is independent of any specific symbol or domain knowledge.
- Since our approach is based on the spoken sentences, seamless concatenation is easy to achieve automatically.

Experiment Results of Segmentation

Six documentary videos, "The Great War," "Solar Blast," "Brooklyn Bridge," "Nature's Cheats," "Poison Dart Frogs," and "Red Monkey of Zanzibar," are used as the test materials. A number of segments separated by the no-audio time gaps whose duration is larger than 10 seconds are shown in Table 1.

For example, Figure 14 shows the entire no-audio time gaps for a video, "The Great War." There are 11 gaps larger than 10 seconds that separate the entire video into 12 segments (paragraphs).

To measure the effectiveness of our text segmentation approach, we use the *recall* and *precision* metrics. *Recall* (*C/T*) is the ratio of the number (*C*) of paragraph boundaries

Table 1. Segmentation results for test videos (video length format is hh:mm:ss:1/100second)

Video Name	Video Length	Paragraphs
The Great War	00:58:38:20	12
Red Monkey of Zanzibar	00:26:52:20	19
Solar Blast	00:57:07:15	16
Nature's Cheats	00:26:40:09	7
Poison Dart Frogs	00:27:12:18	19
Brooklyn Bridge	01:01:16:10	15
Total	04:17:45:92	88

detected correctly over the actual number (T) of paragraph boundaries. *Precision* (C/D) is the ratio of the number (C) of paragraph boundaries detected correctly over the total number (D) of paragraph boundaries detected correctly or incorrectly. The performance of our method is illustrated in Table 2. The actual number (T) of paragraph boundaries of each video is subjectively defined based on our understanding of the content because the original scripts of all the videos are not in paragraph format. There may be a variant value of T, depending on different segmentation granularities used.

As seen in Table 2, the overall results are very good. However, the performance for the video, "Nature's Cheats" is not as good as the others because it has a collection of different natural phenomena, and there are some long, speechless portions in the original video that use just pictures to depict the phenomena.

Figure 14. Pauses of video in "The Great War"

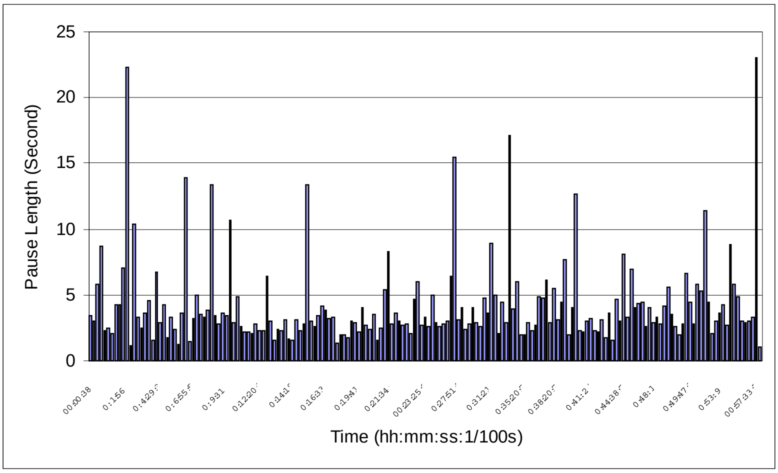


Table 2. Recall and precision of video digest

Video Name	*C*	*D*	*T*	*C/D*	*C/T*
The Great War	11	11	11	1	1
Red Monkey of Zanzibar	18	18	18	1	1
Solar Blast	15	15	15	1	1
Nature's Cheats	5	6	5	0.83	1
Poison Dart Frogs	18	18	18	1	1
Brooklyn Bridge	14	14	14	1	1

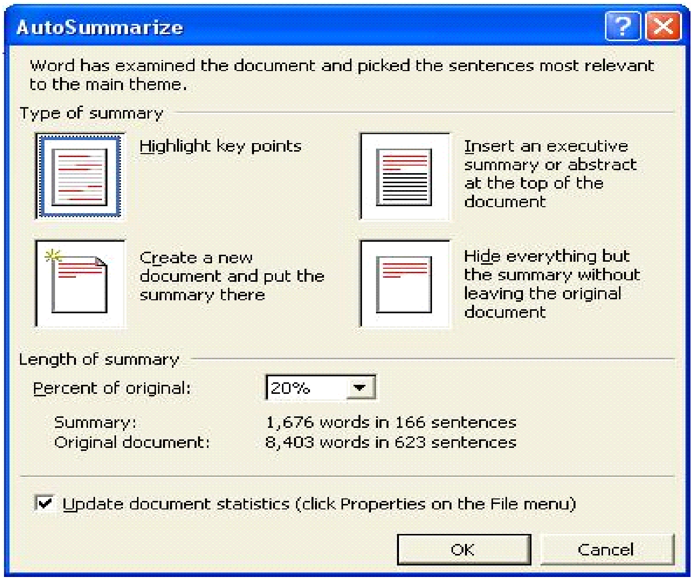
Experiment Results of Text Summarization

In our experiments, we use the AutoSummarize tool of Microsoft Word to extract the dominant sentences from the original text script because of its convenience in changing the summary length. Figure 15 shows the AutoSummarize dialog window requiring the user to choose "Type of summary" and "Length of summary."

Here, we give an example of applying different summarization ratios to the third paragraph that is segmented from the video, "The Great War," using the proposed technique. The actual content of the paragraph is shown in Figure 16.

The results of applying different summarization ratios by changing "Percent of Original" in Figure 15 are shown in Figure 17 (a), (b), and (c). If we put 5% as "Percent of Original," we get the result in Figure 17(a) (left). In other words, the most important

Figure 15. Dialog window of AutoSummarize



Word has examined the document and picked the sentences most relevant to the main theme.

Type of summary

Highlight key points

Insert an executive summary or abstract at the top of the document

Create a new document and put the summary there

Hide everything but the summary without leaving the original document

Length of summary

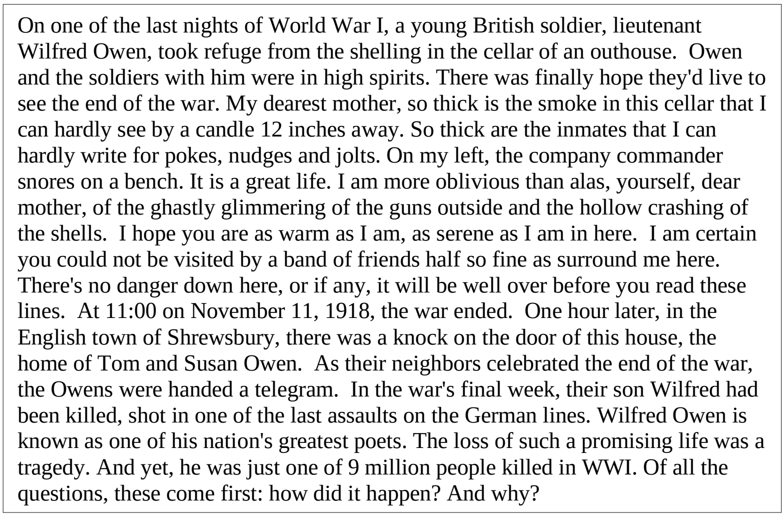
Percent of original: 20%

Summary: 1,676 words in 166 sentences

Original document: 8,403 words in 623 sentences

☑ Update document statistics (click Properties on the File menu)

Figure 16. The third paragraph of "The Great War"



On one of the last nights of World War I, a young British soldier, lieutenant Wilfred Owen, took refuge from the shelling in the cellar of an outhouse. Owen and the soldiers with him were in high spirits. There was finally hope they'd live to see the end of the war. My dearest mother, so thick is the smoke in this cellar that I can hardly see by a candle 12 inches away. So thick are the inmates that I can hardly write for pokes, nudges and jolts. On my left, the company commander snores on a bench. It is a great life. I am more oblivious than alas, yourself, dear mother, of the ghastly glimmering of the guns outside and the hollow crashing of the shells. I hope you are as warm as I am, as serene as I am in here. I am certain you could not be visited by a band of friends half so fine as surround me here. There's no danger down here, or if any, it will be well over before you read these lines. At 11:00 on November 11, 1918, the war ended. One hour later, in the English town of Shrewsbury, there was a knock on the door of this house, the home of Tom and Susan Owen. As their neighbors celebrated the end of the war, the Owens were handed a telegram. In the war's final week, their son Wilfred had been killed, shot in one of the last assaults on the German lines. Wilfred Owen is known as one of his nation's greatest poets. The loss of such a promising life was a tragedy. And yet, he was just one of 9 million people killed in WWI. Of all the questions, these come first: how did it happen? And why?

5% of this paragraph is a sentence, "At 11:00 on November 11, 1918, the war ended." If we put 10% as "Percent of Original," we get the result in Figure 17(a) (right). If we put 20% as "Percent of Original," we get the result in Figure 17(b), and so on.

The relationship among these different lengths of summaries is that the summary result of the larger summary ratio includes that of the smaller summary ratio as seen in Figure 18. For instance, the summary result of the 20% ratio includes all that of the 10%. The summary contents of the 20~40% ratios are found to express the idea of the paragraph reasonably.

As the examples of using different summary ratios demonstrate, all six videos in our test set are segmented into the corresponding number of paragraphs as seen in Table 1 by using a 10- second threshold for the no-audio time gaps. Then, by applying different summarization ratios, we get the results in Table 3 (Time format is hh:mm:ss:1/100s). It is very useful for a user since she/he can choose any ratio at runtime. If a user just wants to see a quick view for the semantic meaning of the video content, the user can select a brief video digest of about 5~10% of the original video length. Larger ratio values (50~80%) can be used for various purposes, in case of time shortage. Figure 19, 20, and 21 show the actual contents of summaries using the 5% ratio from three test videos, "Red Monkey of Zanzibar," "Nature's Cheats," and "Poison Dart Frogs." As seen in these figures, they have the most important points in the videos.

FUTURE WORK

The preliminary result of our video digest method is promising. It is efficient in keeping the semantic meaning of the original video content, and provides various

Figure 17(a). Summary results of ratio 5% (left) and 10% (right)

At 11:00 on November 11, 1918, the war ended.

On one of the last nights of World War I, a young British soldier, lieutenant Wilfred Owen, took refuge from the shelling in the cellar of an outhouse. At 11:00 on November 11, 1918, the war ended.

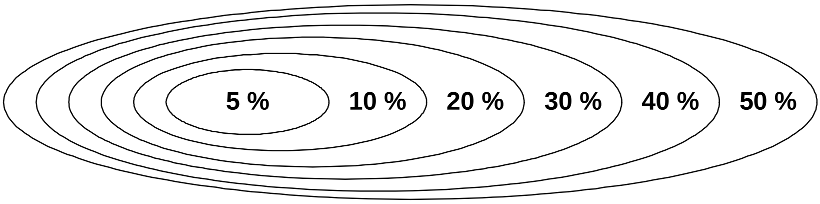
Figure 17(b). Summary result of ratio 20%

On one of the last nights of World War I, a young British soldier, lieutenant Wilfred Owen, took refuge from the shelling in the cellar of an outhouse. It is a great life. At 11:00 on November 11, 1918, the war ended. Wilfred Owen is known as one of his nation's greatest poets.

Figure 17(c). Summary result of ratio 50%

On one of the last nights of World War I, a young British soldier, lieutenant Wilfred Owen, took refuge from the shelling in the cellar of an outhouse. Owen and the soldiers with him were in high spirits. There was finally hope they'd live to see the end of the war. So thick are the inmates that I can hardly write for pokes, nudges and jolts. On my left, the company commander snores on a bench. It is a great life. I am more oblivious than alas, yourself, dear mother, of the ghastly glimmering of the guns outside and the hollow crashing of the shells. At 11:00 on November 11, 1918, the war ended. In the war's final week, their son Wilfred had been killed, shot in one of the last assaults on the German lines. Wilfred Owen is known as one of his nation's greatest poets. The loss of such a promising life was a tragedy.

Figure 18. Relationship among the different lengths of summaries



versions with various lengths of video summaries automatically. To improve the proposed scheme more, we will address the following issues:

- Currently, we are using 10 seconds as a threshold value for paragraph segmentation. This value cannot be universal for all different types of videos. We will study an algorithm to find an optimal value for each video since one value for all types is not practical.

Table 3. Different summarization ratios of six test videos

Text Percentage	**5%**	**10%**	**20%**	**50%**	**100%**
The Great War	00:02:47	00:05:20	00:10:05	00:24:24	00:58:34
Red Monkey of Zanzibar	00:00:42	00:02:01	00:03:22	00:09:00	00:26:52
Solar Blast	00:02:31	00:04:29	00:09:13	00:15:25	00:57:07
Nature’s Cheats	00:00:48	00:02:05	00:03:58	00:12:31	00:26:40
Poison Dart Frogs	00:00:39	00:02:08	00:03:28	00:09:35	00:27:12
Brooklyn Bridge	00:03:06	00:05:28	00:09:45	00:27:50	01:01:16
Total	00:10:33	00:21:31	00:39:41	01:38:45	04:17:41

Figure 19. Summary result of “Red Monkey of Zanzibar” using ratio 5%

Shy and reclusive, these monkeys are lazing high in the trees the way forest monkeys have always lived. Zanzibar Island, 25 miles from the Tanzanian coast. Remarkably, every Shamba monkey seems to know that there's nothing like a piece of charcoal to ease indigestion. As more and more bush is destroyed, the duikers may have no place left to hide. A greater bush baby. In the forest, the smell of smoke is a smell of danger. Soon the whole troop finds the courage to taste the charred sticks.

Figure 20. Summary result of “Nature’s Cheats” using ratio 5%

Strong males fight for the right to win females. The plumage of the males varies in color. Not all females are won by fighting. Male Natterjack toads court their females by serenading them at dusk. Ant larvae and eggs are kept in a special chamber. Deep inside the reed bed, well hidden from predators, a reed warbler is brooding her eggs. The Nephila spider, goliath, has just caught a butterfly for lunch. As it moves from plant to plant, it unwittingly transfers the pollen, so fertilizing the lilies. The most theatrical con artist is a hog nosed snake. The indigo snake retreats in disgust. The lily trotter stays put. Hardly a cheat? The male bees home in and make frenzied attempts to mate with the flowers.

Figure 21. Summary result of “Poison Dart Frogs” using ratio 5%

The islands are thick with tropical vegetation. Most frogs stay hidden. Scientist Kyle Summers is not intimidated. This frog can be handled without any risk. It's called a strawberry poison dart frog. These frogs eat a lot of ants, and that's unusual. Brown tree frog males over-inflate their throats to amplify their calls. Poison frogs do things differently of course. So females on one island might prefer red males, whereas on another island they might prefer a green male. Banana plantations and coconut groves have replaced natural rainforest.

- The current data set of videos has around four hours of six documentaries. We will include other types of videos such as movies and TV dramas to which we will apply our scheme to generate various summaries.
- We will implement a prototype that processes the first step through the last step automatically.
- We will organize and represent the summary results using MPEG-7 standard and XML.

CONCLUDING REMARKS

We presented two types of video abstractions, video summary and video skimming, in this chapter. As we mentioned, video summary is a set of salient images (key frames) selected from an original video sequence. Video skimming, which is called a preview, consists of a collection of image sequences along with the corresponding audios from an original video sequence. It can be classified into two sub-types: highlight and summary sequence. The highlight has the most interesting and attractive parts of a video, while the summary sequence renders the impression of the content of the entire video. Among all types of video abstractions, summary sequence conveys the highest semantic meaning of the content of an original video.

We discussed a number of methods for video summary and video skimming, and introduced a new technique to generate video- summary sequences. In this new approach, the video segmentation is performed by a top-down fashion to reflect the content of the video. One of the natural language processing tools is used effectively to produce various lengths of different summaries. We tested the proposed approach based on four hours of documentary videos, and the test results provided the promising results. We will further investigate the issues mentioned in Future Work.

REFERENCES

Abdel-Mottaleb, M., & Dimitrova, N. (1996). CONIVAS: CONtent-based image and video access system. *Proceedings of ACM International Conference on Multimedia*, Boston, MA, 427-428.

