

國立交通大學

資訊工程學系

博士論文



高階視訊處理、擷取、特徵粹取及視訊
結構化計算之研究

Towards High-Level Content-Based Video
Retrieval and Video Structuring

研究生：陳敦裕

指導教授：李素瑛 教授

中華民國九十三年九月

高階視訊處理、擷取、特徵粹取及視訊結構化計算之研究

Towards High-Level Content-Based Video Retrieval and
Video Structuring

研究生：陳敦裕

Student : Duan-Yu Chen

指導教授：李素瑛 博士

Advisor : Dr. Suh-Yin Lee

國立交通大學

資訊工程學系

博士論文



Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao-Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Computer Science and Information Engineering
September 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年九月

高階視訊處理、擷取、特徵粹取及視訊結構化計算之研究

學生：陳敦裕

指導教授：李素瑛 教授

國立交通大學資訊工程學系

摘要

隨著數位視訊在教育、娛樂、以及其它多媒體應用的發展下，造成數位視訊資料大量且迅速增加。在此情況之下，對於使用者而言，需憑藉一個有效的工具來快速且有效率地獲得所要的視訊資料。在搜尋視訊資料的方法中，對於使用者而言以內容為基礎之方法最具有高階語意意義，也最為自然且友善。因此，以視訊內容為基礎之搜尋、瀏覽以及擷取吸引各領域的學者研發各種粹取視訊資料中的高階特徵，以提供有效率地搜尋並擷取資料。但另一方面，隨著視訊資料壓縮法的成熟，愈來愈多的視訊資料以壓縮型態儲存，特別是 MPEG 格式。因此也吸引了愈來愈多的學者投入在壓縮的視訊資料中粹取其高階特徵之研究。本論文主旨在於研發粹取精簡且有效之視訊特徵，並達成具有語意之高階視訊資料結構化。

首先，我們在壓縮視訊資料中偵測移動物體，並提出移動物體追蹤演算法，以追蹤物體並產生物體軌跡，憑藉著物體軌跡，推測相對應之事件並產生事件之標籤，最終建立以事件為基礎之視訊資料結構化瀏覽系統。

在建立高階視訊資料結構化當中，除了視覺資料之外，文字資料亦是更具有語意意義的特徵，因此我們也提出了在壓縮視訊資料當中偵測文字字幕，並利用字幕的長時間出現特性作為濾除雜訊之基礎以及文字字幕其梯度能量較高之特性，以此獲得有意義的文字字幕，提供具語意之視訊結構化之計算。

為了提供有效的視訊資料相似性的比對，以利視訊資料擷取，我們也提出了

兩個以移動物體為基礎之高階特徵(T2D-Histogram Descriptor 以及 Temporal MIMB Moments Descriptor)。與傳統方法在粹取視訊資料特徵僅考慮空間特性不同，我們所提出的兩個 descriptor 利用了視訊資料之空間以及時間的特性。我們以 Discrete Cosine Transform 之能量集中之特性，將各個影格之空間特性作為連結，並大幅降低特徵值之資料量，達到高階視訊特徵精簡化但視訊資料相似性比對高效率的目的。

我們進行了大規模完整的實驗以評估所提各方法的效能。在我們的實驗範圍中，結果顯示，對於眾多的測試視訊資料，我們的視訊資料相似性比對的方法都優於許多著名的方法。



Towards High-Level Content-Based Video Retrieval and Video Structuring

Student: Duan-Yu Chen

Advisor: Prof. Suh-Yin Lee

Department of Computer Science and Information Engineering
National Chiao-Tung University

Abstract

With the increasing digital videos in education, entertainment and other multimedia applications, there is an urgent demand for tools that allow an efficient way for users to acquire desired video data. Content-based searching, browsing and retrieval is more natural, friendly and semantically meaningful to users. With the technique of video compression getting mature, lots of videos are being stored in compressed form and accordingly more and more researches focus on the feature extractions in compressed videos especially in MPEG format. This thesis aims to investigate high-level semantic video features in compressed domain for efficient video retrieval and video browsing.

We propose an approach for video abstraction to generate semantically meaningful video clips and associated metadata. Based on the concept of long-term consistency of spatial-temporal relationship between objects in consecutive P-frames, the algorithm of multi-object tracking is designed to locate the objects and to generate the trajectory of each object without size constraint. Utilizing the object trajectory coupled with domain knowledge, the event inference module detects and identifies the events in the application of tennis sports. Consequently, the event information and metadata of associated video clips are extracted and the abstraction of video streams is

accomplished.

A novel mechanism is proposed to automatically parse sports videos in compressed domain and then to construct a concise table of video content employing the superimposed closed captions and the semantic classes of video shots. The efficient approach of closed caption localization is proposed to first detect caption frames in meaningful shots. Then caption frames instead of every frame are selected as targets for detecting closed captions based on long-term consistency without size constraint. Besides, in order to support discriminate captions of interest automatically, a novel tool – font size detector is proposed to recognize the font size of closed captions using compressed data in MPEG videos.

For effective video retrieval, we propose a high-level motion activity descriptor, object-based transformed 2D-histogram (T2D-Histogram), which exploits both spatial and temporal features to characterize video sequences in a semantics-based manner. The Discrete Cosine Transform (DCT) is applied to convert the object-based 2D-histogram sequences from the time domain to the frequency domain. Using this transform, the original high-dimensional time domain features used to represent successive frames are significantly reduced to a set of low-dimensional features in frequency domain. The energy concentration property of DCT allows us to use only a few DCT coefficients to effectively capture the variations of moving objects. Having the efficient scheme for video representation, one can perform video retrieval in an accurate and efficient way.

Furthermore, we propose a high-level compact motion-pattern descriptor, temporal motion intensity of moving blobs (MIMB) moments, which exploits both spatial invariants and temporal features to characterize video sequences. The energy concentration property of DCT allows us to use only a few DCT coefficients to precisely capture the variations of moving blobs. Compared to the motion activity

descriptors, RLD and SAH, of MPEG-7, the proposed descriptor yield 40% and 21 % average performance gains over RLD and SAH, respectively.

Comprehensive experiments have been conducted to assess the performance of the proposed methods. The empirical results show that these methods outperform state-of-the-art methods with respective various datasets of different characteristics.



誌謝

在博士班在學期間，沒有指導教授 李素瑛老師的耐心指導，絕對無法成就這本博士論文。老師嚴謹的治學態度，在研究與投稿的每個階段屢次再現，屢見老師斟酌字句，費心審視。老師的指引、提攜與照顧，學生銘感於心。

博士論文口試委員之一的 廖弘源老師，在我面臨關鍵時刻，適時給予指導與協助，特別是在論文的撰寫上，給予指導與鼓勵，讓學生深深受益，在此表示由衷感激。

感謝所有口試委員，不吝於提供多年的寶貴研究經驗，充實了本論文的深度與廣度。謝謝 廖弘源老師、陳稔老師、陳銘憲老師、陳良弼老師、楊熙年老師、與吳家麟老師為豐富本論文內涵提供絕佳的意見，在方法適用範圍、方法評比、研究結果的適切論述、方法的差異性等等見解，使本論文更臻完善。諸位口試委員都是我學術研究的最佳典範。

資訊系統實驗室的學長、及學弟妹們，特別是林明言學長、學弟蕭銘和、陳華總及陳漪紋等在研究的坦途上與我攜手並進，謝謝大家也祝福學弟、妹們早日收穫豐碩的研究成果。在此特別感謝學長林明言博士的熱心及研究心得的分享，增添許多研究的動力與樂趣。

家人給予我的關懷與支持讓我在博士求學過程中無後顧之憂，一路走來，父母從不給我壓力，總是給予我溫暖的關懷，姊姊與哥哥對於弟的關心，以及對於父母的照顧，更是給予我勇往直前的動力。沒有他們的犧牲與支持，絕對沒有今日的我，感恩家人的支持，也謝謝姊夫、大嫂及其他親友對我的祝福與勉勵。

一直陪伴在我身旁、沒有怨言、不給我壓力、只有給我鼓勵的，就是我的太太怡君。每當我身心俱疲的時候，她總是能夠化解我內心的焦慮，總是能夠給予我綿綿不絕的支持，雖然她與我屬不同研究領域，但亦總是能激發我對研究的創意與靈感。能夠順利完成博士學位，對於怡君，我有無盡的感謝。

要感謝的人真的很多，在此向所有曾經幫過我的人，致上我真切的謝意。

僅以此論文，獻給我摯愛的家人。

Contents

摘要.....	i
Abstract.....	iii
誌謝.....	vi
Contents.....	vii
List of Figures.....	xi
List of Tables.....	xv
Chapter 1. Introduction.....	1
Chapter 2. Automatic Content Parsing and Semantic Event Identification for Sports Video Abstraction and Description	
2.1 Introduction.....	4
2.2 Overview of The System Architecture.....	7
2.3 Video Segmentation and Shots Selection.....	9
2.3.1 GOP-Based Video Segmentation.....	9
2.3.2 Scene Identification.....	10
2.4 Camera Motion Compensation.....	12
2.4.1 Adaptive Threshold Decision.....	12
2.4.2 Camera Motion Estimation.....	13
2.5 Events Detection and Description.....	16
2.5.1 Object Tracking Algorithm.....	17
2.5.1.1 Object Localization.....	17
2.5.1.2 Object Tracking Forward and Backward.....	21
2.5.2 Events Inference Model.....	23
2.5.3 Event Description Scheme.....	28
2.6 Experimental Results and Discussion.....	29

2.7 Summary.....	35
Chapter 3. Automatic Closed Caption Detection and Filtering in MPEG Videos for Video Structuring.....	38
3.1 Introduction.....	38
3.2 Shot Identification.....	41
3.2.1 Video Segmentation.....	41
3.2.2 Shot Identification.....	41
3.3 Closed Caption Localization.....	44
3.3.1 Caption Frame Detection.....	45
3.3.2 Closed Caption Localization.....	48
3.3.3 Font Size Differentiation.....	52
3.4 Experimental Results and Visualization System.....	57
3.4.1 Experimental Results.....	57
3.4.2 The Prototype System of Video Content Visualization.....	63
3.5 Summary.....	66
Chapter 4. Motion Activity Based Shot Identification and Closed Caption Detection for Volleyball Video Structuring.....	67
4.1 Introduction.....	67
4.2 Video Segmentation.....	70
4.3 Shot Identification.....	71
4.3.1 Moving Object Detection.....	71
4.3.2 Motion Activity Descriptor – 2D Histogram.....	73
4.3.3 Shot Identification Algorithm.....	75
4.4 Closed Caption Localization.....	78
4.4.1 Localization of Superimposed Closed Captions.....	78
4.4.2 Clustering-Based Noise Filtering.....	81

4.5 Experimental Results and Analysis.....	83
4.6 Summary.....	88
Chapter 5. Robust Video Sequence Retrieval Using A Novel Object-Based T2D-Histogram Descriptor.....	90
5.1 Introduction.....	90
5.2 Overview of the Proposed Scheme.....	92
5.3 Characterization of Video Segments.....	93
5.3.1 Moving Object Detection.....	94
5.3.2 Describing Motion Activity in a Video Segment.....	95
5.4 Video Sequence Matching.....	97
5.4.1 Discrete Cosine Transform.....	97
5.4.2 Representation of Video Sequences.....	97
5.4.3 Choice of Similarity Measure.....	100
5.5 Experimental Results and Discussions.....	102
5.5.1 Selecting Appropriate Number of DCT Coefficients.....	103
5.5.2 Choosing an Appropriate Motion Activity Descriptor.....	106
5.5.3 Determining the Best Number of Histogram Bins.....	108
5.5.4 Evaluation of Retrieval Performance.....	109
5.6 Summary.....	114
Chapter 6. Robust Video Similarity Retrieval Using Temporal MIMB Moments.....	115
6.1 Introduction.....	115
6.2 Characterization of Video Segments.....	117
6.2.1 Detecting Moving Blobs in MPEG Videos.....	117
6.2.2 MIMB Moments.....	118
6.2.3 Representing Temporal Variations of MIMB Moments.....	119