Adjeroh, D.A., & Lee, M C. (1997). Adaptive transform domain video scene analysis. *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, Ottawa, Canada, 203-210.

Agnihotri, L. (2001). Summarization of video programs based on closed captions. *Proceedings of SPIE*, Vol.4315, San Jose, CA, 599-607.

Alexander, G. (1997). Informedia: News-on-demand multimedia information acquisition and retrieval. In M. Maybury (Ed.), *Intelligent Multimedia Information Retrieval*, pp. 213-239. Menlo Park, CA: AAAI Press.

Ardizzone, E., & Cascia, M. (1997). Automatic video database indexing and retrieval. *Multimedia Tools and Applications, 4,* 29-56.

Babaguchi, N. (2000). Towards abstracting sports video by highlights. *Proceedings of IEEE International Conference on Multimedia and Expo,* 2000 (ICME 2000), New York, 1519-1522.

Chang, S., Chen, W., Meng, H.J., Sundaram, H., & Zhong, D. (1997). VideoQ: An automated content-based video search system using visual cues. *ACM Proceedings of the Conference on Multimedia* '97, Seattle, Washington, 313-324.

Christel, M., Smith, M., Taylor, C., & Winkler, D. (1998). Evolving video skims into useful multimedia abstractions. *Proceedings of CHI 1998*, Los Angeles, CA, 171-178.

Corridoni, J.M., Bimbo, A.D., Lucarella, D., & Wenxue, H. (1996). Multi-perspective navigation of movies. *Journal of Visual Languages and Computing*, *7*, 445-466.

DeMenthon, D., Kobla, V., & Doermann, D. (1998). Video summarization by curve simplification. *Proceedings of ACM Multimedia 1998*, 211-218.

Dufaux, F. (2000). Key frame selection to represent a video. *Proceedings of IEEE 2000 International Conference on Image Processing*, Vancouver, BC, Canada, 275-278.

Ferman, A., Gunsel, B., & Tekalp, A. (1997). Object-based indexing of MPEG-4 compressed video. *Proceedings of SPIE*-3024, San Jose, CA, 953-963.

Fujimura, K., Honda, K., & Uehara, K. (2002). Automatic video summarization by using color and utterance information. *Proceedings of IEEE International Conference on Multimedia and Expo*, 49-52.

Gerum, C. (1996). *Automatic recognition of audio-cuts* (Automatische Erkennung von Audio-Cuts). Unpublished Master's thesis, University of Mannheim, Germany.

Girgensohn, A., & Boreczky, J. (2000). Time-constrained key frame selection technique. *Multimedia Tools and Applications*, *11*(3), 347-358.

Griffin, D.W., & Lim, J.S. (1984). Signal estimation from modified shot-time Fourier transform. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, ASSP-32(2), 236-243.

Gunsel, B., Ferman, A., & Tekalp, A. (1996). Video indexing through integration of syntactic and semantic features. *Proceedings of 3rd IEEE Workshop on Applications of Computer Vision(WACV'96)*, Sarasota, FL, 90-95.

Hammoud, R., & Mohr, R. (2000, Aug.). A probabilistic framework of selecting effective key frames from video browsing and indexing. *Proceedings of International Workshop on Real-Time Image Sequence Analysis*, Oulu, Finland, 79-88.

Hanjalic, A., & Zhang, H. (1999). An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Transaction on Circuit and Systems for Video Technology*, *9*(8), 1280-1289.

He, L., Sanocki, E., Gupta, A., & Grudin, J. (1999). Auto-summarization of audio-video presentations. *Proceedings of ACM Multimedia'99*, Orlando, FL, 489-493.

Jiang, H., & Elmagarmid, A. (1998). WVTDB - A semantic content-based video database system on the World Wide Web. *IEEE Transactions on Knowledge and Data Engineering*, *10*(6), 947-966.

Jiang, H., Helal, A., Elmagarmid, A.K., & Joshi, A. (1998). Scene change detection techniques for video database system. *Multimedia Systems*, 186-195.

Kang, H. (2001a). Generation of video highlights using video context and perception. *Proceedings of Storage and Retrieval for Media Databases*, SPIE,Vol. 4315, 320-329.

Kang, H. (2001b). A hierarchical approach to scene segmentation. *IEEE Workshop on Content-Based Access of Image and Video Libraries* (CBAIVL 2001), 65-71.

Kim, C., & Hwang, J. (2001). An integrated scheme for object-based video abstraction. *Proceedings of ACM Multimedia 2001*, Los Angeles, CA, 303-309.

Li, B., & Sezan, I. (2002). Event detection and summarization in American football broadcast video. *Proceedings of SPIE, Storage ad Retrieval for Media Databases*, 202-213.

Li, Y. (2001). Semantic video content abstraction based on multiple cues. *Proceedings of IEEE ICME 2001*, Japan.

Li, Y., Zhang, T., & Tretter, D. (2001). An overview of video abstraction techniques. Retrieved from the World Wide Web: *http://www.hpl.hp.com/techreports/2001/HPL-2001-191.html*

Lienhart, R., & Pfeiffer, S. (1997). Video abstracting. *Communications of the ACM*, *4*(12), 55-62.

Lienhart, R., Pfeiffer, S., & Effelsberg, W. (1996). The MoCA workbench: Support for creativity in movie content analysis. *Proceedings of the IEEE Int. Conference on Multimedia Systems '96*, Hiroshima, Japan.

Low, C.Y., Tian, Q., & Zhang, H. (1996). An automatic news video parsing, indexing and browsing system. *Proceedings of ACM International Conference on Multimedia*, Boston, MA, 425-426.

Nagasaka, A., & Tanaka, Y. (1991). Automatic video indexing and full-video search for object appearance. *Proceedings of the IFIP TC2/WG2.6, Second Working Conference on Visual Database Systems*, North-Holland, 113-127.

Neuburg, E.P. (1978). Simple pitch-dependent algorithm for high quality speech rate changing. *Journal of the Acoustic Society of America, 63*(2), 624-625.

Ngo, C.W., Pong, T.C., & Zhang, H.J. (2001, Oct.). On clustering and retrieval of video shots. *Proceedings of ACM Multimedia 2001*, Ottawa, Canada, 51-60.

Oh, J., & Hua, K.A. (2000). Efficient and cost-effective techniques for browsing and indexing large video databases. *Proceedings of ACM SIGMOD*, Dallas, TX, 415-426.

Oh, J., Hua, K. A., & Liang, N. (2000). A content-based scene change detection and classification technique using background tracking Sept 30 - Oct 3, San Jose, CA, 254-265.

Oh, J., & Sankuratri, P. (2002). Computation of motion activity descriptors in video sequences. In N. Mastorakis & V. Kluev (Eds.), *Advances in Multimedia, Video and Signal Processing Systems,* pp. 139-144. New York: WSEAS Press.

Omoigui, N., He, L., Gupta, A., Grudin, J., & Sanocki, E. (1999). Time-compression: System concerns, usage, and benefits. *Proceedings of ACM Conference on Computer-Human Interaction*, 136-143.

Orr, D. B. (1971). A perspective on the perception of time-compressed speech. In P. M. Kjldergaard, D. L. Horton, & J. J. Jenkins (Eds.), *Perception of Language*, pp. 108-119. Englewood Cliffs, NJ: Merrill.

Pfeiffer, S., Lienhart, R., Fischer, S., & Effelsberg, W. (1996). Abstracting digital movies automatically. *Journal of Visual Communication and Image Representation*, *7*(4), 345-353.

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, *1*, 244-256.

Rui, Y., Huang, T., & Mehratra, S. (1999). Constructing table-of-content for videos. *ACM Multimedia Systems*, *7*(5), 359-368.

Smith, M., & Kanade, T. (1997). Video skimming and characterization through the combination of image and language understanding. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* San Juan, Puerto Rico, 775-781.

Sun, X., Kankanhalli, M., Zhu, Y., & Wu, J. (1998). Content-based representative frame extraction for digital video. *Proceedings of IEEE Multimedia Computing and Systems'98,* Austin, TX, 190-193.

Sundaram, H., & Chang, S. (2000). Video scene segmentation using video and audio Features. *ICME2000*, 1145-1148.

Swanberg, D., Shu, C., & Jain, R. (1993). Knowledge-guided parsing in video databases. *Proceedings. of SPIE Symposium on Electronic Imaging: Science and Technology,* San Jose, CA, 13-24.

Taskiran, C., Amir, A., Ponceleon, D., & Delp, E. (2002). Automated video summarization using speech transcripts. *Proceedings of SPIE*, Vol. 4676, 371-382.

Taskiran, C., & Delp, E. J. (1998). Video scene change detection using the generalized trace. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP*), Seattle, Washington, 2961-2964.

Truong, B., Dorai, C., & Venkatesh, S. (2000). New enhancements to cut, fade and dissolve detection processes in video segmentation. *Proceedings of ACM Multimedia 2000*, Los Angeles, CA, 219-227.

Uchihashi, S. (1999). Video Manga: Generating semantically meaningful video summaries. *Proceedings of ACM Multimedia'99*, Orlando, FL, 383-392.

Wactlar, H., Christel, M., Gong, Y., & Hauptmann, A. (1999). Lessons learned from building terabyte digital video library. *Computer*, 66-73.

Wang, J., & Chua, T. (2002). A framework for video scene boundary detection. *Proceedings of the 10th ACM International Conference on Multimedia*, Juan-les-Pins, France, 243-246.

Wolf, W. (1996). Key frame selection by motion analysis. *Proceedings of IEEE International. Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, 1228-1231.

Yfantis, E.A. (2001). An algorithm for key-frame determination in digital video. *Proceedings of 16th ACM Symposium on Applied Computing (SAC 2001)*, Las Vegas, 312-314.

Yoshitaka, A., Hosoda, Y., Hirakawa, M., & Ichikawa, T. (1998). Content-based retrieval of video data based on spatiotemporal correlation of objects. *Proceedings of 1998 IEEE Conference on Multimedia Computing and Systems*, Austin, TX, 208-213.

Yu, H., & Wolf, W. (1997). A visual search system for video and image databases. *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, Ottawa, Canada, 517-524.

Zabih, R., Miller, J., & Mai, K. (1995). A feature-based algorithm for detecting and classifying scene breaks. *Proceedings of the Third ACM International Conference on Multimedia*, San Francisco, CA, 189-200.

Zhang, H.J. (1997). An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, *30*(4), 643-658.

Zhao, L., Qi, W., Li, S., Yang, S., & Zhang, H. (2000). Key-frame extraction and shot retrieval using nearest feature line (NFL). *Proceedings of ACM Multimedia Workshop* 2000, Los Angeles, CA, 217-220.

Zhong, D., Zhang, H., & Chang, S. (1997). *Clustering methods for video browsing and annotation.* Columbia University.

Zhuang, Y., Rui, Y., Huang, T., & Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. *Proceedings of International Conference on Image Processing*, Chicago, IL, 870-886.

Chapter XV

Video Summarization Based on Human Face Detection and Recognition

Hong-Mo Je, Pohang University of Science and Technology, Korea

Daijin Kim, Pohang University of Science and Technology, Korea

Sung-Yang Bang, Pohang University of Science and Technology, Korea

ABSTRACT

In this chapter, we deal with video summarization using human facial information by face detection and recognition. Many methods of face detection and face recognition are introduced as both theoretical and practical aspects. Also, we describe the real implementation of the video summarization system based on face detection and recognition

INTRODUCTION

The growing availability of multimedia data such as video and home equipment creates a strong requirement for efficient tools to manipulate this type of data. Automatic summarization is one such tool that automatically creates a short version or subset of key frames that contains as much information as possible from the original video. Summaries are important because they can rapidly provide users with some information about the content of a large video or set of videos. From a summary, the user should be able to evaluate if a video is interesting or not, for example, if a documentary contains a certain topic, or a film takes place partly in a certain location.

Automatic summarization is the subject of very active research, and several approaches have been proposed to define and identify what is the most important content in a video. Many researchers and engineers have been actively developing technologies for video summarization in order to facilitate efficient management, exchange, and consumption of digital videos. The goal of video summarization is to obtain a compact representation of the original video that usually contains large volume of data. Oh and Hua (2000) stated that there are two methods of video summarization. One is to extract key frames from each shot or scene and present them as the summary of the video. This approach is good for quick browsing of lengthy videos. The other way is to extract its “interesting” or “important” scenes using content-based features such as caption on shot, audio, or visual information. The human face can play an important role in indexing key information for video summarization because it is a unique feature of human beings and it is ubiquitous in TV news, dramas, documentaries, and movie videos. So it can also be salient feature to represent importance of video shot.

In this chapter, we introduce an application of video summarization to extract interesting/important scenes using human face detection and recognition and show how to implement the system. The target person for the system can be a particular person in video sequences, such as an anchor of news or the main actor in a drama or movie. The first step of this application, face detection, is presented in the second section. In the third section, face recognition is reviewed. Then, we describe implementation of the proposed system. Experimental results are shown in the fifth section. Finally, concluding remarks and future works are summarized in the last section.

FACE DETECTION

A face detection problem can be defined as follows: an arbitrary image can be a digitized video signal or a scanned photograph as input, so we must determine whether there are any human faces in the image or not, and if so, report their location. A first step of any face detection system is detecting the locations in images where faces are present. However, face detection from a single image is a challenging task because of variability in scale, location, orientation (upright, rotated), and pose (frontal, profile). Facial expression, occlusion, and lighting conditions also change the overall appearance of faces. The challenges associated with face detection can be attributed to the following factors:

- *Pose:* The images of a face vary due to the relative camera-face pose (frontal, 45 degrees, profile, upside-down), and some facial features (such as an eye or the nose) may become partially or wholly occluded. .
- *Presence or absence of structural components:* Facial features such as beards, mustaches, and glasses may or may not be present, and there is a great deal of variability among these components, including shape, color, and size.
- *Facial expression:* The appearance of faces are directly affected by a person’s facial expression.

- *Occlusion:* Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.
- *Image orientation:* Face images directly vary for different rotations about the camera's optical axis.
- *Imaging conditions:* When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

Yang (2002) has classified face detection methods into four categories on his survey report. The summary of the report is as follows:

Knowledge-based Top-down Methods

These rule-based methods encode human knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features. These methods are designed mainly for face localization. In this approach, face detection methods are developed based on the rules derived from the researcher's knowledge of human faces. It is easy to come up with simple rules to describe the features of a face and their relationships. For example, a face often appears in an image with two eyes that are symmetric to each other, a nose, and a mouth. The relationships between features can be represented by their relative distances and positions.

Facial features in an input image are extracted first, and face candidates are identified based on the coded rules. A verification process is usually applied to reduce false detections. One problem with this approach is the difficulty in translating human knowledge into well-defined rules. If the rules are detailed (i.e., strict), they may fail to detect faces that do not pass all the rules. If the rules are too general, they may give many false positives. Moreover, it is difficult to extend this approach to detect faces in different poses since it is challenging to enumerate all possible cases.

On the other hand, heuristics about faces work well in detecting frontal faces in uncluttered scenes. Yang and Huang (1994) used a hierarchical knowledge-based method to detect faces. Kotropoulos and Pitas (1997) presented a rule-based localization method that is similar to that of Kanade (1973) and Yang and. Huang (1994). First, facial features are located with a projection method that Kanade (1973) successfully used to locate the boundary of a face.

Bottom-up Feature-based Methods

These methods aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to locate faces. These methods are designed mainly for face localization. In contrast to the knowledge-based top-down approach, researchers have been trying to find invariant features of faces for detection. The underlying assumption is based on the observation that humans can effortlessly detect faces and objects in different poses and lighting conditions and so, there must exist properties or features that are invariant over these variabilities. Numerous methods have been proposed to first detect facial features and then to infer the presence of a face. Facial features such as eyebrows, eyes, nose, mouth, and hair-line are commonly extracted using edge detectors. Based on the extracted features, a statistical model is built to describe their relationships and to verify the existence of a face. One problem with these feature-based algorithms is that the image features can be severely corrupted due to illumination, noise, and occlusion. Feature boundaries can be weakened for faces, while shadows can cause numerous strong edges that together render perceptual grouping algorithms useless.