6.3 Experimental Results.....	120
6.3.1 Choice of Similarity Measure.....	120
6.3.2 Evaluation of Retrieval Performance.....	121
6.4 Summary.....	123
Chapter 7. Conclusions and Future Work.....	124
7.1 Conclusions.....	124
7.2 Future Work.....	124
Reference.....	127



List of Figures

Fig. 1-1. Overview of the proposed approaches.....	2
Fig. 2-1. Proposed system architecture of video abstraction and description.....	7
Fig. 2-2. Structure of typical tennis video program.....	9
Fig. 2-3. GOP-based scene change detection algorithm.....	10
Fig. 2-4. Variation of I-frame DC value of a video sequence (frame0-frame1965)....	12
Fig. 2-5. (a) original I-frame (b) result of tennis court region detection.....	13
Fig. 2-6. The approach of camera motion estimation.....	14
Fig. 2-7. An example of camera pan to the direction of left-bottom (frame 237).....	15
Fig. 2-8. Object localization algorithm.....	19
Fig. 2-9. Demonstration of the result of potential object localization, where frame(a) to frame(h) are numbered as 26, 38, 80, 89, 95, 110, 119 and 125.....	20
Fig. 2-10. Three cases of tracking forward (a) object match in previous P-frame (b) object match in P-frame P_{N-2} (c) object match in P-frame P_{N-3}	22
Fig. 2-11. Three cases of tracking backward (a) object match in next P-frame (b) object match in 2 nd P-frame (c) object match in 3 rd P-frame.....	23
Fig. 2-12. Tennis events inference model.....	24
Fig. 2-13. An example of shape variation of server and receiver.....	24
Fig. 2-14. Hierarchical Summary Description Scheme [17].....	28
Fig. 2-15. The system interface shows an example of tennis event detection.....	31
Fig. 2-16. An example of baseline rally event.....	31
Fig. 2-17. An example of serve and volley event.....	32
Fig. 2-18. An example of passing shot event.....	32
Fig. 2-19. Start-frame of a ball boy running clip.....	33
Fig. 2-20. End-frame of a ball boy running clip.....	33

Fig. 3-1. Overview of the system architecture.....	41
Fig. 3-2. Variation of I-frame DC value (a) tennis; (b) football; (c) baseball.....	43
Fig. 3-3. The approach of closed caption localization.....	44
Fig. 3-4. An original frame is divided into R regions (e.g. $R = 6$).....	46
Fig. 3-5. DCT AC coefficients used in text caption detection.....	46
Fig. 3-6. Demonstration of caption frame detection: (a) small closed caption (b) large closed caption.....	48
Fig. 3-7. Illustration of intermediate results of closed caption localization (a) Original frame (b) Closed caption detection (c) Result after applying morphological operation (d) Result after long-term consistency verification.....	49
Fig. 3-8. Potential caption regions are further verified based on the long-term consistency.....	51
Fig. 3-9. Examples of closed caption localization (a) baseball; (b) news; (c) volleyball	51
Fig. 3-10. Overlap-block is interpolated from its two neighboring blocks B_t and B_b	53
Fig. 3-11. The proposed approach of font size differentiation in compressed domain.....	53
Fig. 3-12. Localized closed captions (a) scoreboard (b) trademark.....	55
Fig. 3-13. Variation of AC energy of the scoreboard in Fig. 3-12(a) ($T=2.2$, $V = 0.05$).....	56
Fig. 3-14. Variation of AC energy of the trademark in Fig. 3-12(b) ($T=2.9$, $V= 0.8$).....	56
Fig. 3-15. Horizontal projection profile of DCT AC energy of the scoreboard and the trademark in Fig.3-12(a) and Fig.3-12(b), respectively.....	57
Fig. 3-16. Hierarchical Summary Description Scheme [43].....	64

Fig. 3-17. Video Content Visualization System was composed of two areas – “Playback” and “Visualization”.....	65
Fig. 3-18. Hierarchical structure of the scoreboards was shown while the user clicks the symbol “+”.....	65
Fig. 3-19. Video shots were presented in the detailed video hierarchy.....	66
Fig. 4-1. System architecture of motion activity based video structuring.....	69
Fig. 4-2. Moving objects detection; (a) anchor person; (b) football; (c) walking person; (d) tennis; (e) volleyball game; (f) traffic monitoring.....	72
Fig. 4-3. Workflow of motion activity descriptor.....	74
Fig. 4-4. An example of 2D-histogram computation.....	74
Fig. 4-5. Histograms of shots; (a) Service; (b) Full-court view; (c) Close-up.....	76
Fig. 4-6. Closed caption localization in video frames.....	78
Fig. 4-7. DCT AC coefficients used in localizing superimposed closed captions.....	79
Fig.4-8. Demonstration of the localization of superimposed closed captions (a) original I-frame; (b) result after filtering by using horizontal gradient energy; (c) result after morphological operation; (d) result after filtering using SOM-based algorithm; (e) result after dilation.....	80
Fig. 4-9. Demonstration of testing videos: (a) Video I (b) Video II.....	84
Fig. 4-10. Closed caption localization; (a) original I-frame; (b) result after filtering by horizontal gradient energy; (c) result after morphological operation; (d) result after filtering by SOM-based algorithm; (e) result after dilation	86
Fig. 4-11. Video structure of caption frames as well as service, full-court view, and close-up shots.....	87
Fig. 4-12. The bottom of the interface shows full-court shots.....	87
Fig. 4-13. The bottom of the interface presents close-up shots.....	88
Fig. 5-1. An overview of extracting the proposed T2D-Histogram descriptor – compressed videos are parsed semantically and represented by reduced	

low-dimensional DCT coefficients.....	92
Fig. 5-2. Demonstration of moving object detection (a) anchor person (b) football (c) walking person (d) tennis competition.....	95
Fig. 5-3. Demonstration of the computation of 2D-histogram.....	96
Fig. 5-4. Video sequences are characterized by the object-based T2D-Histogram descriptor and further represented by reduced low-dimensional DCT coefficients	100
Fig. 5-5. Examples of the Close-Up (CUT), Bicycle Racing (BR), Walking Person (WP) and Anchorperson and Interview (API) shots.....	104
Fig. 5-6. Average retrieval performance with different descriptors ($\beta = 8, \alpha \in [1,5]$) (a) X-histogram (b) Y-histogram (c) 2D-histogram (d) Weighted 2D-histogram...	106
Fig. 5-7. Average retrieval performance ($\alpha=2$) with different number of bins (β) (a) $\beta = 4$ (b) $\beta = 6$ (c) $\beta = 8$ (d) $\beta = 10$	107
Fig. 5-8. Average retrieval performance with parameters: $\alpha=2, D: \textit{weighted 2D-histogram}, \beta \in \{4,6,8,10\}$	109
Fig. 5-9. Retrieval performance of the four shot classes (a) API Shots (b) CUT Shots (c) WP Shots (d) BR Shots and (e) Average.....	111
Fig. 5-10. Demonstration of the query result for a CUT shot.....	112
Fig. 5-11. Demonstration of the query result for a BR shot.....	113
Fig. 5-12. Demonstration of the query result for a WP shot.....	113
Fig. 5-13. Demonstration of the query result for an API shot.....	114
Fig. 6-1. Demonstration of MVF noise reduction (a) MVF without filtering; (b) MVF smoothing with a cascaded filter.....	118
Fig. 6-2. Recall versus precision performance of the three shot classes (a) Interview Shots (b) Close-Up Tracking Shots (c) Walking Person Shots (d) Average.....	122

List of Tables

Table 2-1. Ground truth of the testing video.....	33
Table 2-2. Experimental Results of Tennis Event Inference.....	35
Table 3-1. Performance of caption frame detection.....	59
Table 3-2. Performance of closed caption localization after caption frame detected...	67
Table 3-3. Experimental results of font size differentiation based on horizontal projection profile using vertical DCT AC coefficients.....	62
Table 4-1. Result of shot identification (Video I: 163 shots).....	85
Table 4-2. Result of shot identification (Video II: 199 shots).....	85
Table 4-3. Result of closed caption localization.....	85
Table 5-1. Performance using distinct α and four feature descriptors ($\beta = 8$).....	105
Table5- 2. The performance obtained of four descriptors with different β ($\alpha = 2$)...	108
Table 5-3. Comparison of performance using different numbers of histogram bins (β).....	109
Table 5-4. Retrieval performance using the T2D-Histogram descriptor.....	110

Chapter 1. Introduction

Due to the tremendous growth in the number of digital videos, the development of video retrieval algorithms that can perform efficient and effective retrieval task is indispensable. In this proposal, as shown in the top of Fig. 1-1, we propose an object-based video content parsing and event understanding technique in MPEG compressed videos to support semantic content indexing and abstraction. Its aim is to reliably analyze the semantic video contents. Because moving objects and the corresponding trajectories are the important visual cues for content parsing, methods of object detection and object tracking are proposed using motion features. Therefore, a strategy of object-based event inference is introduced according to the spatio-temporal relationships between objects. Since high-level semantic events are domain dependent, the semantic events are detected and inferred from the long-term consistent spatio-temporal relationships between moving objects utilizing specific domain knowledge. Consequently, video content descriptions for MPEG-7 are generated automatically to support efficient content-based retrieval. Here, we use tennis sports videos as a demonstration of the system. Experimental results show the high accuracy of event detections and justify the effectiveness of the proposed mechanism.

Moreover, since object-based features are semantically more meaningful than other visual features, we propose a high-level motion activity descriptor – 2D histogram, as shown in the middle of Fig.1-1, that exploits both spatial and temporal features of moving objects characterize video sequences in a semantic manner. The Discrete Cosine Transform (DCT) is applied to convert the high-level features from the time domain to the frequency domain. Using this transform, the original high-dimensional time domain features used to represent successive frames are significantly reduced to

the low-dimensional features in frequency domain. The energy concentration property of DCT allows us to use only a few DCT coefficients to precisely represent the variations of moving objects. Having the proposed mechanism and the efficient scheme of video representation, one can perform video retrieval in an accurate and efficient way.

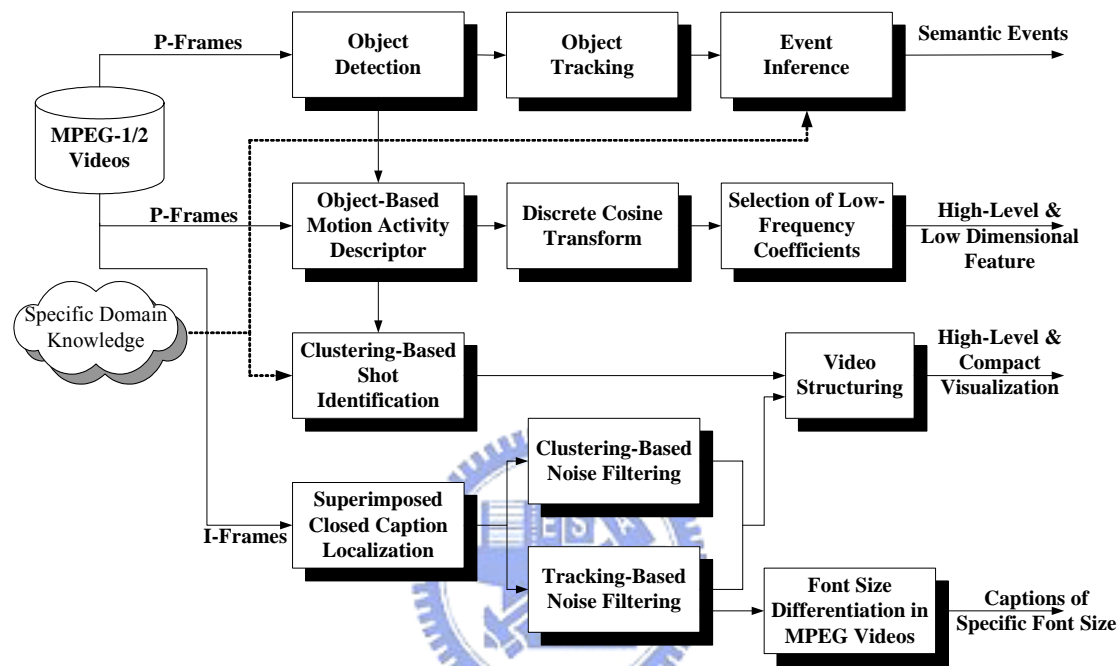


Fig. 1-1. Overview of the proposed approaches

In addition to employing visual features to characterize video shots, textual information in closed captions is also important for users to understand overall video content in a short time. Therefore, as shown in the bottom of Fig. 1-1, a novel approach of automatic closed caption detection and font size differentiation among localized text regions in MPEG videos. The tracking-based noise filtering is exploited to remove the noise of potential captions. When the general closed captions are localized, the designed tool – font size differentiation is used as a filter to assist in the discrimination of the specific and significant text captions, like scoreboards in sports videos.