Sirohey (1993) proposed a localization method to segment a face from a cluttered background for face identification. It uses an edge map via Canny Detector (Canny, 1986) and heuristics to remove and group edges so that only the ones on the face contour are preserved. An ellipse is then fit to the boundary between the head region and the background. Graf, Chen, Petajan, and Cosatto (1995) developed a method to locate facial features and faces in gray-scale images. After band pass filtering, morphological operations are applied to enhance regions with high intensity that have certain shapes (e.g., eyes). Leung, Burl, and Perona (1995) developed a probabilistic method to locate a face in a cluttered scene based on local feature detectors and random graph matching. Their motivation was to formulate the face localization problem as a search problem in which the goal was to find the arrangement of certain facial features that is most likely to be a face pattern.

Yow and Cipolla (1996, 1997) presented a feature-based method that used a large amount of evidence from the visual image and their contextual evidence. Takacs and Wechsler (1995) described a biologically motivated face localization method based on a model of retinal feature extraction and small oscillatory eye movements. Recently, Amit, Geman, and Jedynak (1998) presented a method for shape detection and applied it to detect frontal-view faces in still intensity images.

Human faces have a distinct texture that can be used to separate them from different objects. Augusteijn and Skujca (1993) developed a method that infers the presence of a face through the identification of face-like textures. Human skin color has been used and proven to be an effective feature in many applications from face detection to hand tracking. Although different people have different skin color, several studies have shown that the major difference lies largely between their intensity rather than their chrominance (Graf, Chen, Petajan, & Cosatto, 1995, 1996; Yang & Waibel, 1996). Several color spaces have been utilized to label pixels as skin including RGB (Jebara & Pentland, 1997, 1998; Satoh, Nakamura, & Kanade, 1999), Normalized RGB (Crowley &. Bedrune, 1994; Crowley &. Berard, 1997; Kim, Kim, Ahn, & Kim, 1998; Oliver, Pentland, & Berard, 1997; Qian, Sezan, & Matthews, 1998; Starner & Pentland, 1996; Yang, Stiefelhagen, Meier, &. Waibel, 1998; Yang & Waibel, 1996), HSV or HSI (Kjeldsen &. Kender, 1996; Saxe & Foulds, 1996; Sobottka & Pitas, 1996), YCrCb (Chai & Ngan, 1998; Wang & Chang, 1997), YIQ (Dai & Nakano, 1995), YES (Saber & Tekalp, 1998), CIE XYZ (Chen, Wu, & Yachida, 1995), and CIE LUV (Yang & Ahuja, 1998).

Wu, Yokoyama, Pramadihanto, and Yachida (1996) and Wu, Chen, and Yachida (1999) presented a method to detect faces in color images using fuzzy theory. They used two fuzzy models to describe the distribution of skin and hair color in CIE XYZ color space. Five (one frontal and four side views) head-shape models are used to abstract the appearance of faces in images. Sobottka and Pitas (1996) proposed a method for face localization and facial feature extraction using shape and color. First, color segmentation in HSV space is performed to locate skin-like regions. Range and color have also been employed for face detection by Kim et al. (1998). Disparity maps are computed and objects are segmented from the background with a disparity histogram using the assumption that background pixels have the same depth and outnumber the pixels in the foreground objects. Using a Gaussian distribution in normalized RGB color space, segmented regions with a skin-like color are classified as faces. A similar approach has been proposed by Darrell, Gordon, Harville, and Woodfill (2000) for face detection and tracking.

Template Matching

Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the stored patterns are computed for detection. These methods have been used for both face localization and detection.

In template matching, a standard face pattern (usually frontal) is manually predefined or parameterized by a function. Given an input image, the correlation values with the standard patterns are computed for the face contour, eyes, nose, and mouth independently. The existence of a face is determined based on the correlation values. This approach has the advantage of being simple to implement. However, it has proven to be inadequate for face detection since it cannot effectively deal with variation in scale, pose, and shape. Multiresolution, multiscale, sub-templates, and deformable templates have subsequently been proposed to achieve scale and shape invariance.

Craw, Ellis, and Lishman (1987) presented a localization method based on a shape template of a frontal-view face (i.e., the outline shape of a face). Later, Craw, Tock, and Bennett (1992) described a localization method using a set of 40 templates to search for facial features and a control strategy to guide and assess the results from the template-based feature detectors.

Tsukamoto, Lee, and Tsuji (1993, 1994) presented a qualitative model for face pattern (QMF). In QMF, each sample image is divided into a number of blocks, and qualitative features are estimated for each block. Sinha (1994, 1995) used a small set of spatial image invariants to describe the space of face patterns. A hierarchical template matching method for face detection was proposed by Miao, Yin, Wang, Shen, and Chen (1999). Yuille et al. (**???**) used deformable templates to model facial features that fit an *a priori* elastic model to facial features (e.g., eyes). In this approach, facial features are described by parameterized templates. Kwon and da Vitoria Lobo (1994) introduced a detection method based on snakes (Kass, Witkin, and Terzopoulos, 1987; Leymarie & Levine, 1993) and templates were developed. An image is first convolved with a blurring filter and then a morphological operator to enhance edges. Lanitis, Taylor, and Cootes (1995) described a face representation method with both shape and intensity information. They started with sets of training images in which sampled contours such as the eye boundary, nose, chin/cheek were manually labeled, and a vector of sample points was used to represent shape.

Appearance-based Method

In contrast to template matching, the models (or templates) are learned from a set of training images that should capture the representative variability of facial appearance. These learned models are then used for detection. These methods are designed mainly for face detection. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face and non-face images. The learned characteristics are in the form of distribution models or discriminant functions that are consequently used for face detection. Meanwhile, dimensionality reduction is usually carried out for the sake of computation efficiency and detection efficacy. Many appearance-based methods can be understood in a probabilistic framework. An image or feature vector derived from an image is viewed as a random variable x, and this random variable is characterized for faces and non-faces by the class-conditional density functions $p(x \mid face)$ and $p(x \mid nonface)$. Bayesian classification or maximum likelihood can be used to classify a candidate image location as face or non-face. Unfortunately, a straightforward implementation of Bayesian classification is infeasible because of the high dimensionality of x, because $p(x \mid face)$ and $p(x \mid nonface)$ are multimodal, and because it is not yet understood if there are natural parameterized forms for $p(x \mid face)$ and $p(x \mid nonface)$. Hence, much of the work in an appearance-based method concerns empirically validated parametric and nonparametric approximations to $p(x \mid face)$ and $p(x \mid nonface)$. Another approach in appearance-based methods by Turk and Pentland (1991) finds a discriminant function (i.e., decision surface, separating hyperplane, threshold function) between face and non-face classes. Conventionally, image patterns are projected to a lower dimensional space and then a discriminant function is formed (usually based on distance metrics) for classification, or a nonlinear decision surface can be formed using multilayer neural networks by Rowley, Baluja, and Kanade (1998).

Recently, Osuna, Freund, and Girosi (1997) have proposed support vector machines and other kernel-based face detection methods. These methods implicitly project patterns to a higher dimensional space and then form a decision surface between the projected face and non-face patterns. Turk and Pentland (1991) applied principal component analysis to face recognition and detection. Similar to Kirby and Sirovich (1990), principal component analysis on a training set of face images is performed to generate the Eigen pictures (here called Eigenfaces) that span a subspace (called the face space) of the image space. Images of faces are projected onto the subspace and clustered. Similarly, non-face training images are projected onto the same subspace and clustered. Images of faces do not change radically when projected onto the face space, while the projection of non-face images appears quite different. To detect the presence of a face in a scene, the distance between an image region and the face space is computed for all locations in the image. The distance from face space is used as a measure of "faceness," and non-face clusters used by Sung and Poggio (1998). Their method estimates density functions for face and non-face patterns using a set of Gaussians. The centers of these Gaussians are shown on the right.

Recently, methods for video processing applications (for example, content-based video indexing, etc.) also have been introduced (Rowley, Baluja, & Kanade, 1998; Satoh, Nakamura, & Kanade, 1999). Our proposed method, which can automatically detect a human face on video, is presented in Je, Kim, and Bang (2003). The method consists of two stages that use a SVM ensemble (Kim, Pang, Je, Kim, & Bang, 2002) for their individual task. The first stage performs a skin-block classification that classifies the 8X8 block in each shot digital video into skin/non-skin block using skin color and texture information. This stage generates several face candidate regions by connecting and filling the skin blocks with a connected component analysis and morphological filtering. The second stage performs a face/non-face classification using another classifier that verifies the true faces among the face candidates. More detail structure will be described in a later section.

FACE RECOGNITION

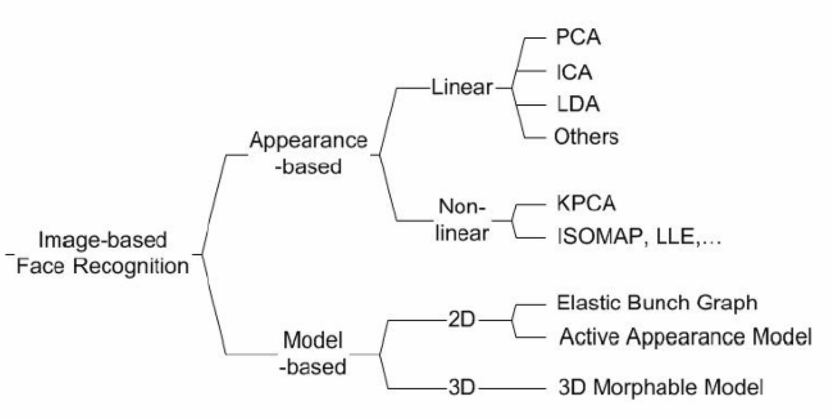
In recent years, face recognition has received substantial attention from researchers in biometrics, pattern recognition, and computer vision communities (Chellappa, Wilson, & Sirohey, 1995; Gong, McKenna, and Psarrou, 2000; Wechsler, Phillips, Bruce, Soulie, & Huang, 1996; Zhao, Chellappa, Rosenfeld, & Phillips, 2000). The machine learning and computer graphics communities are also increasingly involved in face recognition. This common interest among researchers working in diverse fields is motivated by our remarkable ability to recognize people and the fact that human activity is a primary concern both in everyday life and in cyberspace, for example, in Human Computer Interaction (HCI), multimedia communication (e.g., generation of synthetic faces), and content-based image database management.

A number of face recognition algorithms, along with their modifications, have been developed during the past several decades as seen Figure 1. In this section, we discuss two image-based face recognition techniques that are model-based (which uses holistic texture features) and appearance-based (which uses employ shape and texture of the face) methods. Now, we summarize many face recognition algorithms introduced in Xiaoguang (2003).

Appearance-based (View-based) Face Recognition

Many approaches to object recognition and to computer graphics are based directly on images without the use of intermediate 3-D models. Most of these techniques depend on a representation of images that induces a vector space structure and, in principle, requires dense correspondence. Appearance-based approaches represent an object in terms of several object views (raw intensity images). An image is considered as a high-dimensional vector, that is, a point in a high-dimensional vector space. Many view-based

Figure 1. Face recognition methods



approaches use statistical techniques to analyze the distribution of the object image vectors in the vector space, and derive an efficient and effective representation (feature space) according to different applications. Given a test image, the similarity between the stored prototypes and the test view is then carried out in the feature space. This image vector representation allows the use of learning techniques for the analysis and for the synthesis of images. Face recognition can be treated as a space-searching problem combined with a machine-learning problem.

Linear (Subspace) Analysis

Three classical linear appearance-based classifiers, Principal Component Analysis (PCA) (Turk and Pentland, 1991), Independent Component Analysis (ICA) (Bartlett, Lades, & Sejnowski, 1998), and LDA (Belhumeur, Hespanha, & Kriegman, 1997; Swets & Weng, 1996) are introduced in the following. Each classifier has its own representation (basis vectors) of a high-dimensional face vector space based on different statistical viewpoints. By projecting the face vector to the basis vectors, the projection coefficients are used as the feature representation of each face image. The matching score between the test face image and the training prototype is calculated (e.g., as the cosine value of the angle) between their coefficients vectors. The larger the matching score, the better the match. All three representations can be considered as a linear transformation from the original image vector to a projection feature vector, that is, $Y = W^T X$ where Y is the $d \times N$ feature vector matrix, d is the dimension of the feature vector, and W is the transformation matrix. Note that d <<N.

Principal Component Analysis

The Eigenface algorithm uses the Principal Component Analysis (PCA) for dimensionality reduction to find the vectors that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images, and the subspace is called face space. All faces in the training set are projected onto the face space to find a set of weights that describes the contribution of each vector in the face space. To identify a test image, the projection of the image onto the face space is required to obtain the corresponding set of weights. By comparing the set of weights of the faces in the training set, the testing image can be identified.The key procedure in PCA is based on Karhumen-Loeve transformation (Kirby & Sirovich, 1990). If the image elements are considered to be random variables, the image may be seen as a sample of a stochastic process. The Principal Component Analysis basis vectors are defined as the eigenvectors of the scatter matrix S_T,

$$S_T = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T$$

The transformation matrix W_{PCA} is composed of the eigenvectors corresponding to the d largest eigenvalues. Several extensions of PCA are developed, such as modular eigenspaces (Moghaddam, 2002) and probabilistic subspaces (Hyvrinen, 1999).

Independent Component Analysis

Independent Component Analysis (ICA) is similar to PCA except that the distribution of the components are designed to be non-Gaussian. Maximizing non-Gaussianity promotes statistical independence. Bartlett, Lades, and Sejnowski (1998) provided two architectures based on Independent Component Analysis, statistically independent basis images and a factorial code representation, for the face recognition task. The ICA separates the high-order moments of the input in addition to the second-order moments utilized in PCA. Both architectures lead to a similar performance. The obtained basis vectors are based on fast fixed-point algorithm (Hyvrinen, 2000) for the ICA factorial code representation.

LD Analysis

Both PCA and ICA construct the face space without using the face class (category) information. The whole face training data is taken as a whole. In LD Analysis (LDA), the goal is to find an efficient or interesting way to represent the face vector space,but exploiting the class information can be helpful to the identification tasks. The Fisherface algorithm (Belhumeur et al., 1997) is derived from the Fisher Linear Discriminant (FLD), which uses class-specific information. By defining different classes with different statistics, the images in the learning set are divided into the corresponding classes. Then, techniques similar to those used in Eigenface algorithm are applied. The Fisherface algorithm results in a higher accuracy rate in recognizing faces when compared with Eigenface algorithm.

Non-linear (Manifold) Analysis

The face manifold is more complicated than linear models. Linear subspace analysis is an approximation of this non-linear manifold. Direct non-linear manifold modeling schemes are explored to learn this non-linear manifold. In the following subsection, the kernel principal component analysis (KPCA) is introduced and several other manifold learning algorithms are also listed.

Kernel PCA

The kernel PCA-based method by Scholkopf, Smola, and Muller (1998) applies a nonlinear mapping from the input space $\mathbb{R}^N$ to the feature space $\mathbb{R}^L$, denoted by $\psi(x)$, where L is larger than N. This mapping is made implicit by the use of kernel functions satisfying the Mercer's theorem

$$k(x_i, x_j) = \psi(x_i) \bullet \psi(x_j).$$

where kernel functions $k(x_i, x_j)$ in the input space correspond to inner-product in the higher dimensional feature space. Because computing covariance is based on inner-products, performing a PCA in the feature space can be formulated with kernels in the input space without the explicit computation of $\psi(x)$. Suppose the covariance in the

$$K = < \psi(x_i) \psi(x_j) >^T$$

Similar to traditional PCA, the projection coefficients are used as features for classification. Yang (2002) explored the use of KPCA for the face recognition problem. Unlike traditional PCA, KPCA can use more eigenvector projections than the input dimensionality, but a suitable kernel and correspondent parameters can only be decided empirically.

Model-based Face Recognition

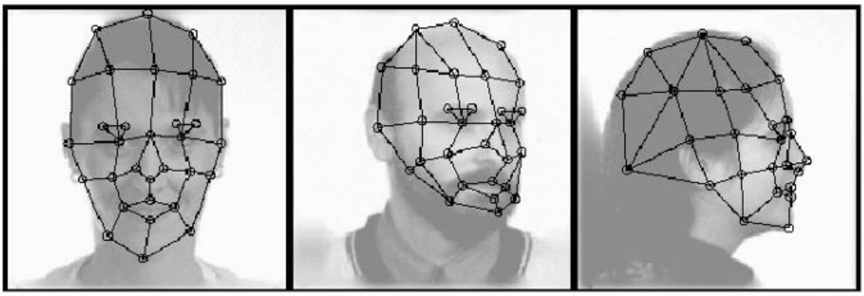
The model-based face recognition scheme is aimed at constructing a model of the human face, which is able to capture the facial variations. The prior knowledge of human face is highly utilized to design the model. For example, feature-based matching derives distance and relative position features from the placement of internal facial elements (e.g., eyes, etc.). Kanade (1973) developed one of the earliest face recognition algorithms based on automatic feature detection. By localizing the corners of the eyes, nostrils, etc., in frontal views, his system computed parameters for each face that were compared (using a Euclidean metric) against the parameters of known faces. A more recent feature-based system, based on elastic bunch graph matching, was developed by Wiskott, Fellous, Krüger, and von der Malsburg (1997) as an extension to their original graph matching system. By integrating both shape and texture, Cootes, Edwards, and Taylor (2001) developed a 2-D morphable face model, through which the face variations are learned. A more advanced 3-D morphable face model is explored to capture the true 3-D structure of human face surface. Both morphable model methods come under the framework of "interpretation through synthesis." The model-based scheme usually contains three steps: (1) constructing the model; (2) fitting the model to the given face image; and (3) using the parameters of the fitted model as the feature vector to calculate the similarity between the query face and prototype faces in the database to perform the recognition.

Feature-based Elastic Bunch Graph Matching

Bunch Graph

All human faces share a similar topological structure. Wiskott et al. (1997) presented a general in-class recognition method for classifying members of a known class of objects. Faces are represented as graphs, with nodes positioned at facial points (such as the eyes, the tip of the nose, some contour points, etc.; see Figure 2 and edges labeled

Figure 2. Multiview faces overlaid with labeled graphs (Wiskott et al., 1997)



with 2-D distance vectors). Each node contains a set of 40 complex Gabor wavelet coefficients, including both phase and magnitude, known as a jet . Wavelet coefficients are extracted using a family of Gabor kernels with five different spatial frequencies and eight orientations; all kernels are normalized to be of zero mean. Face recognition is based on labeled graphs. A labeled graph is a set of nodes connected by edges; nodes are labeled with jets; edges are labeled with distances. Thus, the geometry of an object is encoded by the edges while the gray value distribution is patch-wise encoded by the nodes (jets).

Elastic Graph Matching

To identify a new face, the face graph is positioned on the face image using elastic bunch graph matching. The goal of elastic graph matching is to find the fiducial points on a query image and thus to extract from the image a graph that maximizes the graph similarity function. This is performed automatically if the face bunch graph (FBG) is appropriately initialized. A face bunch graph consists of a collection of individual face model graphs combined into a stack-like structure, in which each node contains the jets of all previously initialized faces from the database. To position the grid on a new face, the graph similarity between the image graph and the existing FBG is maximized. Let S_Φ be the similarity between two jets, defined as

$$S_\Phi(J,J') = \frac{\sum_j a_j a'_j \cos(\Phi_j - \Phi'_j - \overrightarrow{dk_j})}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}$$

where a_j and Φ_j are magnitude and phase of the Gabor coefficients in the j-th jet, respectively $\overline{d}$ is the displacement between locations of the two jets. $\overrightarrow{k_j}$ determines the wavelength and orientation of the Gabor wavelet kernels. After the grid has been positioned on the new face, the face is identified by comparing the similarity between that face and every face stored in the FBG. Graphs can be easily translated, rotated, scaled, and elastically deformed, thus compensating for the variance in face images that is commonly encountered in a recognition process.

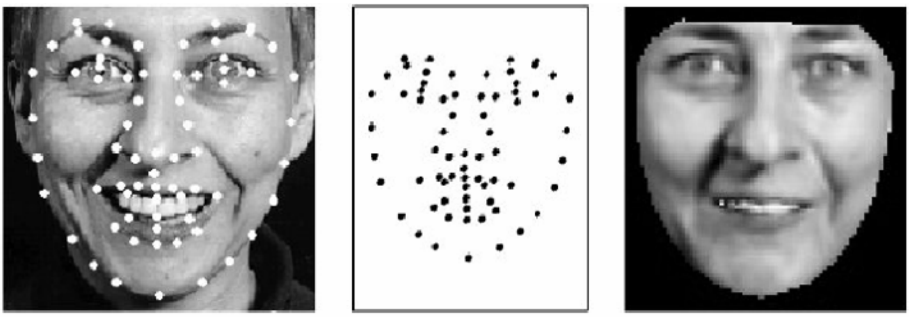
Active Appearance Model (AAM)

An Active Appearance Model (AAM) is an integrated statistical model that combines a model of shape variation with a model of the appearance variation in a shape-normalized frame. An AAM contains a statistical model of the shape and gray-level appearance of the object of interest that can generalize to almost any valid example. Matching to an image involves finding model parameters that minimize the difference between the image and a synthesized model example, projected into the image. The potentially large number of parameters makes this a difficult problem.

AAM Construction

The AAM is constructed based on a training set of labeled images, where landmark points are marked on each example face at key positions to outline the main features

Figure 3. The training image is split into shape and shape-normalized texture (Cootes et al., 2001)

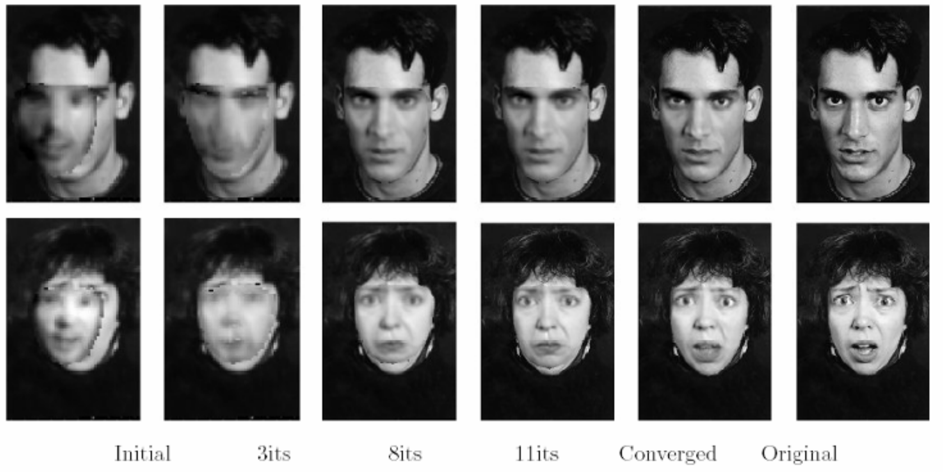


(shown in Figure 3). The shape of a face is represented by a vector consisting of the positions of the landmarks, $s = (x_1, y_1, \ldots, x_n, y_n)^T$, where (x_j, y_j) denotes the 2-D image coordinate of the j-th landmark point. All shape vectors of faces are normalized into a common coordinate system. The principal component analysis is applied to this set of shape vectors to construct the face shape model, denoted $s = \bar{s} + P_s b_s$, where $\bar{s}$ is a shape vector, $\bar{s}$ is the mean shape, P_S is a set of orthogonal modes of shape variation and b_S is a set of shape parameters. In order to construct the appearance model, the example image is warped to make the control points match the mean shape. Then the warped image region covered by the mean shape is sampled to extract the gray-level intensity (texture) information. Similar to the shape model construction, a vector is generated as the representation, $g = (I_1, \ldots, I_m)^T$,where I_j denotes the intensity of the sampled pixel in the warped image. PCA is also applied to construct a linear model $g = \bar{g} + P_g b_g$, where $\bar{g}$ is the mean appearance vector, P_g is a set of orthogonal modes of gray-level variation and b_g is a set of gray-level model parameters. Thus, all shape and texture of any example face can be summarized by the vectors b_S and b_g.The combined model is the concatenated version of b_S and b_g, denoted as follows:

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s P_s^T (s - \bar{s}) \\ P_g^T (g - \bar{g}) \end{pmatrix}$$

where W_S is a diagonal matrix of weights for each shape parameter, allowing for the difference in units between the shape and gray-scale models. PCA is applied to b also, b = Q_c, where c is the vector of parameters for the combined model.

Figure 4. Examples of the AAM fitting iterations (Cootes et al, 2001)



AAM Fitting

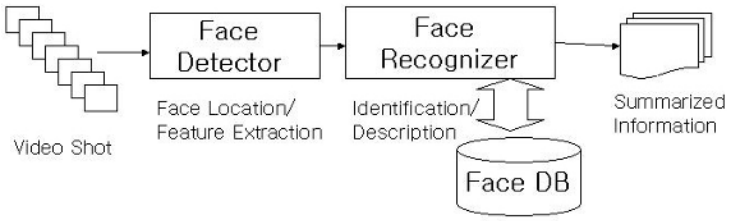
Given a new image and constructed model, the metric used to measure the match quality between the model and image is $\Delta = |\delta I|^2$, where δI is the vector of intensity differences between the given image and the image generated by the model tuned by the model parameters, called residules. The AAM fitting seeks the optimal set of model parameters that best describes the given image. Cootes et al. (1998) observed that displacing each model parameter from the correct value induces a particular pattern in the residuals. In the training phase, the AAM learned a linear model that captured the relationship between parameter displacements and the induced residuals. During the model fitting, it measures the residuals and uses this model to correct the values of current parameters, leading to a better fit. Figure 4 shows examples of the iterative AAM fitting process.

VIDEO SUMMARIZATION SYSTEM BASED ON HUMAN FACE INFORMATION

Both overviews and methods for human face detection and recognition have been presented previous sections. In this section, we will describe the detail structure, operation of the video summarization system using human face, and actual implementation. The system is comprised of three major components — the Face Detector, the Face Recognizer and the FaceDB.

Figure 5 illustrates the block diagram of the system. The Face Detector accepts a single frame from video sequences as input and returns a location of the human face in the frame if one exists there. Then, it extracts a few principle components projected on eigenfaces as the feature vector for the Face Recognizer. The FaceDB contains a set of

Figure 5. Block diagram of video summarization system based on the human face



face description feature vectors. The Face Recognizer find the best matched person with the query feature vectors that were given by Face Detector. Then, it indexes the face descriptor for the current frame and notes some useful information, such as frame number, face location, and person name (if it was given), etc. Later, this information is used for summarization. More detail manuscript will be described.

Face Detector

The Face Detector of the proposed system has an ensemble method as face detection. In this paragraph, we explain theoretical background and constructions of Support Vector Machine (SVM) ensemble-based face detection.

Support Vector Machine Ensemble

Basically, a concept of the ensemble is a kind of multiple classifier system (MCS) based on the combination of outputs of a set of different classifiers. Therefore,an ensemble of classifiers is a collection of several classifiers whose individual decisions are combined in some way to classify the test examples. In the field of pattern recognition, it is known that an ensemble often gives a much better performance than the individual classifiers that compose it.

Haris and Ganapathy (2000), Giacinto, Roli, and Bruzzone (2000), Hansen and Salamon (1990), and many other researchers have investigated the neural network ensemble. Moreover, using an idea similar to a neural network ensemble, a genetic fuzzy predictor ensemble was proposed by Kim and Kim (1997). They provided a high-classification performance for various applications, such as image classification, time serious prediction, and financial trend analysis.

Hansen and Salamon (1990) show why the ensemble gives a better performance than individual classifiers as follows. Assume that an ensemble of n classifiers exists: $\{f_1, f_2, ..., f_n\}$ and consider a test data x. If all the classifiers are identical, they are wrong at the same data, where an ensemble will show the same performance as individual classifiers. However, if the classifiers are different and their errors are uncorrelated, then when $f_i(x)$ f_i (x) is wrong, most other classifiers except for $f_i(x)$ may be correct. Thus, the result of majority voting can be correct. More precisely, if the error of individual classifier is $p < 1 / 2$ and the errors are independent, then the probability p_E that the result of majority voting is incorrect is

$$\sum_{k=n/2}^{n} p^{k}(1-p)^{n-k} \left(< \sum_{k=n/2}^{n} \left(\frac{1}{2}\right)^{k} \left(\frac{1}{2}\right)^{n-k} = \sum_{k=n/2}^{n} \left(\frac{1}{2}\right)^{n} \right).$$

When the size of classifiers n is infinite, the probability p_E goes to zero.

Burges (1998) stated that the SVM has been known to show a good generalization performance and is easy to learn the exact parameters for the global optimum. So the ensemble of SVMs may not be considered as a method to improve the classification performance greatly. However, since a practical SVM has been implemented using the approximated algorithms in order to reduce the computation complexity of time and space, a single SVM may not learn the exact parameters for the global optimum. Sometimes, the support vectors obtained from the learning is not sufficient to classify all unknown test examples completely. Therefore, we cannot guarantee that a single SVM always provides the global optimal classification performance over all test examples.

To overcome this limitation, we propose to use an ensemble of support vector machines. Similar arguments are mentioned above about the general ensemble of classifiers and can also be applied to the ensemble of support vector machines. Figure 6 shows a general architecture of the proposed SVM ensemble

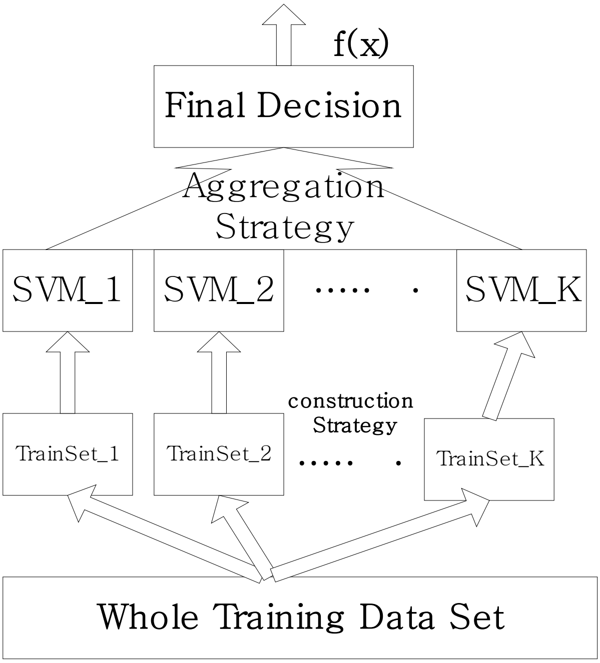
Methods for Constructing the SVM Ensemble

Many methods for constructing an ensemble of classifiers have been developed. We apply some general methods among them to construct a SVM ensemble. The main point of constructing SVM ensembles is that each individual SVM is as different from one another as possible. A well-known approach is to use different training sets for different SVMs. Its representative methods are bagging and boosting.

First , we will explain a bagging technique to construct the SVM ensemble. In bagging, several SVMs are trained independently via a bootstrap method and they are aggregated via an appropriate combination technique. Usually, we have a single training set $TR = \{x_i; y_i \mid i = 1,2,...,l\}$, but we need K training samples sets to construct the SVM ensemble with K independent SVMs. From this statistical fact, we need to make the training sample sets as different as possible in order to obtain higher improvement of the aggregation result. To accomplish this, we often use the bootstrap technique as follows. Bootstrapping builds K replicate training data sets $\{TR_k^B \mid k = 1,2,..., K\}$ by randomly re-sampling, but with replacement, from the given training data set TR repeatedly. Each example x_i in the given training set TR may appear repeated times or not at all in any particular replicate training data set. Each replicate training set will be used to train a certain SVM.