To provide users a compact form of video content, video structuring is a crucial

step in video content analysis and is the process of extracting temporal structural information of video sequences. It involves detecting temporal boundaries, identifying meaningful segments of a video and then building a compact representation of video content. Therefore, we propose a novel approach to automatically parse MPEG sequences and then to construct a table of video content based on the textual information on superimposed closed captions and the semantic classes of video shots. First, video sequences are efficiently segmented into shots using the approach of GOP-based video segmentation. Each video shot is then characterized to be a novel feature – object-based motion activity, which takes into account the spatio-temporal motion activity among moving objects obtained from motion information of the compressed data. The shots are then classified into semantic classes when the specific domain-knowledge is employed. Finally, a clustering-based algorithm is exploited to distinguish the target captions – superimposed closed captions from the high-textured background regions in the shots of interest. Having the proposed video structuring approach, the system can allow users to browse video sequences at different levels of detail in an efficient way.

The rest of the thesis is organized as follows. Chapter 2 shows the algorithm of video event detection in compressed domains. Effective algorithm of closed caption detection and filtering is illustrated in Chapter 3. Semantic video structuring for volleyball games is introduced in Chapter 4. Two high-level compact video descriptors and their corresponding matching measurements are described in Chapter 5 and Chapter 6, respectively. Finally, Chapter 7 concludes this thesis.

Chapter 2. Automatic Content Parsing and Semantic Event Identification for Sports Video Abstraction and Description

2.1 Introduction

The tremendous growth in the amount of digital videos is driving the need for more effective methods to access and acquire desired video data. Advances in automatic content analysis and feature extraction improve capabilities for effectively searching and filtering videos along perceptual features and semantics. Content-based indexing provides users natural and friendly query, searching, browsing and retrieving. In order to provide users more efficient and effective access methods, it is necessary to support high-level and semantic features for video content representation and indexing. The need of representation and indexing for high-level and semantic features motivates the emerging standard MPEG-7, formally called multimedia content description interface [1]. However, the methods that produce the desired features are non-normative part of MPEG-7 and are left open for research and future innovation.

In many practical queries of MPEG-7 database, high-level and semantic features can support users to acquire desired data more efficiently and effectively. Features of high-level semantics can be extracted and inferred from the closed caption streams [2], the edge information [3], the variation of camera motions and also from spatial-temporal relationship of object locations in uncompressed [4-7] or compressed domain [8-10]. In order to save computation cost and storage space, recently more researches extract features or segment video data directly in compressed video domain [11-13] instead of uncompressed raw data. To support semantic indexing of video content, domain specific knowledge is useful for content identification or annotation and is often applied accordingly. Some researches focus on classification

of video content by identifying significant camera operations [14-15] by using motion vectors of MPEG video streams with specific domain knowledge. In general, distinct camera operations would apply to different kinds of video events. For example, in a basketball game the slam-dunk may correspond to the zoom-in operation and the fast break may be with panning camera motion. However, shots of the same event may be regarded as different kinds of events when these shots are taken by various photographers. Babaguchi et al. [2] search the predefined keywords of American football games in closed caption streams to find out the possible time intervals, which contain the event-shots and subsequently apply the low-level color feature to discover shots similar to predefined events. However, this method would be confronted with some limitations. The target events would be lost due to the reason that the announcer or the commentator may not explain the whole game clearly enough. In addition, target event detection in sports videos based on simple color features would not work well while the court of games is in different colors.

Although these examples of semantic content analysis have achieved certain goals of interest, the features exploited are not general enough. Analyzing video content based on appearance of moving blobs or objects is more general and clearly advantageous since it can show the variation of objects in consecutive frames and even the relationship or event between objects while prior domain knowledge is applied. In addition, few researches focus on video abstraction based on event inference directly from compressed videos. Sports videos contain, besides game competition clips, many clips of commercials, close-up of players or clips that the competition is not actually ongoing. Hence, it is necessary to remove these insignificant clips from the large amount of video sequences so that users can browse or retrieve the desired relevant video data more efficiently.

Therefore, in this chapter, we propose an approach for video abstraction to generate

semantically meaningful video clips and associated metadata. It exploits the efficient mechanism of scene change detection and the effective high-level features of spatial-temporal relationships between objects in MPEG compressed domain. In video segmentation, the proposed GOP-based scene change detection [16] is utilized to segment video streams into shots efficiently since video streams are examined GOP by GOP to detect scene cuts instead of frame by frame and the experimental results show the effectiveness of the approach. Generally, in sports videos, the clips of sports competition are the focus of interest. Shots identification mechanism is proposed to distinguish the interesting shots for further sports event detection. Moreover, objects should be located for event understanding. Based on the concept of long-term consistency of spatial-temporal relationship between objects in consecutive P-frames, the algorithm of multi-object tracking is designed to locate the objects and to generate the trajectory of each object without size constraint. Utilizing the object trajectory coupled with domain knowledge, the event inference module detects and identifies the events in the application of tennis sports. Consequently, the event information and metadata of associated video clips are extracted and the abstraction of video streams is accomplished. Furthermore, video content descriptions and description schemes based on the Hierarchical Summary Description Scheme [17] in MPEG-7 are generated automatically to support high-level video content indexing, retrieval and browsing.

The rest of the chapter is organized as follows. The overview of the proposed video abstraction approach is described in section 2.2 and the video segmentation and shots identification are presented in section 2.3. Section 2.4 presents the method of global motion estimation and section 2.5 describes the object-tracking algorithm. Experimental results and discussions are shown in section 2.6. Conclusion and future work are given in section 2.7.

2.2 Overview of The System Architecture

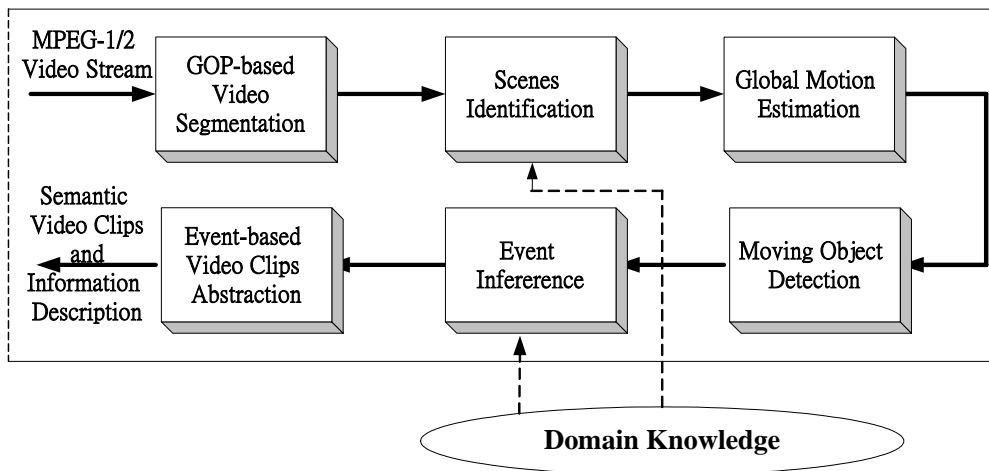
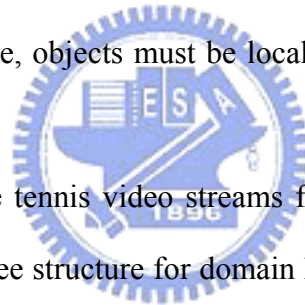


Fig. 2-1. Proposed system architecture of video abstraction and description

Fig. 2-1 shows the proposed system architecture for event-based semantic abstraction of videos. Video streams are first segmented into shots using the proposed GOP-based video segmentation and segmented shots are further classified using the color-based scene identification. In general, sports shots can be classified as two types according to the color features. The first type is the shot consisting of the competition court or field whose color variation is small throughout the whole shot and the second type is the shot including the commercials, close-up shots, the crowd, etc. in which the color variation is relatively significant. Significant video clips that contain competition court are usually the shots of interest and are thus selected for further event inference. In order to reduce computation cost, objects are detected using the motion information in P-frames. However, in sports videos, the camera is not static because it may pan or tilt to capture the players. To localize the positions of objects robustly, camera motion must be estimated. Instead of exploiting the motion estimation model such as affine motion model, the camera motion indicated by the dominant motion is characterized using the histogram-based method, in which motion vectors in P-frames are directly extracted and used for camera motion estimation.

Moreover, after objects are tracked, the trajectories of objects in a video shot can be

obtained and be exploited to infer high-level semantic events of sports videos with the aid of domain knowledge. Video shots are distinguished into semantically meaningful clips based on the events inferred in the previous phase. After the thorough procedure, semantic video clips are obtained and the associated high-level metadata can be used for automatic generation of video descriptions, video indexing and video abstraction. For example, three major events in tennis games are: serve and volley, baseline rally and passing shot. Players always staying near baseline are considered as baseliners and thus the corresponding event is regarded as baseline-rally. When one of the players is a serve-and-volleyer, the event would be serve-and-volley or passing shot according to the final position of the serve-and-volleyer. These events are defined in terms of not only objects appearing in a time interval but also spatial relationships between the objects. Therefore, objects must be localized in a time point and further be tracked in a time interval.



In the experiments, we use tennis video streams formatted in MPEG-2 as testing sequences and its' temporal tree structure for domain knowledge is shown in Fig. 2-2. A match can be played to the best of some sets (the player needs to win two sets out of three in order to win the match or to win three sets out of five in order to win the match). A set consists of several games (say six games) and a game is made up of some points (say four points) [18]. It is worth noting that such a tree can be constructed for any kind of sports games. The proposed object-based video analysis scheme can be applied to most kind of sports games and even the well-structured videos such as news because their video sequences can be structured as a tree and the video content can be modeled or described using objects. Therefore, given the structure and the domain knowledge, we are able to adapt the event detection scheme for specific application domain. The details of each module are explained in the following sections.