The representative boosting algorithm is the AdaBoosting algorithm (Schapire, 1999). Like bagging, each support vector machine is trained using a different training set. There is probability distribution function $p_l(x)$ over all training samples x for the l-th iteration. For the l-th iteration, a training set for the l-th support vector machine is sampled with a replacement according to the distribution function $p_l(x)$. The l-th support vector machine is trained using that training set. The error rate ε_l of the l-th support vector

Figure 6. A general architecture of the SVM ensemble



machine on the training samples is computed and used to adjust the probability distribution $p_{l+1}(x)$ for the (l + 1)-th iteration.

Aggregating Support Vector Machines

After training, we need to aggregate several independently trained SVMs in an appropriate combination manner. We consider two types of combination techniques, such as the linear and nonlinear combination method. The linear combination method, as a linear combination of several SVMs, includes the majority voting and the LSE-based weighting. The majority voting and the LSE-based weighting are often used for bagging and boosting, respectively. A nonlinear method, as a nonlinear combination of several SVMs, includes the double-layer hierarchical combination that uses another upper-layer SVM to combine several lower-layer SVMs.

Majority voting is the simplest method for combining several SVMs.

Let f_k $(k = 1,2,...,K)$ be a decision function of the kth SVM in the SVM ensemble and C_j $(j = 1,2,...,C)$ denote a label of the j-th class. Next, let $N_j = \#\{k \mid f_k(x) = C_j\}$, i.e. the number of SVMs whose decisions are known to the jth class. Next, the final decision of the SVM ensemble $f_{maj}(x)$ for a given test vector x due to the majority voting is determined by $f_{maj}(x) = \arg\max_j N_j$.

LSE-based Weighting

The LSE-based weighting treats several SVMs in the SVM ensemble with different weights. Often, the weights of several SVMs are determined in proportion to their accuracies of classifications. Here, we propose to learn the weights using the LSE method as follows.

Let $f_k(k = 1,2,...,K)$ be a decision function of the kth SVM in the SVM ensemble that is trained by a replicate training data set $\mathrm{T}_k^B = \{(x_i'; y_i' \mid i = 1,2,..., L\}$. The weight vector w can be obtained by $w_E = A^{-1}y$, where $A = (f_i(x))_{K\times L}$ and $y = (y_j)_{1\times L}$. Next, the final decision of the SVM ensemble $f_{LSE}(x)$ for a given test vector x due to the LSE-based weighting is determined by $f_{LSE}\,(\mathrm{x}) = sign(w.[(f_i(x))_{K\times 1}])$.

Double-layer Hierarchical Combining

We use another SVM to aggregate the outputs of several SVMs in the SVM ensemble. This combination consists of a double-layer of SVMs hierarchically where the outputs of several SVMs in the lower layer feed into a super SVM in the upper layer. This type of combination looks similar of the mixture of experts introduced by Jordan and Jacobs (1994).

Let $f_k(k = 1,2,...,K)$ be a decision function of the kth SVM in the SVM ensemble and F be a decision function of the super SVM in the upper layer. Then, the final decision of the SVM ensemble $f_{SVM}(x)$ for a given test vector x due to the double-layer hierarchical combining is determined $f_{SVM} = F(f_1(x), f_2(x), ..., f_K(x))$, where K is the number of SVMs in the SVM ensemble.

Face Detection Using the SVM Ensemble

Figure 7 shows the schematic of our proposed face detection system. We implemented the system on a Pentium III-1GHz, 256MB memory with VC++6.0. All key frame images are generated from a Shot Change Detection (SCD) program designed by our research group. The images have the size of 320*240. For the SVM learning and classification, we also implemented the SVM module with SMO algorithm.

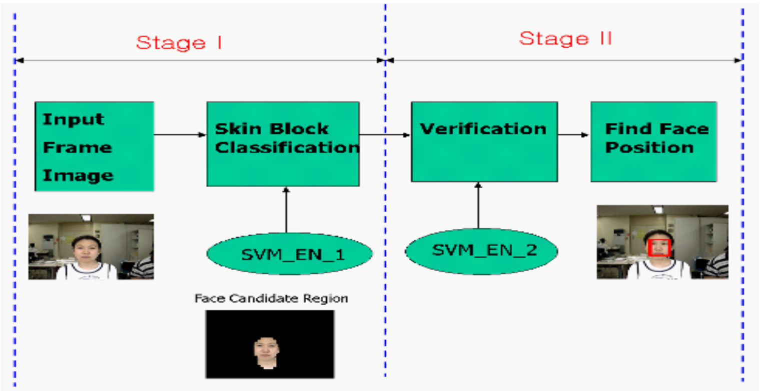
Stage I : Skin Block Classification

At this stage, skin block classification was performed based on chrominance and the texture information. Figure 8 shows a schematic diagram of Stage I. We take 8x8 sample patches for the skin and the non-skin block training. All skin blocks are classified by SVM_EN_1. As a result, a binary mask image is generated for each frame where the skin block indicates "1" and non-skin block indicates "0." After Stage I, the binary mask image is post-processed by the connected component analysis and the morphological operation (Castleman, 1996); then we can obtain the face candidate region, which is a lump of a skin block. In real implementation, SVM_EN_1 consists of five independently trained SVMs.

Stage II : Face or Not Classification

Stage II locates the true face location with sub-window matching over the face candidate region generated from Stage I.

Figure 7. Overview of face detection system



1. *Training step:* Each face is represented by 40x40 sub-window image and each face image is represented by a feature vector consisting of dominant 50 PCA components for each sub-window sample. Face examples are collected in a similar manner by Rowley et al.(1998). From each 50 people, arbitrary rotation or scaling has been performed to the original images. From it, we obtain 500 face examples. Non-face examples are generated by bootstrapping and randomly selected from face-like objects.
2. *Classification step:* SVM_EN_2 classifies whether the current sub-window is a face or not. To detect faces larger than sub-window size(40x40), we scale up and down the size of sub-window in the face candidates, but the selected sub-window is rescaled into the size of 40x40 and 50 PCA feature components are obtained from it.

We repeat the classification step until all sizes of the sub-window are considered in the face candidate region. In real implementation, SVM_EN_2 consists of five or 11 independently trained SVMs. Figure 9 shows the detail sequence of Stage II.

Figure 8. Stage I - Skin block classification

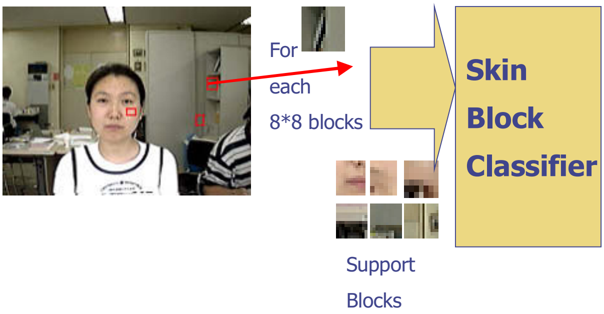
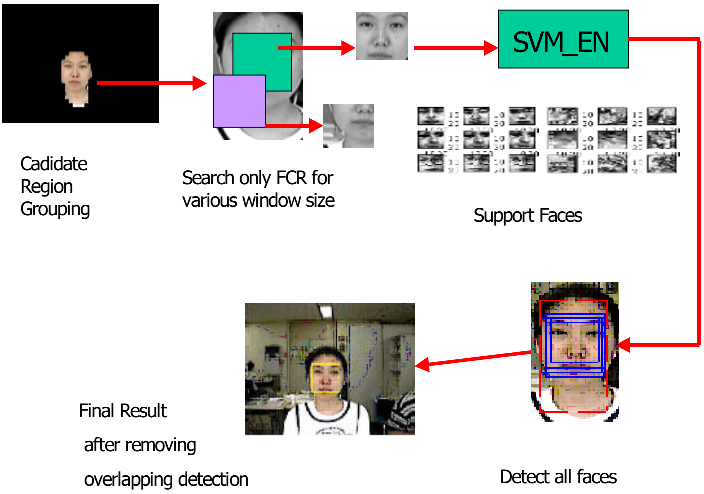


Figure 9. Stage II - Find true face location



In Figure 10 and Figure 11, part of the face and non-face examples for training are shown. Each example was rescaled into 20x20 size and applied histogram equalization.

Figure 10. Face examples

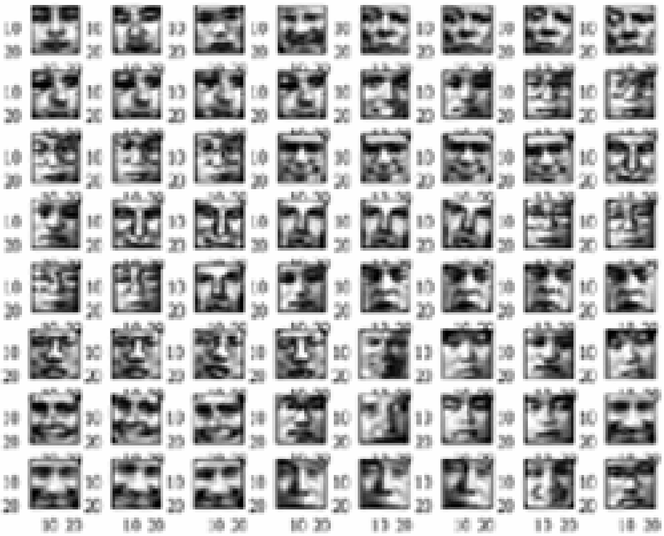


Figure 11. Non-face examples

The Face Recognizer

After the face detection step, the Face Recognizer hands over facial features of detected face from the Face Detector. In this paragraph, we explain theoretical background and algorithms of face recognition using eigenface.

Eigenface Method

The task of face recognition is discriminating input feature data into several classes of persons in video. The input data are usually highly noisy (e.g., the noise is caused by differing lighting conditions, pose, etc.), yet the input images are not completely random and, in spite of their differences, there are patterns that occur in any input signal. Such patterns, which can be observed in all signals, could be the presence of some objects (eyes, nose, mouth) in any face, as well as relative distances between these objects. These characteristic features are called eigenfaces in the face recognition domain. They can be extracted out of original image data by means of a mathematical tool called Principal Component Analysis (PCA). By means of PCA one can transform each original image of the training set into a corresponding eigenface. An important feature of PCA is that one can reconstruct any original image from the training set by combining the eigenfaces. Remember that eigenfaces are nothing less than characteristic features of the faces. Therefore one could say that the original face image can be reconstructed from eigenfaces if one adds up all the eigenfaces in the right proportion. Each eigenface represents only certain features of the face, which may or may not be present in the original image. If the feature is present in the original image to a higher degree, the share of the corresponding eigenface in the "sum" of the eigenfaces should be greater. If, on the contrary, the particular feature is not or almost not present in the original image, then the corresponding eigenface should contribute a smaller or not at all part to the sum of eigenfaces. So, in order to reconstruct the original image from the eigenfaces, one has to build a kind of weighted sum of all eigenfaces; that is, the reconstructed original image is equal to a sum of all eigenfaces, with each eigenface having a certain weight. This weight specifies to what degree the specific eigenface is present in the original image.

If one uses all the eigenfaces extracted from original images, one can reconstruct the original images from the eigenfaces exactly. But, one can also use only a part of the eigenfaces, and then the reconstructed image is an approximation of the original image. However, one can ensure that losses due to omitting some of the eigenfaces can be minimized. This happens by choosing only the most important eigenfaces.

Face Recognition Algorithm

The algorithm for face recognition using eigenfaces is as follows: the original images of the training set are transformed into a set of eigenfaces. Afterwards, the weights are calculated for each image of the training set and stored in the set *W*. Upon observing an unknown image *X*, the weights are calculated for that particular image and stored in the vector W_X. Afterwards, W_X is compared with the weights of images that one knows for certain are faces.

The original scheme for determination of the eigenfaces using PCA is presented follow as:

Step 1: Prepare the data

The faces constituting the training set (Γ_i) should be prepared for processing.

Step 2: Subtract the mean

The average matrix ψ has to be calculated, then subtracted from the original faces (Γ_i) and the result stored in the variable Φ_i

$$\psi = \frac{1}{M}\sum_{n-1}^{M}\Gamma_n, \Phi_i = \Gamma_i - \psi :$$

$$=$$

Step 3: Calculate the covariance matrix

In the next step the covariance matrix C is calculated according to

$$C = \frac{1}{M}\sum_{n-1}^{M}\Phi_n\Phi_n^T$$

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

In this step, the eigenfaces u_i and the corresponding eigenvalues λ_i should be calculated. The eigenvectors (eigenfaces) must be normalized so that they are unit vectors, that is, of length 1. The description of the exact algorithm for determination of eigenvectors and eigenvalues is omitted here, as it belongs to the standard arsenal of most math programming libraries.

Step 5: Select the principal components

From M eigenfaces u_i, only M' should be chosen, which have the highest eigenvalues. The higher the eigenvalue, the more characteristic features of a face does the particular eigenvector describe. Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of characteristic features of the faces. After M' eigenfaces u_i are determined, the ”training” phase of the algorithm is finished.

Classifying the Faces

The process of classification of a new (unknown) face Γ_{new} to one of the classes (known faces) proceeds in two steps. First, the new image is transformed into its eigenface components. The resulting weights form the weight vector Ω^T_{new}

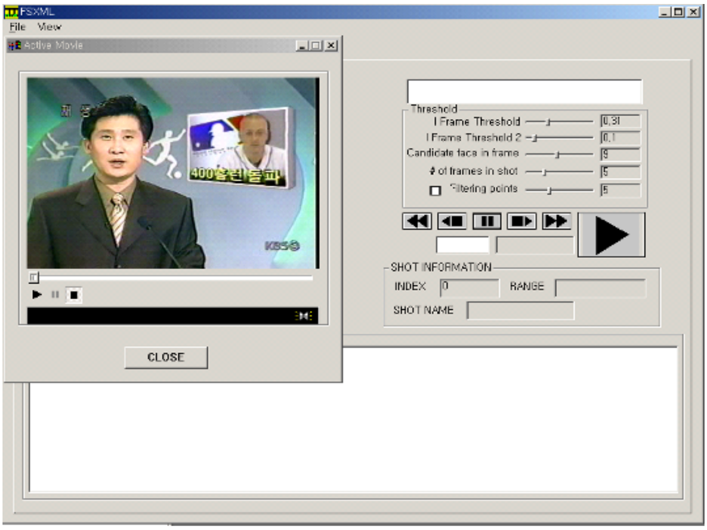
$$\omega_k = u_k^T(\Gamma_{new} - \psi), k = 1...M' \text{ , } \Omega^T_{new} = [\omega_1 \quad \omega_2 \quad ... \quad \omega_{M'} \quad]$$

The Euclidean distance between two weight vectors $d(\Omega_i, \Omega_j)$ provides a measure of similarity between the corresponding images i and j.

System Integration

Figure 12 indicates the User Interface (UI) of the proposed Video Summarization System embedding the Face Detector and Recognizer. The System consists of three stages. In Stage I, the Face Detector operates face detection onto frame image. If a human face exists there, the face detector returns the position of face and extracts facial features. In the next stage, the Face Recognizer performs a description with two modes: one is semi-auto mode, the other is full-auto mode. If the system works on new video data and FaceDB has no information about human faces or indices for the video, it is essential to manually describe human facial information by human-handed text annotation for a part of video

Figure 12a. UI of video summarization system



Threshold

I Frame Threshold 0,31

I Frame Threshold 2 0,1

Candidate face in frame 9

# of frames in shot 5

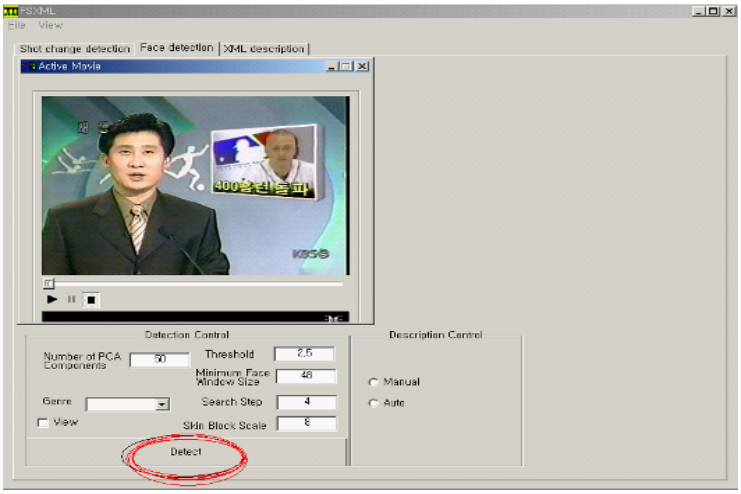
□ Filtering points 5

SHOT INFORMATION

INDEX 0 RANGE

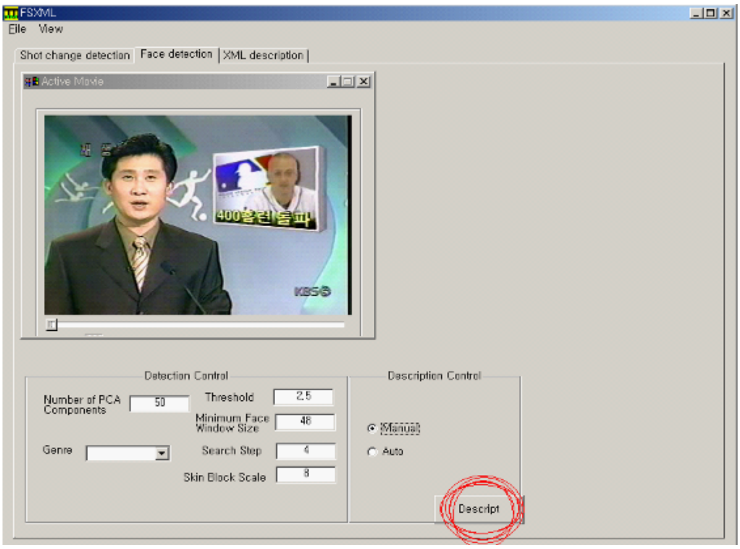
SHOT NAME

Figure 12b. Face detection



Number of PCA Components 50
Threshold 2.5
Minimum Face Window Size 48
Genre
Search Step 4
☐ View
Skin Block Scale 8

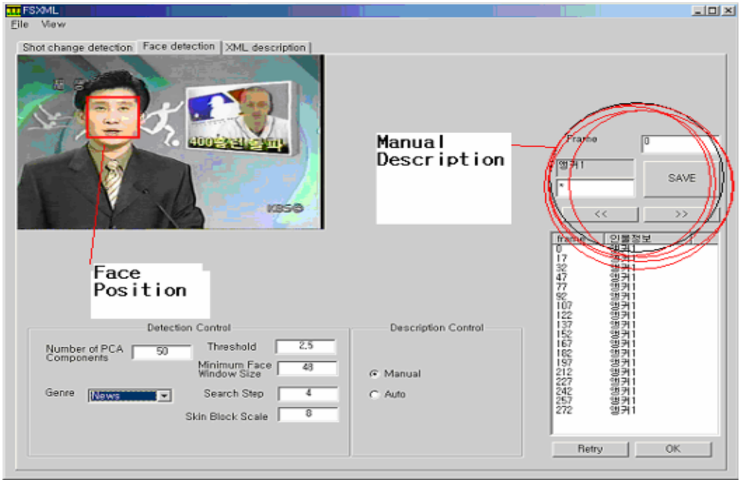
Figure12c. Semi-auto mode description



Number of PCA Components 50
Threshold 2.5
Minimum Face Window Size 48
Genre
Search Step 4
Skin Block Scale 8
● Manual
○ Auto

frames. And then, for the rest part, the Face Recognizer can index face information automatically. This mode is the semi-auto mode. Otherwise, the Face Recognizer is operating in full-auto mode. For all video frames for which the description process is done, Video Summarization System can support a verification of facial information since the Face Recognizer has a small degree of error, such as false alarm, false reject. In last

Figure 12d. Verification



Frame: 0

앵커1

Detection Control

Number of PCA Components: 50

Threshold: 2.5

Minimum Face Window Size: 48

Genre: News

Search Step: 4

Skin Block Scale: 8

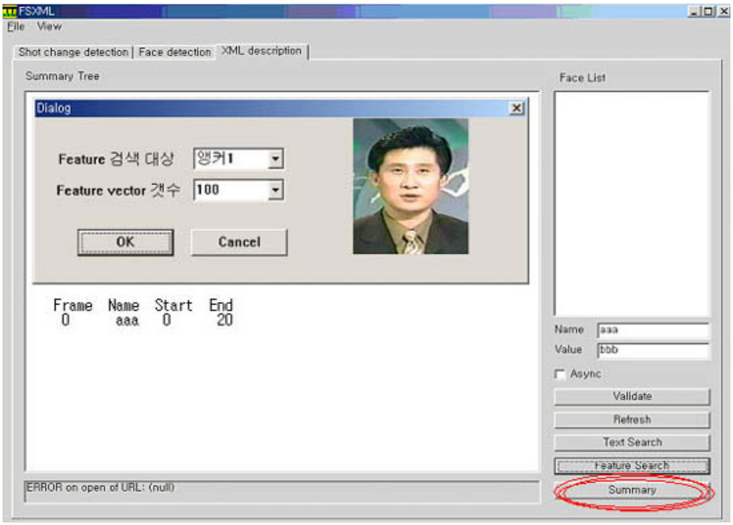
Description Control

● Manual

○ Auto

frame	인물정보
0	앵커1
17	앵커1
32	앵커1
47	앵커1
77	앵커1
92	앵커1
107	앵커1
122	앵커1
137	앵커1
152	앵커1
167	앵커1
182	앵커1
197	앵커1
212	앵커1
227	앵커1
242	앵커1
257	앵커1
272	앵커1

Figure 12e. Summarization



Feature 검색 대상: 앵커1

Feature vector 갯수: 100

Frame	Name	Start	End
0	aaa	0	20

Name: aaa

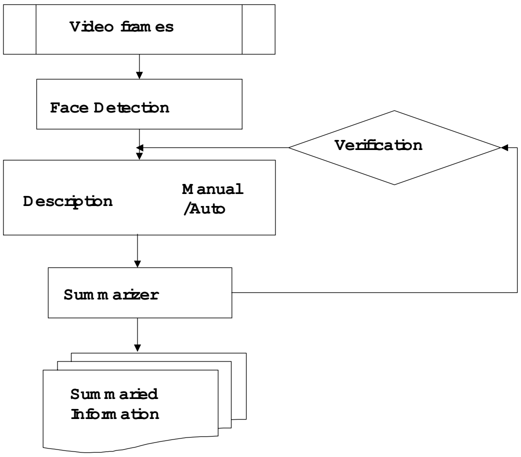
Value: bbb

□ Async

ERROR on open of URL: (null)

stage, the summarizer generates abstraction, browsing, or retrieval information of the video using the descriptor(index) for human faces. Figure 13 shows the block diagram of the processing flows. Figure 12(b-e) indicate the action or state of the Face Detector, Recognizer and Summarizer, respectively.

Figure 13. Block diagram of video summarization system



EXPERIMENTAL RESULTS

We will present our experimental results in this section. To estimate the accuracy of the face detection rate, we performed the simulation that is a face/non-face classification on the POSTECH_IM_DB (2001). In addition, results of the human-face-based summarization for the example video clips that are TV news and serial historical dramas also presented.

Face Detection Results

Skin Block Classification

We have performed the skin block classification using four different experiment conditions, such as SVM_P, SVM_G, SVM_EN(5 SVM_P), and a neural network, respectively. The training set consists of 100 examples of skin blocks and non-skin blocks, respectively. Two-hundred and fifty skin and non-skin blocks were tested for classification, respectively. Table 1 shows the result of the skin block classification.

Table 1. Result of skin block classification

	SVM_P	SVM_G	SVM_EN	NN
FALSE	18	11	5	7
MISS	15	7	2	12
Correct	93.4%	92%	98.6%	96.2%
#SV	16	52	20	N/A
Kernel	2D-Poly	RBF	2D-Poly	50 hidden

Table 2. Result of face/non-face classification

	SVM_S	EN_5	EN_11	MLP
FALSE	75	42	38	52
MISS	39	20	8	8
Correct	89.6%	93.8%	95.4%	94.0%
#SV	86	52	57	N/A

Here, "FALSE" refers to the false alarm that is classified into the skin even it is a non-skin block; "MISS" means that the skin block is misclassified into the non-skin block; and "#SV" indicates the average number of support vectors. Usually, SVM will have a greater computation as #SV increases. From the table, we note that SVM_EN shows the best performance among the four different experiments.

Face/Non-face Classification

We performed face/non-face classification on the POSTECH_IM_DB (2001) that includes the 1,000 faces and 1,000 non-faces under various light conditions for the training set. The test data is the remaining examples in the Face DB that are not included in the training set. Table 2 shows the results of the face/non-face classification experiment. EN_5 and EN_11 refer to the SVM ensembles that use five and 11 SVMs, respectively. SVM_S means a single SVM. Each SVM's kernel is used 2-D-Poly, because it has a better performance than the RBF kernel. As the number of SVMs increases, the classification performance improves. Especially, the performance of EN_11 is comparable with that of MLP's, the best result.

Face Recognition Results

We performed our experiment on ten new clips and two serious TV dramas. The train data(manual description) is one news clip, one episode per dramas. The results for the news clips are superior to dramas, because most news anchors are frontal-view and static, but the faces of actors/actresses in dramas are usually dynamic posed with various conditions. Table 3 presents the results of Face Recognition where a "FAR" is the False Alarm Rate, "FRR" is the False Reject Rate, "Frame" is the number of frames in which a human face exists, and "People" is the number of actors/actresses in the video.

Table 3. Result of face recognition

	#	**Frames**	**People**	**FAR(%)**	**FRR(%)**
News	9	3670	2	2	0.5
Drama #1	5	256788	6	17.2	8.6
Drama #2	10	134566	3	10.5	3.1

REMARKS AND FUTURE WORKS

In this chapter, we dealt with video summarization using human facial information by face detection and recognition. Also, we have described the real implementation of a video summarization system. In particular, we have presented an extensive survey of human face detection and recognition. So many efforts and ideas were introduced. Finding a face region and making out an identity from a still image or a video is one of the key first step in intelligent information processing. Both face detection and recognition on video data are promising for user-friendly video applications (for example, content-based indexing, intelligent video browsing). If any user wants answer to the following type of queries: "Are there any faces in this video sequences that matches Keanu Reeves?" or "In which movies in this data base does Brad Pitt appear? or "Find the advent scene in which Angelina Joli performed." The human face-based approach may help provide the answers automatically. However, face detection and recognition on video still have many challenging problems as, in general, the human face is exposed to very different illumination conditions, poses, and sizes. In addition, direct face recognition from a video sequences is a very meaningful challenge. Up to now, fairly simple thresholding of difference images has been used for locating a moving person's face, and has been followed by 2-D recognition algorithms. In our opinion, recognition from video offers excellent opportunities for applying several concepts from the Image Understanding. In particular, the usefulness of flow fields for the segmentation of the face region, and the reconstruction and refinement of 3-D structure from 2-D images must be investigated.

REFERENCES

Acosta, E., Torres, L., & Delp, E. (2002). An automatic face detection and recognition system for video indexing applications. *International Conference on Acoustics, Speech & Signal Processing*, 3644-3647.

Amit, Y., Geman, D., & Jedynak, B. (1998). Efficient focusing and face detection. *Face Recognition: From Theory to Applications, 163*(1), 124-156.

Augusteijn, M.F., & Skujca, T.L. (1993). Identification of human faces through texture-based feature recognition and neural network technology. *Proceedings of IEEE Conference Neural Networks,* 392-398.

Bartlett, M.S., Lades, H.M., & Sejnowski, T.J. (1998). Independent component representations for face recognition. *Proceedings of SPIE*, *3299*(1), 528-539.

Belhumeur, P.N., Hespanha, J.P., & Kriegman, D.J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis & Machine Intelligence, 19*(7), 711-720.

Bishop, C.M. (1995). *Neural networks for pattern recognition*. Oxford, UK: Clarendon Press.

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining & Knowledge Discovery*, *2*(2), 121-167.

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Analysis & Machine Intelligence, 8*(6), 679-698.

Castleman, K.R. (1996). *Digital image processing*. Englewood Cliffs, NJ: Prentice Hall.

Chai, D., & Ngan, K.N. (1998). Locating facial region of a head-and-shoulders color image. *Proceedings of the Third International Conference Automatic Face & Gesture Recognition*, 124-129.

Chellappa, R., Wilson, C.L., & Sirohey, S. (1995). Human and machine recognition of faces: A survey. *PIEEE*, *83*(1), 705-740.

Chen, Q., Wu, H., & Yachida, M. (1995). Face detection by fuzzy matching, *Proceedings of the Fifth IEEE International Conference on Computer Vision*, 591-596.

Cootes, T.F., Edwards, G.J., & Taylor, C.J. (1998). Active appearance models. In *Proceedings of the ECCV*, *2*(1), 484-498.

Cootes, T.F., Edwards, G.J., & Taylor, C.J. (2001). Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *23*(6), 681-685.

Craw, I., Ellis, H., & Lishman, J. (1987). Automatic extraction of face features. *Pattern Recognition Letters*, *5*(1), 183-187.

Craw, I., Tock, D., & Bennett, A. (1992). Finding face features. *Proceedings of the. Second European Conference Computer Vision*, 92-96.

Crowley. J.L., & Bedrune, J.M. (1994). Integration and control of reactive visual processes. *Proceedings of Third European Conference Computer Vision*, *2*(1), 47-58.

Crowley, J.L., & Berard, F. (1997). Multi-modal tracking of faces for video communications. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 640-645.

Dai, Y., & Nakano, Y. (1995). Extraction for facial images from complex background using color information & SGLD matrices. *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition*, 238-242.

Dai, Y., & Nakano, Y. (n.d.). Face-texture model based on SGLD and its application in face detection in a color scene. *Pattern Recognition*, *29*(6), 1007-1017.

Darrell, T., Gordon, G., Harville, M., & Woodfill, J. (2000). Integrated person tracking using stereo, color, and pattern detection. *International J. Computer Vision*, *37*(2), 175-185.

Dietterich, T.G. (1998). Machine learning research: Four current directions. *The AI Magazine*, *18*(4), 97-136.

Edwards, G.J., Cootes, T.F., & Taylor, C.J. (1998). Face recognition using active appearance models, *Proceedings of the ECCV*, *2*(1), 581-695.

Giacinto, G., Roli, F., & Bruzzone, L. (2000). Combination of neural and statistical algorithms. *Pattern Recognition Letter*, *21*(1), 385-397.

Gong, S., McKenna, S.J., & Psarrou, A. (2000). *Dynamic vision: From images to face recognition*. London: Imperial College Press & World Scientific Publishing.

Graf, H.P., Chen, T., Petajan, E., & Cosatto, E. (1995). Locating faces and facial parts. *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition*, 41-46.

Graf, H.P., Cosatto, E., Gibbon, D., Kocheisen, M., & Petajan, E. (1996). Multimodal system for locating heads and faces. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 88-93.

Hansen, L., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(1), 993-1001.

Haris, M., & Ganapathy, V. (2000). Neural network ensemble for financial trend prediction. *IEEE*, 157-161.

Hyvrinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *Neural Computing Surveys*, *2*(1), 94-128.

Hyvrinen, A. (2000). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Networks*, *10*(3), 626-634.

Je, H.M., Kim, D., & Bang, S.Y. (2003). Human face detection in digital video using SVM ensemble. *Neural Processing Letter*, *17*(1), 239-252.

Jebara, T.S., & Pentland, A. (1997). Parameterized structure from motion for 3D adaptive feedback tracking of faces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 144-150.

Jebara, T.S., Russell, K., & Pentland, A. (1998). Mixtures of Eigenfeatures for real-time structure from texture. *Proceedings of the Sixth IEEE International Conference on Computer Vision*, 128-135.

Jordan, M., & Jacobs, R. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*(5), 181-214.