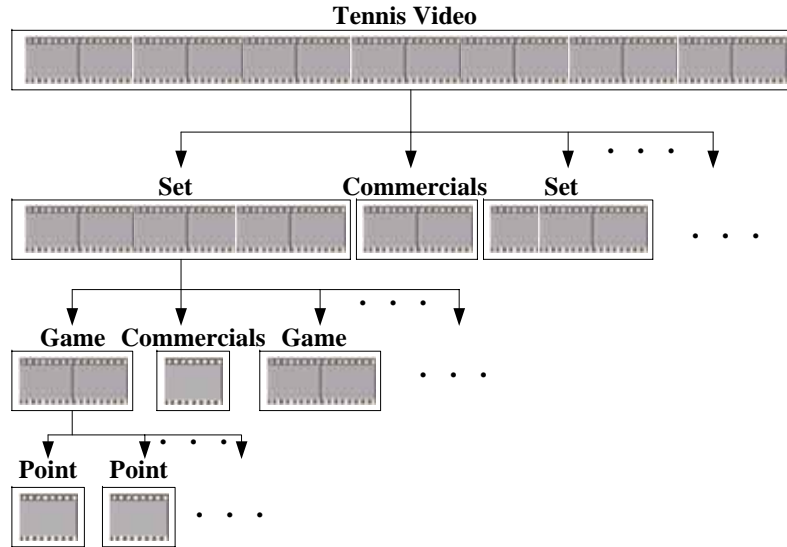


Fig. 2-2. Structure of typical tennis video program

2.3 Video Segmentation and Shots Selection

2.3.1 GOP-Based Video Segmentation

Video data is segmented into clips to serve as logical units called “shots” or “scenes”. Fig. 2-3 illustrates our proposed GOP-based scene change detection approach [16]. In MPEG-II format [19-20], GOP layer is a random accessed point and contains GOP header and a series of encoded pictures including I, P and B-frames. The size of a GOP is about 10 to 20 frames, which is less than the minimum duration of two consecutive scene changes (about 20 frames) [21].

We first detect possible occurrences of scene change GOP by GOP (inter-GOP) instead of frame by frame to speed up the computation. The difference between each consecutive GOP-pair is computed by comparing the I-frames in each consecutive GOP-pair. If the difference of DC coefficients between these two I-frames is larger than the threshold, then there may have scene change in between these two GOPs. Hence, the GOP that contains the scene change frame is located. In the second step – intra GOP scene change detection, we further compute the ratio of forward to backward and the ratio of backward to forward motion vectors in B-frames. By

comparing the two ratios with predefined thresholds, the actual frame of scene change within a GOP can be located. The experimental results in [16] are convincing and justify that the efficiency and the effectiveness of video segmentation.

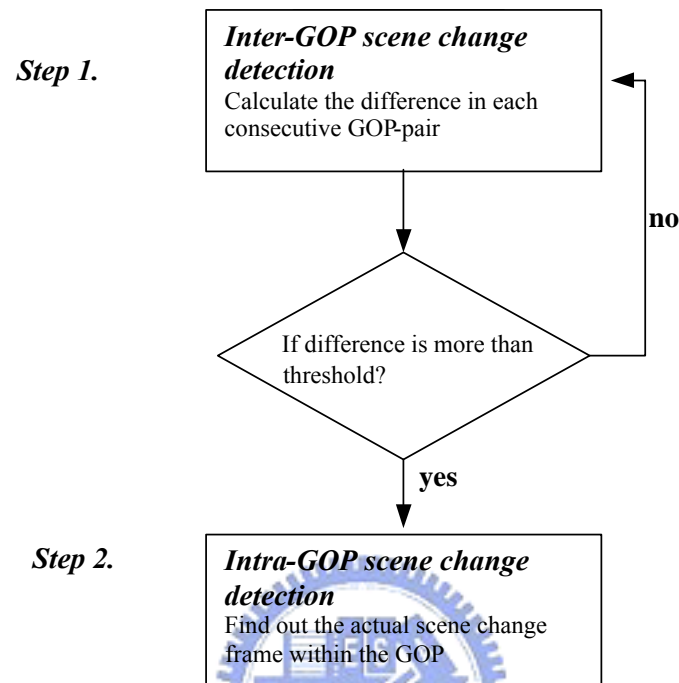


Fig. 2-3. GOP-based scene change detection algorithm

2.3.2 Scene Identification

While the boundary of each shot is detected, the video sequence is segmented into shots consists of various types of clips, which need further processing to identify the scenes. In order to detect and infer events, application domain of interest needs to be specified and knowledge model needs to be incorporated. Taking sports videos as an example, such as tennis, football and baseball, the clips might be commercials, close-up shots and competition court shots. However, commercials may not be interesting to clients and only the ongoing competition shots in sports games are clients' concern. Hence, only the clips of interest are meaningful and need to be processed and analyzed further. Therefore, scene identification is to recognize the clips of the type desired (say competition court shots).

Focusing on tennis games, we observe that the variation of the intensity of the

tennis court frame is very small through the whole clip and the value of intensity variation between consecutive frames is very similar. In contrast, the intensity of the commercial clips and close-up clips varies significantly in each frame and the difference of the intensity variance between two neighboring frames is relatively large. Therefore, the DC-image of each I-frame, which consists of DC coefficients of each block, is used to compute the intensity variation of I-frames. In addition to the intensity variance of each I-frame, the variance of each shot is also computed to be the shot feature. The definition of the frame variance and that of shot variance are shown in Eq. (2-1) and Eq. (2-2). $DC_{i,j}$ means the j th block of the i th frame and N represents the total number of blocks in a frame. $FVar_{s,i}$ is the intensity variance of the frame i in shot s and the variance of shot s is expressed by $SVar_s$, where M is the total number of frames in shot s . The variation of the intensity variance of each I-frame in a video sequence from frame-0 to frame-1965 is exhibited in Fig. 2-4. In the video sequence, four clips of tennis court are marked by the dotted ellipses and the close-up clips are marked by the dotted rectangles. The last clip of this sequence is an advertisement clip signed by the dotted circle. From Fig. 2-4, we can see that the intensity variance of the tennis court clips is very small and the intensity values of them are very similar through the whole clip. Thus, the clips of tennis court can be indicated and selected by the characteristic of the value of intensity variance being small in each frame and permanent through the shot.

$$FVar_{s,i} = \sum_{j=1}^N DC_{i,j}^2 / N - \left(\sum_{j=1}^N DC_{i,j} / N \right)^2 \quad (2-1)$$

$$SVar_s = \sum_{i=1}^M FVar_{s,i}^2 / M - \left(\sum_{i=1}^M FVar_{s,i} / M \right)^2 \quad (2-2)$$

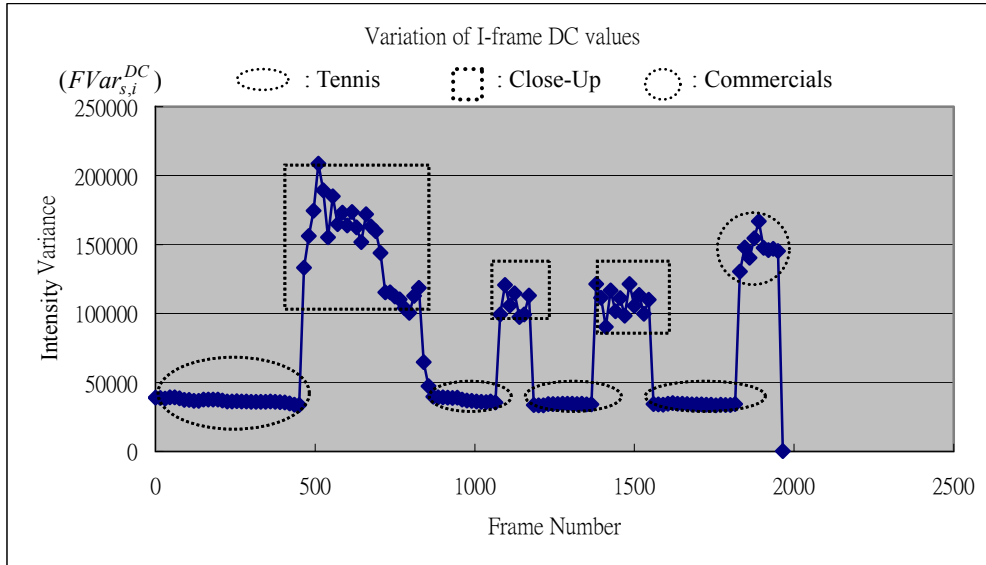


Fig. 2-4. Variation of I-frame DC value of a video sequence (frame0-frame1965)

2.4 Camera Motion Compensation

Camera motion estimation is a necessary and important step for object localization. To compute the camera motion of shots, generally motion vectors of all MBs in P-frames are used for estimation. However, most regions in consecutive frames of competition court clips are very similar and thus motion information of these regions cannot actually reflect the information of global motion. Therefore, the adaptive threshold decision scheme is proposed for camera motion estimation. In addition, the mechanism of dominant motion computation based on histogram is proposed to estimate the camera motion efficiently. Section 2.4.1 presents the approach of adaptive threshold decision and the approach of camera motion estimation is described in section 2.4.2.

2.4.1 Adaptive Threshold Decision

In order to select the threshold for the global motion estimation adaptively, we need to detect the outline regions that their intensity is different from the region of competition court. Hence, the DC coefficient of each block in the first I-frame of each competition court shot is extracted and used to represent the block intensity. The

adaptive threshold decision is defined in Eq. (2-3) where T_{global} means the threshold for global motion estimation, N represents the number of macroblocks in an I-frame and α can be set to a half of the outline region or larger than that because most regions (say more than half) would have similar motion directions when the camera pans or tilts.

$$T_{global} = \alpha \left(\sum_{i \in [1, N]} MB_i - \sum_{j \in court} MB_j \right) \quad (2-3)$$

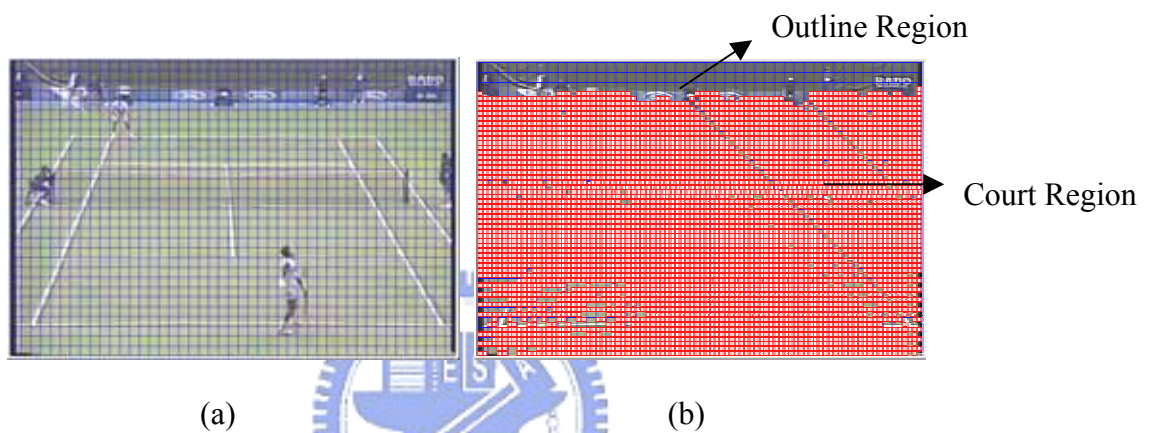


Fig. 2-5. (a) original I-frame (b) result of tennis court region detection

An example of the outline region detection is demonstrated in Fig. 2-5. The largest region is the region of tennis court as marked in the bottom of Fig. 2-5 (b) and other unmarked regions in the top of Fig. 2-5(b) belonging to the outline regions are used for adaptive threshold decision.

2.4.2 Camera Motion Estimation

To correctly locate the position of players, camera motion should be estimated to compensate players for the camera motion. In this section, a fast and simplified camera motion detection approach is proposed. Fig. 2-6 shows the procedure of the camera motion detection. For the computation efficiency, only the motion vectors of P-frames are used for camera motion analysis since in general, in a 30 fps video consecutive P-frames separated by two or three B-frames, are still similar and would

not vary too much. Therefore, it is sufficient to use the motion information of P-frames only to detect camera motions.

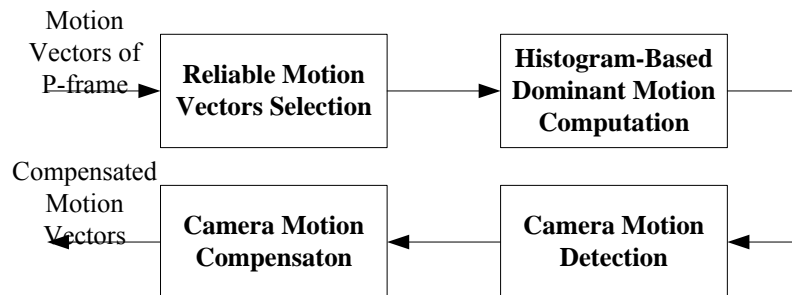


Fig. 2-6. The approach of camera motion estimation

However, the motion vectors of P-frames or B-frames in MPEG-2 compression standard are best match and may not actually represent correct motions in a frame because the motion estimation in MPEG videos is for the purpose of data compression. This problem in the sports video streams is more serious since consecutive frames in competition court clips are very similar. This will lead to the situation that for a macroblock in competition court, it is easy to find a good match around its neighbor in the reference frame. However, this motion estimation does not mean that the position of the macroblock is correctly located in its reference frame. Therefore, in order to achieve more robust analysis, it is necessary to select the regions that do not belong to the area of competition court for global motion estimation, since the motion vectors of the area of competition court cannot actually reflect the global motion. Taking the tennis court as an example, in Fig. 2-7, we can observe that motion vectors in the upper part of the frame are more reliable since these macroblocks are of similar motion vector magnitude and direction, but in most of the macroblocks within the area of tennis court, the magnitudes and directions are not consistent and are very noisy.

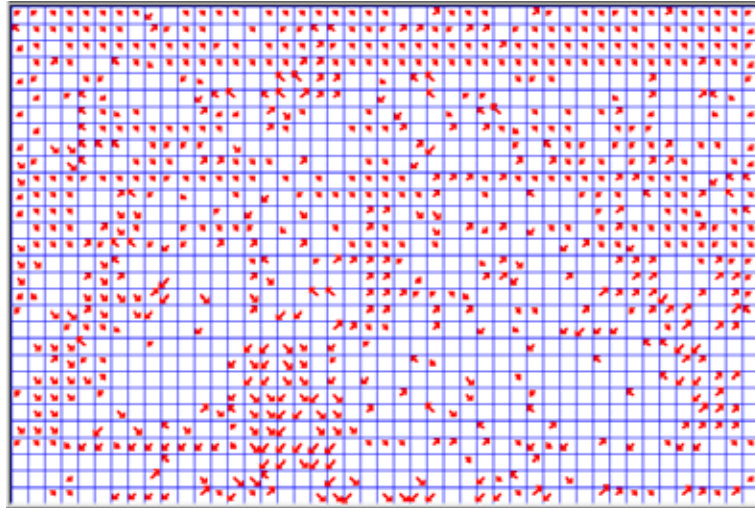


Fig. 2-7. An example of camera pan to the direction of left-bottom (frame 237)

The histograms of magnitude and direction of motion vectors in the outline region, which are more reliable for camera motion estimation, are computed to acquire dominant motion direction and dominant motion magnitude to further identify whether camera motion, pan and tilt, happens or not. Using the approach of histogram-based dominant motion computation, we can avoid matrix multiplications, which are computationally inefficient when motion vectors are fit to affine motion models. Furthermore, pan and tilt are two major camera motions in most sports and can be detected fast and correctly by the proposed motion vector histogram-based approach. The threshold T_{global} that is adaptively decided is used to identify the existence of camera motion in a frame. The magnitude and direction of camera motion are obtained by using Eq. (2-4) and Eq. (2-5).

$$SDMH_i = \#(Bin_{DMH-1,i}) + \#(Bin_{DMH,i}) + \#(Bin_{DMH+1,i}) \quad (2-4)$$

$$SDAH_i = \#(Bin_{DAH-1,i}) + \#(Bin_{DAH,i}) + \#(Bin_{DAH+1,i}) \quad (2-5)$$

DMH means the dominant magnitude of motion vector histogram, *DAH* the dominant direction of motion vector histogram, $SDMH_i$ the summation of three bins ($Bin_{DMH-1,i}$, $Bin_{DMH,i}$ and $Bin_{DMH+1,i}$) of magnitude histogram of the i^{th} frame, $SDAH_i$ the summation of three bins ($Bin_{DAH-1,i}$, $Bin_{DAH,i}$ and $Bin_{DAH+1,i}$) of direction histogram of the i^{th} frame, and $\#(Bin_{j,i})$ represents the value of the j^{th} bin in the i^{th} frame.

In the ideal situations, macroblocks in an object would have the same motion magnitude and direction. However, although the entire object moves toward the same direction, some regions in the object might have different but similar motion magnitudes and direction because objects in real world are not rigid in their shape and size. Consequently, to tolerate the error of motion estimations, the values of $Bin_{DMH-1,i}$, $Bin_{DMH,i}$ and $Bin_{DMH+1,i}$ of magnitude histogram ($Bin_{DAH-1,i}$, $Bin_{DAH,i}$ and $Bin_{DAH+1,i}$ of direction histogram) are summed to examine whether the summation $SDMH_i$ ($SDAH_i$) is larger than the threshold or not. If $SDMH_i$ and $SDAH_i$ are both larger than the threshold T_{global} , camera motion happened, and *DMH* and *DAH* are identified as magnitude and direction of camera motion in frame i . Moreover, motion vectors are compensated with the magnitude and direction of camera motion for further player detections.

2.5 Events Detection and Description

To infer events of sports games, we need to track the positions of players in consecutive frames and generate a trajectory for each player. However, the intrinsic

problem of motion estimation in MPEG-2 standard mentioned in the previous section makes players tracking difficult. Moreover, the difficulty is also due to the varied shape or size of players in consecutive frames. Therefore, in order to solve these problems, we propose a robust algorithm to track players in consecutive P-frames. Focusing on tennis videos, we have to recognize the server further by utilizing the proposed algorithm of server and receiver differentiation. The object-tracking algorithm is introduced in section 2.5.1 and the server-receiver differentiation algorithm is shown in section 2.5.2. The description scheme and descriptor in MPEG-7 for tennis game are presented in section 2.5.3.

2.5.1 Object Tracking Algorithm

2.5.1.1 Object Localization

Object localization algorithm is to locate potential objects in video shots for subsequent object tracking. The overview of the algorithm of potential object localization is shown in Fig. 2-8. Initially, we verify if there is any camera motion of each P-frame and compensate motion vectors with global motion if camera motion happens. Otherwise, noisy motion vectors are eliminated directly without motion compensation. Subsequently, motion vectors that have similar magnitude and direction are clustered together and this group of associated macroblocks of similar motion vectors is regarded as a potential object. Details are presented in the object localization algorithm.

Object Localization Algorithm

Input: N P-frames of a video clip $\{P_1, \dots, P_N\}$

Output: N object sets $\{Obj_{1,n_1}\}, \{Obj_{2,n_2}\}, \dots,$ and $\{Obj_{N,n_N}\}$, where N is total number of P-frames and Obj_{j,n_j} means the n_j^{th} object of the j^{th} P-frame. Each object size is measured in terms of number of macroblocks.

1. Analyze motion vector of inter-coded macroblocks in a P-frame to see if there is any camera motion.
2. If there is no camera motion, go to step 3. If camera motion is detected, motion vectors that are not noisy are compensated with camera motion magnitude and direction.
3. Cluster motion vectors that are of similar magnitude and direction into the same group with region growing approach.

MV_1	MV_2	MV_3
MV_4	Center	MV_5
MV_6	MV_7	MV_8

3.1 Set search windows (W) size 3x3 macroblocks

3.2 Search all macroblocks (MB) within W , and compute the difference ($diffMag_k$ and $diffAng_k$) of motion vector magnitude ($|MV|$) and direction ($\angle MV$) between center MV_{center} and its neighboring eight motion vectors MV_k within W .

$$diffMag_k = abs(|MV_{center}| - |MV_k|)$$

$$diffAng_k = abs(\angle MV_{center} - \angle MV_k), \text{ where } k \in [1,8] \text{ and } MV_{center} \text{ is the}$$

motion vector in the center position of W

$MV_k \in$ motion vectors within W except MV_{center}

$$\text{For all } k \in [1,8], \text{ flag } F_k = \begin{cases} 1, & diffMag_k < T_{Mag} \text{ and } diffAng_k < T_{Ang} \\ 0, & \text{otherwise} \end{cases}$$

, where T_{Mag} is the predefined threshold for motion vector magnitude

and T_{Ang} is the threshold for motion vector direction

If $\sum_{k=1}^8 F_k \geq 6$, mark F_{center} of MV_{center} as 1, where F_{center} is the flag of the center motion vector within W .

Otherwise, set all flags within W to 0.

3.3 Go to step 3.2 until all MBs are processed.

- 3.4 Group MBs that are marked as 1 into the same cluster.
- 3.5 Compute each object center and record its associated macroblocks.
- 3.6 Generate one object set for each P-frame.
4. Go to step 1 until all P-frames are processed.

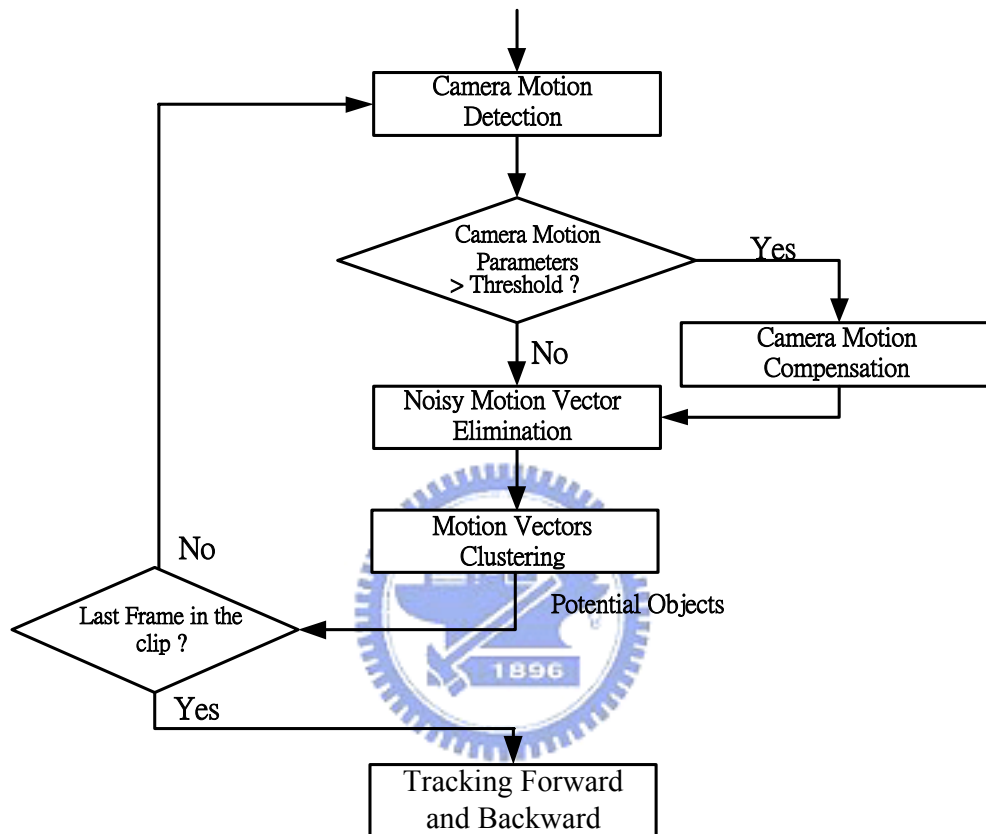


Fig. 2-8. Object localization algorithm

By applying the object localization algorithm, potential objects are located for each frame and the result is demonstrated in Fig. 2-9. Potential objects are marked by the bold-line rectangles. We can see that two players are localized except for the frame of Fig. 2-9(g), in which the top player is not detected. Since the top-player may turn and twist his body and its shape changes dramatically, therefore its associated macroblocks cannot find the matched macroblocks. Besides, some noisy objects also appear in these frames. However, our target is to locate the two players. In order to automatically recognize the two players and filter out noisy objects, long-term

consistency of the spatial-temporal relationship of objects in consecutive frames is employed as the measurement to check if two objects in successive frames are the same one. Therefore, the forward and backward object-tracking algorithm based on long-term consistency is proposed and is described in the following section.