Juell, P., & Marsh, R. (1996). A hierarchical neural network for human face detection. *Pattern Recognition*, *29*(5), 781-787.

Kanade, T. (1973). *Picture processing by computer complex and recognition of human faces*. PhD thesis, Kyoto University.

Karhunen, K. (1946). Uber Lineare Methoden in der Wahrscheinlichkeitsrechnung, Annales Academiae Sciientiarum Fennicae. *Series AI: Mathematica-Physica*, *37*(1), 3-79.

Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: Active contour models. *Proceedings of First IEEE International Conference on Computer Vision*, 259-269.

Kim, D. & Kim, C.-H. (1997). Forecasting time series with genetic fuzzy predictor ensemble. *IEEE Transaction on Fuzzy Systems*, *5*(4), 523-535.

Kim, H.-C., Pang, S., Je, H.-M., Kim, D., & Bang, S.-Y. (2002a). Support vector machine ensemble with bagging. *Lecture Notes of Computer Science*, 2388, Pattern Recognition with Support Vector Machines, pp. 397-4. New York: Springer.

Kim, H.-C., Pang, S., Je, H.-M., Kim, D., & Bang, S.-Y. (2002b). Pattern classification using support vector machine ensemble. ICPR 2002.

Kim, S.H., Kim, N.K., Ahn, S.C., &. Kim, H.G. (1998). Object oriented face detection using range and color information. *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, 76-81.

Kirby, M., & Sirovich, L. (1990). Application of the Karhunen-Lo´eve procedure for the characterization of human faces. *IEEE Trans. Pattern Analysis & Machine Intelligence*, *12*(1), 103-108.

Kjeldsen, R., & Kender , J. (1996). Finding skin in color images. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 312-317.

Kotropoulos, C., & Pitas, I. (1997). Rule-based face detection in frontal views. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 4, 2537-2540.

Kwon, Y.H., & da Vitoria Lobo, N. (1994). Face detection using templates. *Proceedings of the International Conference on Pattern Recognition*, 764-767.

Lades, M., Vorbruggen, J., Buhmann, J., Lange, J., Malsburg, C., Wurtz, R., & Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans Computers*, *42*(3), 300-310.

Lanitis, A., Taylor, C. J., & Cootes, T. F. (1995). An automatic face identification system using flexible appearance models. *Image and vision computing, 13*(5), 393-401.

Leung, T., Burl, M., & Perona, P. (1995). Finding faces in cluttered scenes using labeled random graph matching. In *Proceedings of the Fifth International Conference on Computer Vision*, 637-644.

Leymarie. F., & Levine, M.D. (1993). Tracking deformable objects in the plan using an active contour model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *15*(6), 617-634.

Lorente, L., & Torres, L. (1999). Face recognition of video sequences in a MPEG-7 context using a global eigen approach. *International Conference on Image Processing*, Kobe, Japan.

McKenna, S., Gong, S., & Raja, Y. (1997). Face recognition in dynamic scenes. *British Machine Vision Conference*.

Miao, J., Yin, B., Wang, K., Shen, L., & Chen, X. (1999). A hierarchical multiscale and multiangle system for human face detection in a complex background using gravity-center template. *Pattern Recognition*, *32*(7), 1237-1248.

Miyake, Y., Saitoh, H., Yaguchi, S., & Tsukada, N. (1990). Facial pattern detection and color correction from television picture for newspaper printing. *J. Imaging Technology*, *16*(5), 165-169.

Moghaddam, B. (2002). Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence, 24*(6), 780-788.

MPEG Requirements Group (1998). MPEG-7: Context and objectives. Doc. ISO/MPEG N2460, October 1998 / Atlantic City, NJ, USA.

Oh, J. H., & Hua, K.A. (2000). An efficient technique for summarizing video using visual contents. *IEEE International Conference on Multimedia and Expo,* 1167-1170.

Oliver, N., Pentland, A., & Berard, F. (1997). LAFER: Lips and face real time tracker. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 123-129.

Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: An application to face detection. *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition,* 130-136.

Pentland, A., Moghaddam, B., & Starner, T. (1994). View-based and modular eigenspaces for face recognition. *Proceedings of CVPR*, 84-91.

POSTECH_IM_DB (2001). Retrieved from: *http://nova.postech.ac.kr*

Qian, R.J., Sezan, M.I., & Matthews, K.E. (1998). A robust real-time face tracking algorithm. *Proceedings of the IEEE International Conference on Image Processing*, 131-135.

Rowley, H., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 20*(1), 23-38.

Saber, E., & Tekalp, A.M. (1998). Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. *Pattern Recognition Letters, 17*(8), 669-680.

Satoh, S., Nakamura, Y., & Kanade, T. (1999). NameIt: Naming and detecting faces in news videos. *IEEE Multimedia*, *6*(1), 22-35.

Saxe, D., & Foulds, R. (1996). Toward robust skin identification in video images. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*,379-384.

Schapire, R.E. (1999). A brief introduction to boosting. *Proceedings of the Sixth International Joint Conference On AI*, 1-6.

Scholkopf, B., Smola, A., & Muller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*(5), 1299-1319.

Sinha, P. (1994). Object recognition via image invariants: A case study. *Investigative Ophthalmology and Visual Science*, *35*(4), 1735-1740.

Sinha, P. (1995). *Processing and recognizing 3-D forms*. Ph.D. Thesis, Massachusetts Institute of Technology.

Sirohey, S.A. (1993). Human face segmentation and identification. Technical Report CS-TR-3176, University of Maryland.

Sobottka, J., & Pitas, I. (1996). Segmentation and tracking of faces in color images. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition,* 236-241.

Sobottka, K., & Pitas, I. (1996). Face localization and feature extraction based on shape and color information. *Proceedings of the IEEE International Conference on Image Processing*, 483-486.

Starner, T., & Pentland, A. (1996). Real-time ASL recognition from video using HMM's. Technical Report 375, Media Lab, Massachusetts Institute of Technology.

Sumi, Y., & Ohta, Y. (1995). Detection of face orientation and facial components using distributed appearance modeling. In *Proceedings of the International Workshop on Automatic Face and Gesture Recognition,* 254-259.

Sun, Q.B., Huang, W.M., & Wu, J.K. (1998). Face detection based on color and local symmetry information. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition,* 130-135.

Sung, K., & Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *20*(1), 39-51.

Swets, D.L., & Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *18*(8), 831-836.

Takacs, B., & Wechsler, H. (1995). Face location using a dynamic model of retinal feature extraction. *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition*, 243-247.

Tsukamoto, A., Lee, C.W., & Tsuji, S. (1993). Detection and tracking of human face with synthesized templates. *Proceedings of the First Asian Conference on Computer Vision*, 183-186.

Tsukamoto, A., Lee, C.W., & Tsuji, S. (1994). Detection and pose estimation of human face with synthesized image models. *Proceedings of the International Conference on Pattern Recognition*, 754-757.

Turk, M., & Pentland, A. (1991a). Face recognition using eigenfaces. *Proceedings of the IEEE Computer Society Conference on CVPR*, 586-591.

Turk, M., & Pentland, A. (1991b). Eigenfaces for recognition, *Journal of Cognitive Neuroscience*, *3*(1), 71-86.

Wang, H., & Chang, S.F. (1997). A highly efficient system for automatic face region detection in MPEG video. *IEEE Trans. Circuits and Systems for Video Technology*, *7*(4), 615-628.

Wechsler, H., Phillips, P., Bruce, V., Soulie, F., & Huang, T. (1996). *Face recognition: From theory to applications*. New York: Springer-Verlag.

Wiskott, L., Fellous, J.M., Krüger, N., & von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *19*(7), 775-779.

Wu, H., Chen, Q., & Yachida, M. (1999). Face detection from color images using a fuzzy pattern matching method. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *21*(6), 557-563.

Wu, H., Yokoyama, T., Pramadihanto, D., & Yachida, M. (1996). Face and facial feature extraction from color image. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 345-350.

Xiaoguang, L. (2003). Image analysis for face recognition – A summary note. Retrieved online at: *http://www.cse.msu.edu/~lvxiaogu/publications/ImAna4FacRcg_Lu.pdf*

Yang, G. & Huang, T.S. (1994). Human face detection in complex background. *Pattern Recognition*, *27*(1), 53-63.

Yang, J., Stiefelhagen, R., Meier, U., & Waibel, A. (1998). Visual tracking for multimodal human computer interaction. *Proceedings of ACM Human Factors in Computing Systems Conference (CHI 98)*, 140-147.

Yang, J., & Waibel, A. (1996). A real-time face tracker. *Proceedings of the Third Workshop on Applications of Computer Vision*, 142-147.

Yang, M.H. (2002). Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, DC, 215-220.

Yang, M.H., & Ahuja, N. (1998). Detecting human faces in color images. *Proceedings of the IEEE International Conference on Image Processing*, *1*(1), 127-130.

Yang, M.-H., Kriegman, D. J. & Ahuja, N. (2002). Detecting Faces in Images. *IEEE Trans. on PAMI*, *24*(1), 34-58.

Yow, K.C., & Cipolla, R. (1996). A probabilistic framework for perceptual grouping of features for human face detection. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 16-21.

Yow, K.C., & Cipolla, R. (1996). Feature-based human face detection. CUED/F-INFENG/TR 249, Cambridge University.

Yow, K.C. & Cipolla, R. (1997). Feature-based human face detection, image and vision. *Computing*, *15*(9), 713-735.

Yuillee, A.L., Halljnan, P.W., & Cohen, D.S. (1992). Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, *8*(1), 299-311.

Zhao,W., Chellappa, R., Rosenfeld, A., & Phillips, P.J. (2000). Face recognition: A literature survey. CVL Technical Report, University of Maryland. Retrieved online at: *ftp://ftp.cfar.umd.edu/TRs/CVLReports-2000/TR4167-zhao.ps.gz*

About the Authors

Sagarmay Deb is an international consultant and researcher in IT. He has been to many places in the world for consulting and research. He did his post-graduate studies in the US. His research interests are multimedia databases, content-based image retrieval, various indexing techniques and electronic commerce. He contributes to books on multimedia databases. He attends international conferences and also writes research papers for international journals. In addition, he is a reviewer of contributions to international conferences and journals. Currently, he is with the University of Southern Queensland, Australia, and is involved in research work. Marquis Who's Who, a well-known publisher of biographies of people of notable achievements, has included Deb's biography in its Seventh Edition of *Who's Who in Science and Engineering* in 2003. Also, his biography has been selected for inclusion in the *2000 Outstanding Scientists of the 21st Century Issue*, to be published by International Biographical Centre, Cambridge, UK, in 2004, in recognition of his achievements in the field of scientific research.

* * *

Hussein Abdel-Wahab received a PhD (1976) and an MS (1973) from the University of Waterloo in computer communications, and a BS in electrical engineering from Cairo University (1969). Currently, he is a full professor of computer science at Old Dominion University. In addition, he is an adjunct professor of computer science at the University of North Carolina at Chapel Hill and a faculty member at the Information Technology Lab of the National Institute of Standards and Technology. He previously held faculty positions at North Carolina State University, the University of Maryland, and the Rochester Institute of Technology. He served as a consultant to many organizations including IBM, MCNC, and MITRE Corp. He is the principal investigator in the design and implementation of XTV, a pioneer X-window-based teleconferencing system. His main research interests are collaborative desktop multimedia conferencing systems, and real-time distributed information sharing. His research has been supported by NSF, ONR, IBM, MCNC, MITRE, ARPA among others. He is a senior member of IEEE Computer Society and a member of the Association for Computing Machinery.

Sung-Yang Bang received a BS in electronic engineering from Kyoto University, Japan (1966), an MS in electrical and electronic engineering from Seoul University, Seoul, Korea (1969), and a PhD in computer science from Texas at Austin, USA (1974). From 1974-1975, he worked as an assistant professor at Wayne State University, USA. From 1979-1981, he was member of the technical staff in Bell Laboratory. From 1981-1984, he was a section head and division director at the Korea Institute of Electronic Technology, Seoul, Korea. From 1984-1986, he was director of the R&D Center in Union System Ltd., Seoul, Korea. He is currently professor of computer science and engineering at POSTECH. His research interests include neuro computing, pattern recognition, and human visual information processing.

Shu-Ching Chen received his PhD from the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana, USA (December 1998). He also received master's degrees in computer science, electrical engineering, and civil engineering from Purdue University, West Lafayette, IN. He has been an assistant professor in the School of Computer Science (SCS) at Florida International University (FIU) since August 1999. He is currently the director of Distributed Multimedia Information System Laboratory and the associate director of the Center for Advanced Distributed System Engineering (CADSE). Dr. Chen's research interests include distributed multimedia database systems and information systems, data mining, databases, and multimedia communications and networking. He received an outstanding faculty research award from SCS at FIU in 2002.

Dwipal A. Desai received a BS in computer engineering from Pune University, India. He is currently working toward his master's degree in computer science at USC. He is a research assistant pursuing research on high-speed data stream recording and playback at Integrated Media System Center (IMSC) and Data Management Research Laboratory at the Computer Science Department at USC. His current projects include high-definition video streaming and distributed immersive performance system. To contact: dwipalde@usc.edu

Laurence S. Dooley received his BSc (Hons), MSc, and PhD in electrical and electronic engineering from the University of Wales, Swansea (1981, 1983 and 1987, respectively). Since June 1999, he has been professor of multimedia technology at Gippsland School of Computing & IT, Monash University, Australia, where his major research interests are in multimedia signal processing, fuzzy image segmentation, mobile multimedia communications and technology transfer strategies for small regional business. He has published around 100 international scientific peer-reviewed journal articles, book chapters, and conference papers, and successfully supervised 11 PhD/master's research students. He is also currently the executive director of the Monash Regional Centre for Information Communications Technology. Professor Dooley is a senior member of the IEEE, a chartered engineer (CEng), and a corporate member of the British Computer Society (MBCS).

Waleed E. Farag received a BS in electronics and communications engineering from Zagazig University (Egypt) (1993) with distinguished honors. He then obtained his master's degree in computer engineering from the same university (1997). In 1998, he joined the PhD program in the Department of Computer Science, Old Dominion University (USA), where he got his PhD in 2002. Dr. Farag joined the Department of Computer Science at Indiana University of Pennsylvania (USA) as a faculty member in Fall 2002. His research interests include information hiding and network security, video and image indexing and retrieval techniques, distance learning applications, artificial neural networks, optimization techniques, multimedia applications, wireless and mobile computing, and parallel computer architecture. Dr. Farag has written numerous publications in his areas of interest. He is a reviewer for a number of international conferences and is an IEEE member.

Kun Fu received a BS in computer engineering from Beijing University of Posts and Telecommunications (BUPT) and an MS in engineering science from the University of Toledo. He is currently working towards a PhD in computer science at USC. He did research at the Data Communication Technology Research Institute and National Data Communication Engineering Center in China prior to coming to the US and is currently a research assistant working on high-speed large scale distributed continuous media systems at the Integrated Media System Center (IMSC) and Data Management Research Laboratory (DMRL) at the Computer Science Department at USC. He is a student member of the IEEE and the IEEE Computer Society. To contact: kunfu@cs.usc.edu

Phalguni Gupta received a doctoral degree from Indian Institute of Technology Kharagpur, India (1986). He works in the field of data structures, sequential algorithms, parallel algorithms, and online algorithms. From 1983-1987, he was in the Image Processing and Data Product Group of the Space Applications Centre (ISRO), Ahmedabad, India, and was responsible for software for correcting image data received from Indian Remote Sensing Satellite. In 1987, he joined the Department of Computer Science and Engineering at the Indian Institute of Technology, Kanpur, India. Currently, he is a professor in the department. He is responsible for several research projects in the area of biometric systems, mobile computing, image processing, graph theory and network flow. Dr. Gupta is a member of the Association Computing Machinery (ACM).

Hun-Hui Hsu is an associate professor in the Department of French Language and Literature at Tamkang University since 1997. Dr. Hsu received his BS in the French language and literature at Tamkang, Taiwan. He received an MS and a PhD from Paris VII University, French (1990 and 1996, respectively).