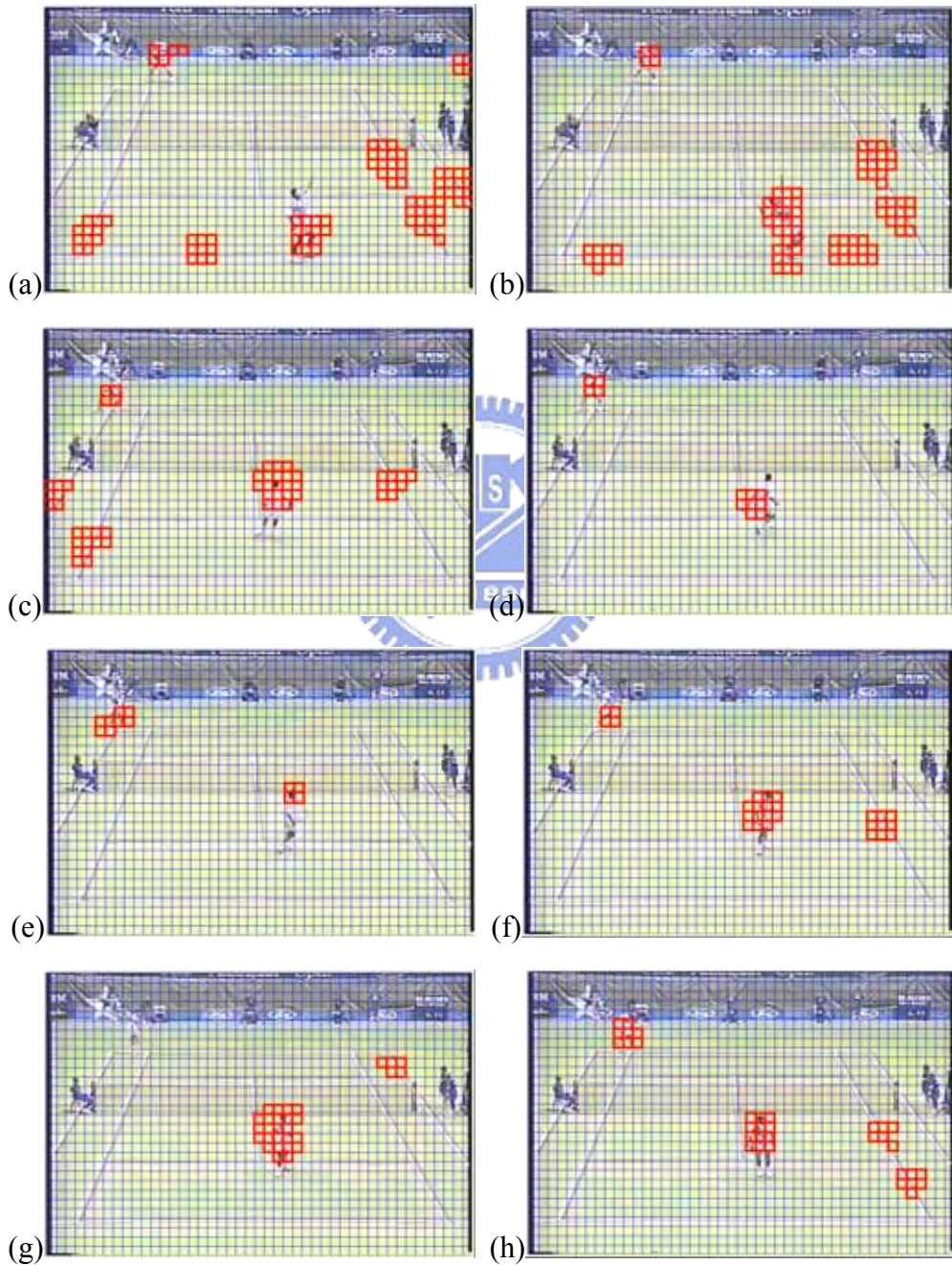


Fig. 2-9. Demonstration of the result of potential object localization, where frame(a) to frame(h) are numbered as 26, 38, 80, 89, 95, 110, 119 and 125.

2.5.1.2 Object Tracking Forward and Backward

While potential objects are located, those objects that are not long-term consistent are regarded as noise and should be removed from the candidates. To compute the long-term consistency of each object, the motion information of each object in P-frames is used to track forward and backward. The forward and backward object-tracking algorithms are demonstrated in Fig. 2-10 and Fig. 2-11 respectively. In Fig.2-10(a), the first case is that object i $Obj_{N,i}$ of P-frame P_N matches an object j $Obj_{N-1,j}$ of P-frame P_{N-1} by using the motion vector $MV_{N,i}$ and object $Obj_{N-1,j}$ continues to search if any object matched in the previous P-frame. However, if there is no match for $Obj_{N,i}$ in P_{N-1} , $Obj_{N,i}$ searches if any object matched in P_{N-2} by using the motion vector $2MV_{N,i}$ which is weighted by the frame distance between target and reference frames. While there is an object $Obj_{N-2,j}$, which matches the object $Obj_{N,i}$, the frame P_{N-2} is set as the target frame and $Obj_{N-2,j}$ continues to find if any object matched in the previous P-frame. The concept of the third case is similar to the 2nd case except that the weighted motion vector is $3MV_{N,i}$. Furthermore, if $Obj_{N,i}$ cannot find any matched object in the previous three P-frames, the procedure of object tracking for $Obj_{N,i}$ is terminated.

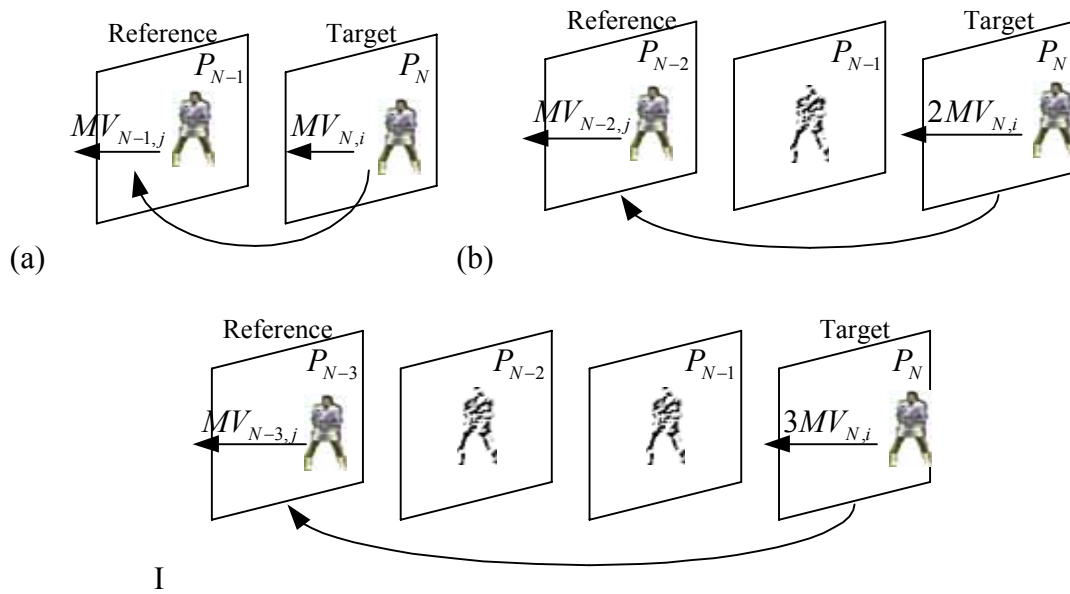


Fig. 2-10. Three cases of tracking forward (a) object match in previous P-frame (b) object match in P-frame P_{N-2} (c) object match in P-frame P_{N-3}

The procedure of tracking backward is shown in Fig. 2-11 and the three cases are analogous to those of tracking forward. However, the reference direction of inter-coded macroblocks is forward reference and thus we can just use the motion information of objects of next frame to trace forward to previous frame while we want to realize backward tracking. Hence, in Fig. 2-11, the dotted line illustrates conceptually the backward object tracking from target frame to reference frame. In Fig. 2-11(a), all objects in frame P_{N+1} are searched to see if any object matches the object $Obj_{N,i}$. However, in Fig. 2-11(b), if there is no match in frame P_{N+1} , the objects in P_{N+2} are sought to find the matching object by using the weighted motion vector $2MV_{N+2,j}$. The case in Fig. 2-11(c) is similar to the 2nd case and the frame distance 3 weights the motion vector $MV_{N+3,j}$ and the procedure of backward object tracking is terminated when there is no match for $Obj_{N,i}$ in consecutive three P-frames.

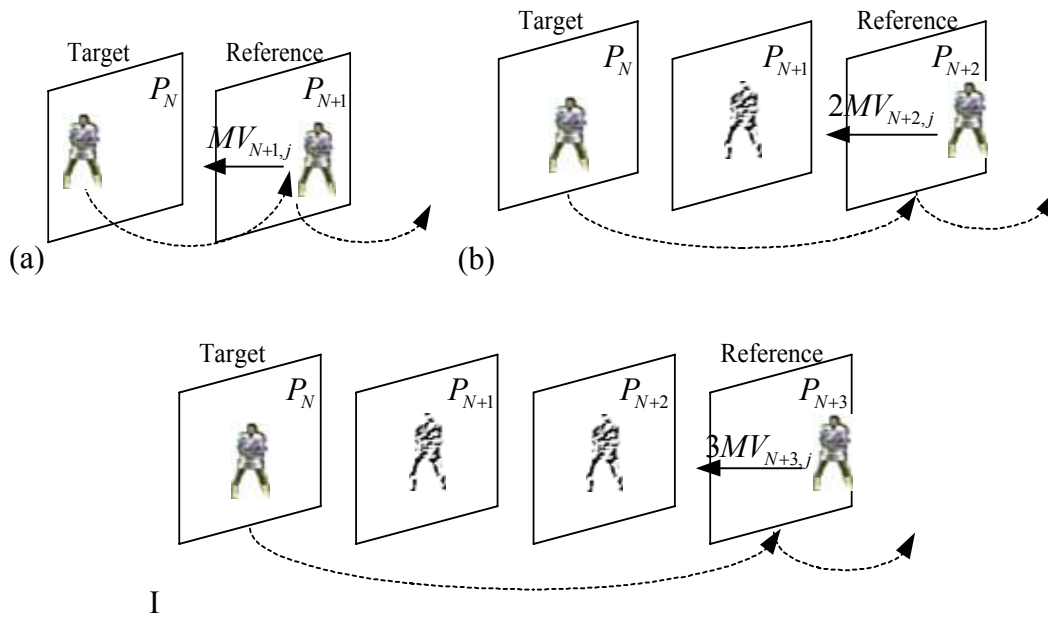


Fig. 2-11. Three cases of tracking backward (a) object match in next P-frame (b) object match in 2nd P-frame (c) object match in 3rd P-frame

By applying the algorithm of forward and backward tracking, we may generate several trajectories of each object. However, based on the long-term consistency of objects, the longest trajectory is what we concern and hence other trajectories of the object are ignored. In addition, the longest two trajectories of objects are kept and these two objects are regarded as the two players.

2.5.2 Events Inference Model

While the object trajectory is acquired, we can infer video events from the object trajectory by applying some domain knowledge. Thus an event inference model, as shown in Fig. 2-12, is designed to infer events of tennis game from two trajectories of top and bottom players. In this chapter, three events of interest are identified: “serve and volley”, “baseline rallies” and “passing shot” since they are the major occurrences in tennis competitions. Notice that it is necessary to distinguish between server and receiver before event inferences. Server should be located for server related events, “serve and volley” and “passing shot”. Therefore, we propose an algorithm to differentiate between server and receiver based on the observation that the shape of

server varies more than receiver in consecutive P-frames from “two players ready” state to “one player serves” state.

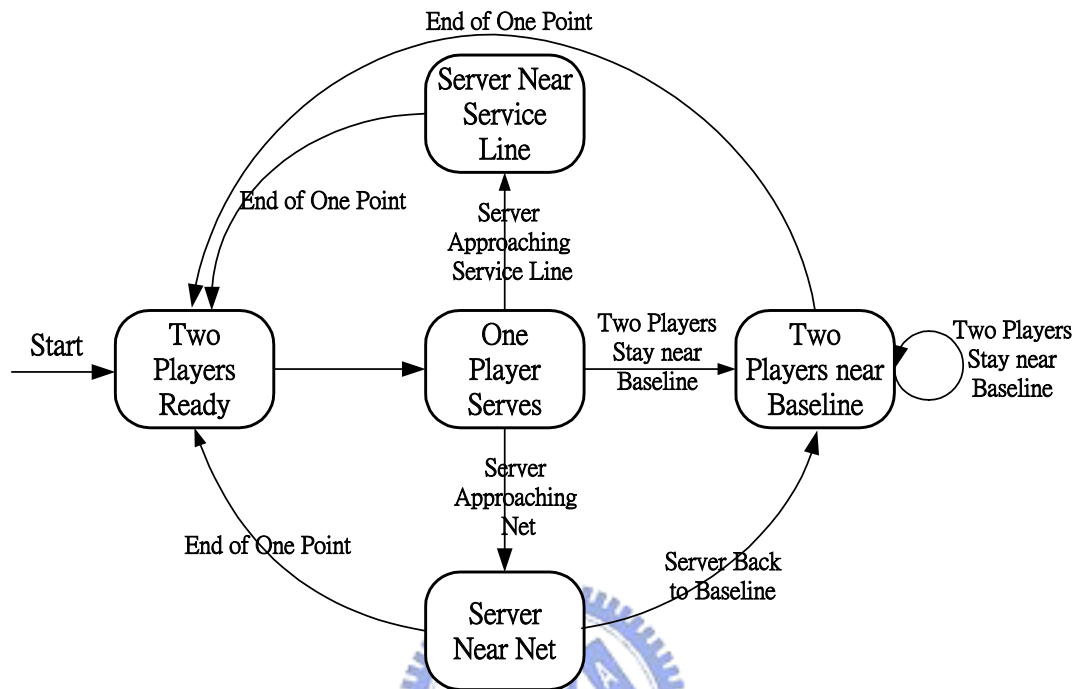


Fig. 2-12. Tennis events inference model

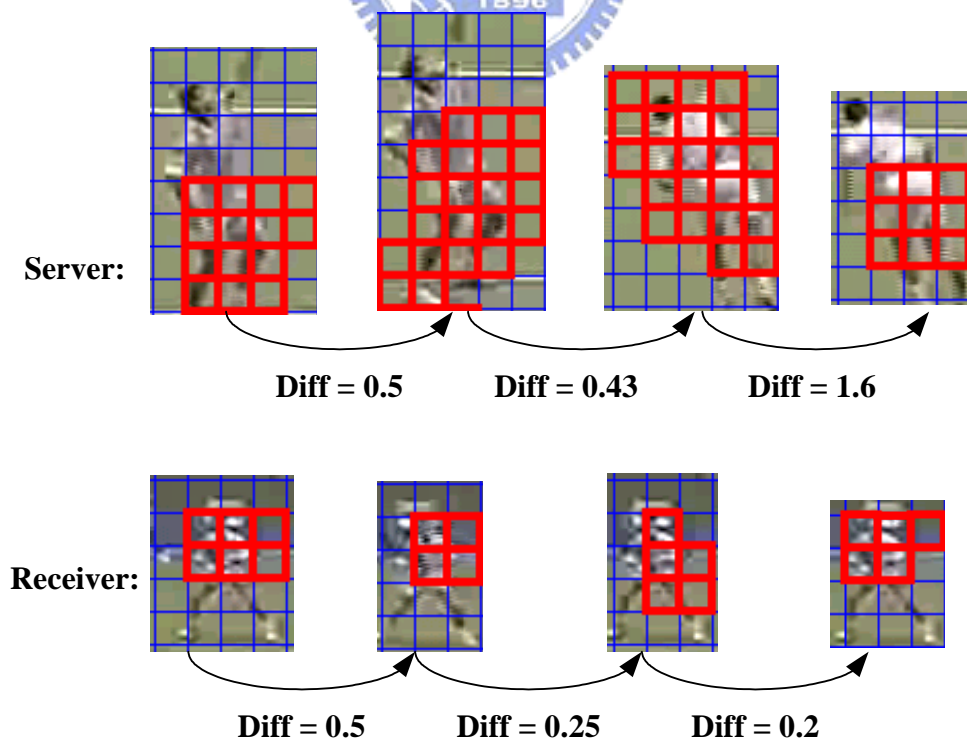


Fig. 2-13. An example of shape variation of server and receiver

Algorithm: Server and Receiver Differentiation Algorithm

Input: Top player $\{TP_1, TP_2, \dots, TP_N\}$ and bottom player $\{BP_1, BP_2, \dots, BP_N\}$ in consecutive P-frames $\{P_1, \dots, P_N\}$

Output: {Server, Receiver}

1. Set $i=0$, $Stop = 0$, $TP_{prob} = 0$ and $BP_{prob} = 0$
2. Do until $Dist_{i-1,i}^{TP} \geq Threshold$ or $Dist_{i-1,i}^{BP} \geq Threshold$

$$i = i + 1$$

Compute the center position of Top Player TP_i and Bottom Player BP_i respectively

$$C_{TP,i}(x, y) = \frac{1}{m} \sum_{j=1}^m MB_{i,j}(x, y), \quad C_{BP,i}(x, y) = \frac{1}{n} \sum_{j=1}^n MB_{i,j}(x, y)$$

, where $\{MB_{i,1}, MB_{i,2}, \dots, MB_{i,m}\} \in TP_i$ and $\{MB_{i,1}, MB_{i,2}, \dots, MB_{i,n}\} \in BP_i$

m is the number of MBs in TP_i and n is the number of MBs in BP_i

3. If $Dist_{i-1,i}^{TP} = \| C_{TP,i-1}(x, y) - C_{TP,i}(x, y) \| < Threshold$ and

$$Dist_{i-1,i}^{BP} = \| C_{BP,i-1}(x, y) - C_{BP,i}(x, y) \| < Threshold \quad \text{Then}$$

Compute $TP_{i-1} \otimes TP_i$ and $BP_{i-1} \otimes BP_i$

If $Norm(\sum (TP_{i-1} \otimes TP_i)) < Norm(\sum (BP_{i-1} \otimes BP_i))$ Then

$$BP_{prob} = BP_{prob} + 1$$

Else If $Norm(\sum (TP_{i-1} \otimes TP_i)) > Norm(\sum (BP_{i-1} \otimes BP_i))$ Then

$$TP_{prob} = TP_{prob} + 1$$

4. If $TP_{prob} > BP_{prob}$ Then **Server = TP**

Else **Server = BP**

In the server and receiver differentiation algorithm, we first compute the center of top and bottom player. The distance of top player (bottom player) between consecutive P-frames is computed in the third step. If both the distance $Dist_{i-1,i}^{TP}$ and $Dist_{i-1,i}^{BP}$ are smaller than β macroblocks (say three), it means that players do not actually move and are still in “two players ready” state. In order to obtain the shape variations of two players, we utilize the exclusion Boolean operation \otimes to compute the shape difference between consecutive P-frames. The center of TP_{i-1} and TP_i (BP_{i-1} and BP_i) are overlapped and macroblocks in TP_{i-1} and TP_i (BP_{i-1} and BP_i) are excluded ($TP_{i-1} \otimes TP_i$ and $BP_{i-1} \otimes BP_i$). The exclusion results of each macroblock-pair are summed to be the shape difference between frame $i-1$ and i . However, usually one player, either the server or receiver, is closer to the camera than the other one and the shape of the player closer to the camera would be larger in size. Therefore, to prevent the object size from being taken into account, the summation of the exclusive results should be normalized by the object size which is defined as the average of the minimum size between the object pair TP_{i-1} and TP_i . The equation of normalization is defined in Eq. (2-6) and Eq. (2-7). To manifest the size variation of objects between consecutive P-frames, the shape difference $\sum(TP_{i-1} \otimes TP_i)$ or $\sum(BP_{i-1} \otimes BP_i)$ is normalized by the minimum value of the size of the object pair instead of normalizing by the average or maximum size.

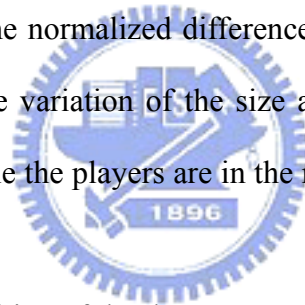
$$Norm(\sum(TP_{i-1} \otimes TP_i)) = \sum(TP_{i-1} \otimes TP_i) / Min(|TP_{i-1}|, |TP_i|) \quad (2-6)$$

$$Norm(\sum(BP_{i-1} \otimes BP_i)) = \sum(BP_{i-1} \otimes BP_i) / Min(|BP_{i-1}|, |BP_i|) \quad (2-7)$$

By applying the proposed server-receiver differentiation algorithm, bottom player is a potential server if its shape difference is larger than that of top player and hence

its possibility value BP_{prob} is incremented. In contrast, if the shape difference of top player is larger than that of bottom player, the potential server is the top player and TP_{prob} is incremented. Subsequently while the distance $Dist_{i-1,i}^{TP}$ or $Dist_{i-1,i}^{BP}$ is larger than threshold, two players are recognized as starting moving. The possibility values (TP_{prob} and BP_{prob}) of top and bottom player are examined to indicate which player is the server. Top player is the server if TP_{prob} is larger than BP_{prob} and bottom player is the server if BP_{prob} is larger than TP_{prob} .

An example of the result of the server-receiver differentiation algorithm is demonstrated in Fig. 2-13. The detected object is represented by the bold-line rectangles. We can see that the normalized difference of the server is larger than the receiver and it means that the variation of the size and shape of the server is more obvious than the receiver while the players are in the ready state of the state transition diagram.



The following are the definition of the three events.

1. Baseline Rally

Baseline Rally means that two players stay near baseline of tennis court in consecutive frames. We can infer baseline rally event from the trajectories of two players while these two trajectories within a video clip are near the baseline, i.e. the state transfers from initial state – two players ready state – to one player serves state and finally falls in “two players near baseline” state.

2. Serve and Volley

The event server and volley – means that the server serves and then approaches the net to volley, i.e. from “one player serves” state to “server near net” state in the inference model.

3. Passing Shot

Passing shot means that a player approaches the service line and then stops. We can infer this event from a trajectory, which ends its path near the service line, i.e. the state finally falls in “server near service line” state.