Jenq-Neng Hwang received his PhD from the University of Southern California (USA). He is currently a professor in the Department of Electrical Engineering, University of Washington, Seattle. He has published more than 180 journal, conference papers, and book chapters in the areas of image/video signal processing, computational neural networks, multimedia system integration and networking. Dr. Hwang is the co-author of an edited book, *Handbook of Neural Networks for Signal Processing* (CRC Press, July 2001). Dr. Hwang received the 1995 IEEE Signal Processing Society's Annual Best Paper Award. Dr. Hwang is a fellow of IEEE, and a founding member of Multimedia Signal Processing Technical Committee of IEEE Signal Processing Society. He served as chairman of the Neural Networks Signal Processing Technical Committee in IEEE Signal Processing Society from 1996-1998, and was the Society's representative to IEEE Neural Network Council from 1997-2000. He was program co-chair of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Seattle 1998.

Sae Hwang received an MS in computer science from the Texas A&M-CC University (2002). Currently, he is pursuing a PhD in computer science at the University of Texas at Arlington (USA). He is also working for Multimedia Information Group at the University of Texas at Arlington. His research topics include multimedia database management systems, medical imaging, multimedia communications, and data mining.

Hong-Mo Je received a BS in computer science from Yeungnam University, Taegu, Korea (1999) and an MS in computer science and engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea (2002). From 2002-2003, he worked at the Information Research Center in Samsung Data Systems (SDS) in Seoul, Korea. He is currently a PhD candidate student of computer science and engineering at POSTECH. His research interests include intelligent multimedia processing and biometrics.

Fan Jiang received a BS in electronic engineering from Tsinghua University, Beijing, China (2002). He is currently working toward his MS at the Research Institute of Image and Graphics in the Electronic Engineering Department, Tsinghua University. His research interests include content-based video analysis and pattern recognition.

Tatsuyuki Kawamura is a PhD student at the Artificial Intelligence Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology, Japan. He has been heavily involved in the development of residual memory and ubiquitous memories, and partially works on I'm Here! His research interests are in wearable computing, ubiquitous computing, human computer interaction, interpersonal communication, augmented memory, context awareness, and human interface. He has an ME in information science from the Nara Institute of Science and Technology in Japan. He received the best presentation award and the JSAI award from the Japanese Society for Artificial Intelligence (JSAI). He is a member of the IEEE. To contact: tatsu-k@is.aist-nara.ac.jp

Masatsugu Kidode is a professor at the Artificial Intelligence Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology, Japan. After he spent nearly 30 years at Toshiba Corp. in image understanding research and their industrial application developments, he has conducted real-world oriented AI techniques at his laboratory. He is mainly interested in multimedia information understanding and their applications, such as computer vision, human interface, artificial intelligence, robotics, and so on. To contact: kidode@is.aist-nara.ac.jp

Daijin Kim earned a BS in electronic engineering from Yonsei University, Seoul, Korea (1981), an MS in electrical and electronic engineering from KAIST, Taejon, Korea (1984), and a PhD in electrical and computer engineering from Syracuse University, USA (1991). From 1984-1986, he worked at HDTV Team, Korea Broadcasting System, Research Engineer, Seoul, Korea. From 1988-1991, he was employed by Environmental Science and Forest in SUNY, New York, as a research assistant and part-time research engineer. From 1992-1999, he was an associate professor at the Department of Computer Engineering at DongA University, Pusan, Korea. Since 1999, he has been an associate professor at the Department of Computer Science & Engineering in POSTECH, Pohang, Korea. His research interests include soft computing, content-based video processing, biometrics, and human computer interface.

Yasuyuki Kono is an associate professor at the Artificial Intelligence Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology, Japan. He received his PhD from the Graduate School of Engineering Science, Osaka University (1994). He conducted research on multimodal interfaces and on spoken dialog systems at Research & Development Center, Toshiba Corporation. He is also interested in multimodal understanding, spoken dialog systems, intelligent interfaces, and wearable interfaces. To contact: kono@is.aist-nara.ac.jp

Jeongkyu Lee received an MS in computer science from Sogang University in South Korea in 2001. He has been a doctoral student in the Department of Computer Science and Engineering at the University of Texas, Arlington, since 2002. He worked for IBM from 2000-2001, where he was a database administrator. His current research interests include multimedia database management systems, ontology for multimedia data indexing, and database management under Internet environments.

Yi-Chun Liao received a BS and MS from the Department of CSIE, Tamkang University, Taipei, Taiwan (1999 and 2001). He is currently a PhD candidate in the Department of CSIE at Tamkang University. His current research interests include multimedia computing, communication protocol, and video processing.

Yi-Jen Liu received bachelor and master's degrees in the Department of CSIE from Tamkang University, Taipei, Taiwan (2000 and 2002). He is currently a PhD candidate in the Department of CSIE at Tamkang University. His current research interests include multimedia computing, communication protocol, mobile computing, and content-based retrieval.

Michael Rung-Tsong Lyu received a BS (1981) in electrical engineering from the National Taiwan University, an MS (1985) in computer engineering from the University of California, Santa Barbara, and a PhD (1988) in computer science from the University of California, Los Angeles. He is a professor in the Computer Science and Engineering Department of the Chinese University of Hong Kong. He worked at the Jet Propulsion Laboratory, Bellcore, and Bell Labs, and taught at the University of Iowa. His research interests include software reliability engineering, software fault tolerance, distributed systems, image and video processing, web technologies, multimedia systems, and wireless communications. He has published more than 150 papers in these areas. He initiated the International Symposium on Software Reliability Engineering (ISSRE) and was program chair for ISSRE1996, Program co-chair for WWW10, and general chair for ISSRE2001. He also received Best Paper Awards in ISSRE1998 and in ISSRE2002. He is the editor for two book volumes: *Software Fault Tolerance* (Wiley, 1995) and the *Handbook of Software Reliability Engineering* (IEEE and McGraw-Hill, 1996). He has been an associated editor of *IEEE Transactions on Reliability*, *IEEE Transactions on Knowledge and Data Engineering*, and the *Journal of Information Science and Engineering*. Dr. Lyu is a fellow of IEEE.

Arun K. Majumdar is currently the dean (Faculty and Planning) and professor of the Computer Science and Engineering (CSE) Department at the Indian Institute of Technology, Kharagpur, India. Earlier, he served as head of the CSE Department (1992-1995 and again from 1998-2001). Dr. Majumdar received an MTech and PhD from the University of Calcutta (1968 and 1973, respectively). He also earned a PhD in electrical engineering from the University of Florida, Gainesville, Florida (1976). Professor Majumdar served as a faculty member at the Indian Statistical Institute, Calcutta, and the Jawaharlal Nehru University, New Delhi. He was a visiting professor in the Computing and Information Sciences Department of the University of Guelph, Canada (1986-1987). Professor Majumdar has more than 140 research publications in international journals and conferences. He has supervised several PhD and master's level theses. He is a fellow of the Institute of Engineers (India), fellow of the Indian National Academy of Engineering, and a senior member of the IEEE (USA). His research interests include data and knowledge based systems, medical information systems, design automation, and image processing.

Muthyala Mohan is currently pursuing his MTech in computer and information technology at the Indian Institute of Technology, Kharagpur, India. He did his BTech in computer science and engineering from S.K. University, Andhra Pradesh, India. His research interests are in the areas of video database and content-based retrieval.

Manzur Murshed received a BScEngg (Hons.) in computer science and engineering from Bangladesh University of Engineering and Technology (BUET) (1994) and PhD in computer science from the Australian National University (1999). He is currently the director of research and a senior lecturer at Gippsland School of Computing and Information Technology and the joint director of Multimedia Research Group at Faculty of IT, Monash University, Australia. His major research interests include multimedia signal processing and communications, parallel and distributed computing, algorithms, and multilingual systems development. He has published more than 40 refereed international journal and conference publications. Dr. Murshed is a member of the IEEE.

Jung Hwan Oh received an MS and PhD in computer science from the University of Central Florida (1997 and 2000, respectively). During his study, he also worked for Office of Research at the University of Central Florida, where he led a project to implement a relational database handling proposals, awards of grants, and faculty information. As soon as he finished his PhD, he worked as a visiting professor in the School of Electrical Engineering and Computer Science at the University of Central Florida. Dr. Oh joined the

Department of Computer Science and Engineering at the University of Texas, Arlington (2001). He is a member of IEEE, Association of Computing Machinery (ACM), and International Association of Science and Technology for Development (IASTED) Technical Committee on "Database." He is the author or coauthor of many journal articles, book chapters, and conference papers, and leads the Multimedia Information Group in the University of Texas, Arlington. His research topics include video database management systems, image database management systems, medical video and image processing, and video communications in wired and wireless environments.

Manoranjan Paul received a BSc (Hons.) in computer science and engineering from Bangladesh University of Engineering and Technology (BUET) (1997). He joined as a lecturer in the Computer Science and Engineering Department, Ahsanullah University of Science and Technology, Bangladesh (1997) and was promoted to an assistant professor in 2000. He was admitted as a PhD student in Gippsland School of Computing and IT (GSCIT) of Monash University, Australia in 2001, and also appointed as an assistant lecturer (part time) in GSCIT since his PhD candidacy. His major research interests are in the fields of image/video coding, multimedia communication, video indexing, video-on-demand, image segmentation, and artificial intelligence. He has published more than 10 refereed international journal and conference publications. Mr. Paul is a student member of the IEEE.

Timothy K. Shih is a professor in the Department of Computer Science and Information Engineering at Tamkang University, Taiwan, ROC. His research interests include multimedia computing and networking, software engineering, and formal specification and verification. He was a faculty of the Computer Engineering Department at Tamkang University in 1986. From 1993-1994, he was a part time faculty of the Computer Engineering Department at Santa Clara University. He was also a visiting professor at the University of Aizu, Japan in the summer of 1999. Dr. Shih received his BS and MS in computer engineering from Tamkang University and California State University, Chico (1983 and 1985, respectively). He also received his PhD in computer engineering from Santa Clara University in 1993. Dr. Shih has published more than 200 papers and participated in many international academic activities. Dr. Shih has received many research awards, including Tamkang University research awards, NSC research awards (National Science Council of Taiwan), and IIAS research award of Germany. He also has received many funded research grants from NSC, from the Institute of Information Industry, Taiwan, and from the University of Aizu, Japan. Dr. Shih has been invited frequently to give tutorials and talks at domestic and international conferences and research organizations. For more information, visit: http://www.mine.tku.edu.tw/chinese/teacher/tshih.htm. To contact: tshih@cs.tku.edu.tw

Mei-Ling Shyu has been an assistant professor at the Department of Electrical and Computer Engineering, University of Miami since January 2000. She received her PhD from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA, in 1999. She also received her MS in computer science, MS in electrical engineering, and MS in restaurant, hotel, institutional, and tourism management from Purdue University, West Lafayette, IN, USA (1992, 1995, and 1997, respectively). Her research interests include multimedia networking, wireless communications and networking, data mining, multimedia database systems, multimedia information systems, and database systems.

Richa Singh received a BTech (CSE) from the Institute of Engineering & Technology Jaunpur, India (2002). She is presently working at the Indian Institute of Technology, Kanpur, to design and to develop a multimodal biometric system that includes face, fingerprint, and iris recognition and has been sponsored by the Ministry of Communication and Information Technology, Government of India, India. Her current areas of interest are pattern recognition, image and video processing, biometric authentication, and neural network.

Jiqiang Song received his BS in computer science and PhD in computer science and application from Nanjing University (1996 and 2001, respectively). He is currently a postdoctoral fellow at Department of Computer Science and Engineering, the Chinese University of Hong Kong. His research interests include videoconference indexing, video coding, graphics recognition, automatic interpretation of engineering drawings, image processing, and video processing. He won the first place in the GREC 2003 Arc Segmentation Contest. He has published more than 20 papers in these areas. He is a member of IEEE, IEEE Computer Society and IAPR.

Shamik Sural completed his PhD in 2000. He is currently an assistant professor in the School of IT at the Indian Institute of Technology, Kharagpur, India. Dr. Sural has worked for more than 10 years in IT consulting projects in both India and the US. During this period, he managed and executed large application software projects involving n-tier client server architecture, rational unified process, and SEI CMM Level 5 certified software quality management techniques. Dr. Sural's research interests are in the areas of image processing, pattern recognition, and multimedia databases. He is a member of the IEEE and the IEEE Computer Society.

Takahiro Ueoka is a PhD student at the Artificial Intelligence Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology, Japan. He has proposed the concept of *I'm Here!* and is also devoted to developing its wearable interface. His research interests address in human computer interaction, wearable computing, augmented memory, human interface, and object recognition. He received an ME in information science from the Nara Institute of Science and Technology. He is a member of the IEEE. To contact: taka-ue@is.aist-nara.ac.jp

Mayank Vatsa received a BTech (CSE) from the Institute of Engineering & Technology Jaunpur, India (2002). At present, Mr. Vatsa is actively involved in the development of a multimodal biometric system that includes face, fingerprint, and iris recognition at the Indian Institute of Technology, Kanpur, India. The work has been sponsored by the Ministry of Communication and Information Technology, Government of India, India. His current areas of interest are pattern recognition, image and video processing, biometric authentication, and artificial intelligence.

Quan Wen received a BS in electrical engineering and an MS in computer science from Sichuan University, China (1994 and 1997, respectively). From 1997-2002, he worked at China Telecom, with emphasis on video-on-demand techniques. Now, he is a PhD student in the Department of Computer Science and Engineering at the University of Texas, Arlington, and also a member of the Multimedia Information Group. His research interests relate to various aspects of video, such as video processing, video database, and video delivery.

Chengcui Zhang is a PhD candidate at the school of Computer Science in Florida International University. She received her BS and MS in Computer Science from Zhejiang University in China. Her research interests include multimedia data mining, video and image database retrieval, transportation surveillance video databases, and GIS data filtering. Over the last three years, she has received several awards, including the Presidential Fellowship and the Best Graduate Student Research Award.

Yanchun Zhang is currently an associate professor and director of the Internet Technologies and Applications Research Lab (ITArl), School of Computer Science and Mathematics, Victoria University, Melbourne City, Australia. He earned a PhD in computer science from the University of Queensland (1991). His research areas cover database and information systems, object-oriented and multimedia database systems, distributed and multidatabase systems, database support for cooperative work, electronic commerce, Internet/Web information systems, web data management, and visual database processing. He is a regular contributor to international journals and conferences. He is editor-in-chief of *World Wide Web* (*WWW Journal*) and *Internet and Web Information Systems* (*IWIS Journal*) from Kluwer Academic Publishers. He is a chairman of International Web Information Systems Engineering Society (WISE Society).

Yu-Jin Zhang received a PhD in applied science from Liège University, Belgium (1989). From 1989-1993, he was with the Delft University of Technology, The Netherlands, as a post-doctoral fellow and research scientist. In 1993, he joined Tsinghua University, Beijing, China, where he is a full professor of image engineering since 1997. He took a sabbatical leave as a visiting professor at National Technological University, Singapore, in 2003. His research interests are mainly in the area of image engineering, including image processing, image analysis, and image understanding, as well as their applications. He has authored more than 200 research papers and book chapters, and he is the author of eight technical books including two monographs: *Image Segmentation* and Content-*based Visual Information Retrieval*' (Science Press). He is the vice president of China Society of Image and Graphics, the deputy editor-in-chief of the *Journal of Image and Graphics*, and on the editorial board of several international and national journals. He has served as program co-chairs of The First and Second International Conferences on Image and Graphics (ICIG2000, ICIG2002). He is a senior member of IEEE.

Roger Zimmermann received his PhD from the University of Southern California, Los Angeles, where he is a research assistant professor with the Computer Science Department, the director of the Data Management Research Laboratory (DMRL), and also a research area director with the Integrated Media Systems Center (IMSC). His research interests are in the area of streaming media architectures and distributed database integration. He has published extensively at conferences and in journals and has also built prototypes, such as the Yima streaming media server and the Remote Media System (RMI). He is a member of ACM and IEEE and can be reached at rzimmerm@usc.edu.

Index