2.5.3 Event Description Scheme

Tennis event description is based on the Hierarchical Summary Description Scheme of MPEG-7 as shown in Fig. 2-14. While the event of each shot of tennis competition is inferred, the information of the type of events, the boundary of events and the key frame of events are generated automatically and the information can be used in the Highlight Level Description Scheme to support users' query by high-level semantic features.

The description scheme of tennis game is demonstrated in the following. The part in boldface is the highlight level description scheme that can be generated without any manual participation. The name of highlight corresponds to the type of tennis event, the descriptor of video segment locator is described by the event boundary and the position of the key frame in the video sequence is used for the key image locator.

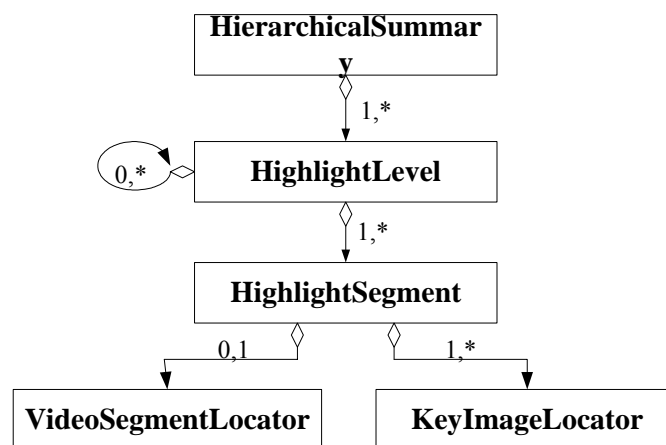


Fig. 2-14. Hierarchical Summary Description Scheme [17]

```

<?XML version= "1.0"><!DOCTYPE MPEG-7>
<Program>
  <MediaInformation>
    <MediaProfile>
      <MediaInstance>
        <Locator>file://sports/tennis/v1.mpg</Locator>
      </MediaInstance>
    </MediaProfile>
    <Players>Pete Sampras vs. Todd Martin</Players>
    <Classification>Sports: Tennis</Classification>
  </MediaInformation>
</Program>
<Summarization>
  <HierarchicalSummary>
    <HighlightLevel name= "Baseline Rally">
      <HighlightSegment name= "Baseline Rally #1">
        <VideoSegmentLocator>
          <MediaTime> 0 430 </MediaTime>
        </VideoSegmentLocator>
        <KeyImageLocator>
          <MediaTime>0</MediaTime>
        </KeyImageLocator>
      </HighlightSegment>
      ... <!-- more video clips -->
    </HighlightLevel>
    <HighlightLevel name="Serve and Volley">
      ...
    </HighlightLevel>
    <HighlightLevel name="Passing Shot">
      ...
    </HighlightLevel>
    ... <!-- more HighlightLevel -->
  </HierarchicalSummary>
</Summarization>

```

2.6 Experimental Results and Discussion

In the experiments, we take MPEG-2 compressed video streams as the testing sequences. The video streams obtained from Star-Sports TV channel are encoded

adopting the GOP structure of IBBPBBPBBPBBPBB at 30 frames per second with the resolution of 720 x 480 pixels. The ground truth of the testing videos is shown in Table 2-1. Taking tennis sports videos as demonstration, two video sequences are selected from Australia Open and US Open, respectively. The length of the first video sequence is 50 minutes and the number of the shots of tennis court view is 300, which are extracted using the proposed approach of scene identification. The length of the second video sequence is 28 minutes and it contains 146 shots of tennis court view. The playing styles of the players in these two sequences are different, in which a classic serve-and-volleyer is present in the first sequence and two players in the second one are baseliners. The experimental results of tennis event detection are detailed in the following.

The event inference model shown in Fig. 2-12 is used to infer three events – “baseline rallies”, “serve and volley” and “passing shot” from the results of objects tracking algorithm. Fig. 2-15 shows the interface of the tennis event detection system. The system shows the coordinates of the trajectory of two players and the result of events inference in the “result” field after choosing the video clip in the “Scene ID” field. From the fields “TP” and “BP” in the system interface, we can see the coordinates of two players, top player (TP) and bottom player (BP) in the scene. Here the value 9 in TP field and the value 14 in BP field are the number of detected top player and the number of detected bottom player in consecutive P-frames within a shot, respectively.

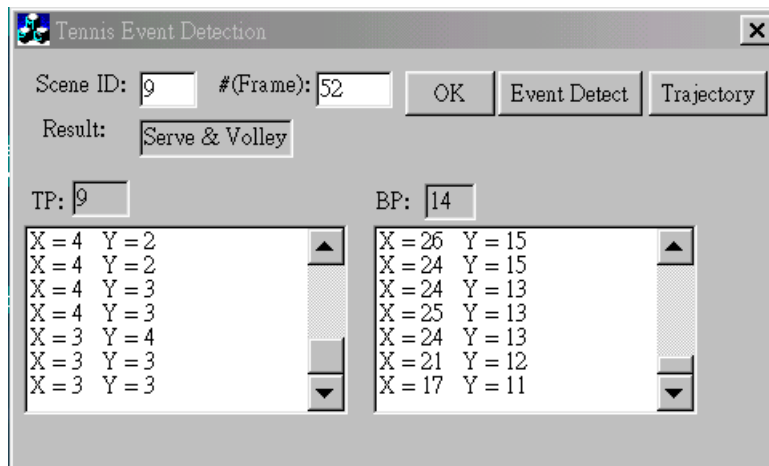


Fig. 2-15. The system interface shows an example of tennis event detection

Examples of the experimental results of trajectories detection are shown from Fig.2-16 to Fig.2-18 and the last frame of each video clip is also displayed to represent the scene. In Fig. 2-16, we can see that the trajectories of two players are both near the baseline and the event is recognized as baseline rallies. Fig. 2-17 shows an example event of “serve and volley”. The bottom player is the server and the final position of the trajectory is very close to the net. From the trajectory, the event can be classified as the event of “serve and volley”. As for the event of “passing shot”, we can see an example in Fig. 2-18. The bottom player moves from the baseline and ends near the service line. Accordingly, the event is identified as “passing shot”.

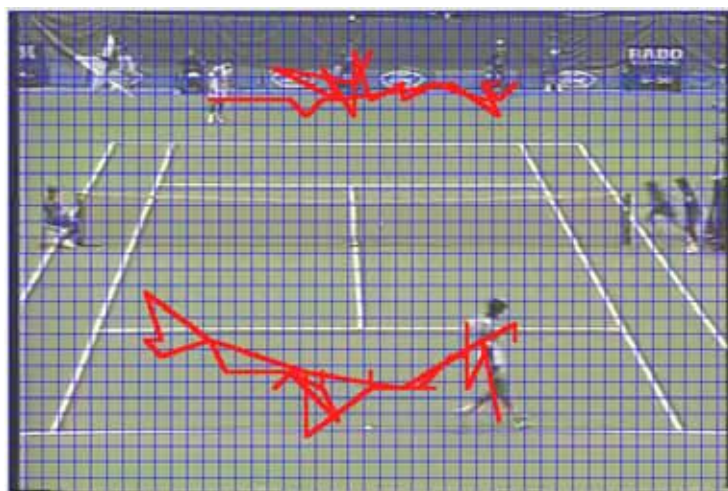


Fig. 2-16. An example of baseline rally event

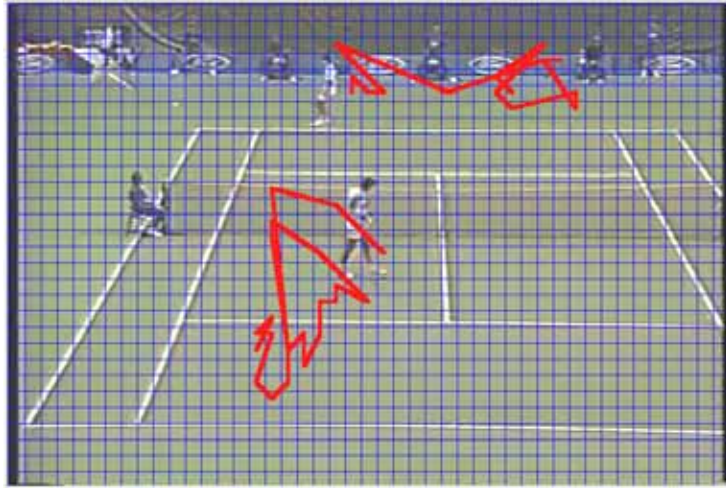


Fig. 2-17. An example of serve and volley event

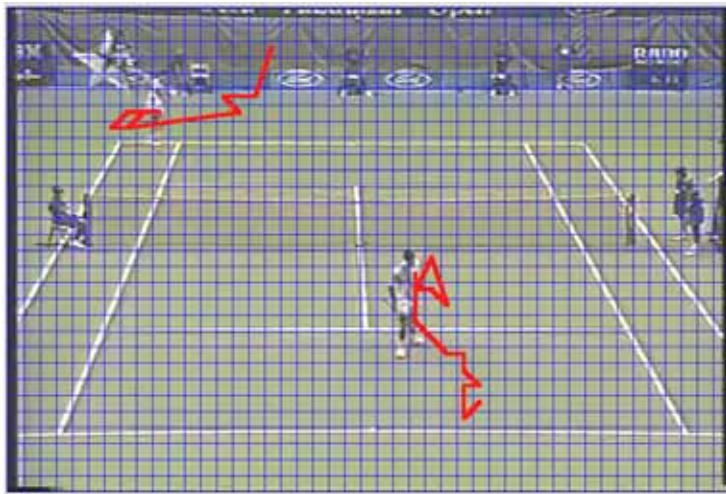


Fig. 2-18. An example of passing shot event

In the experiment, the numbers of effective tennis court clips in these two videos are 230 and 94, respectively while other clips are the shots in which the ball service is not successful or the ball boy runs to pick up the ball. The criterion used for judging whether tennis court clip belongs to the type of ball boy running is that there is a trajectory appearing along a curve near the middle location of y-coordinate or a trajectory across the court in a clip. Examples of detecting the trajectory of a ball boy are shown in Fig. 2-19 and Fig. 2-20. Fig. 2-19 and Fig. 2-20 respectively show the starting and ending frames of a ball boy running clip in which the ball boy is marked by an ellipse. We can observe that the trajectory lies near the center position of each

frame within the clip and hence this clip is classified as the type of ball boy running. This type of tennis court clips is recognized as insignificant and is filtered out thereafter.

Table 2-1. Ground truth of the testing video

Video Sequences	Length	Number of Tennis Court Shots	Number of Competition Shots
Video 1	50 minutes	300	230
Video 2	28 minutes	146	94

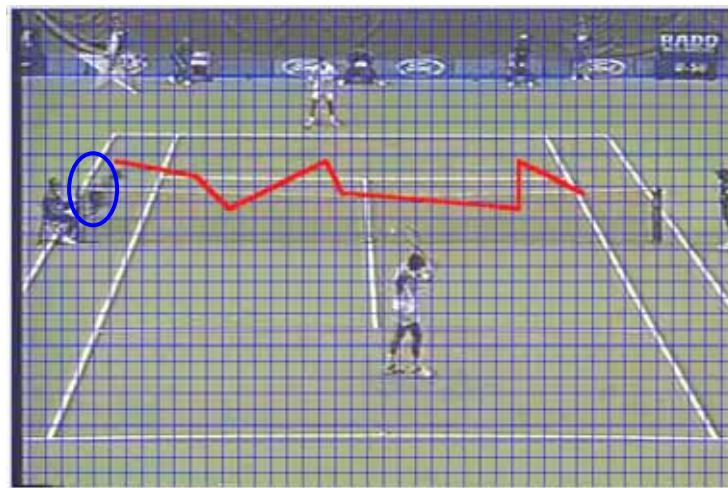


Fig. 2-19. Start-frame of a ball boy running clip

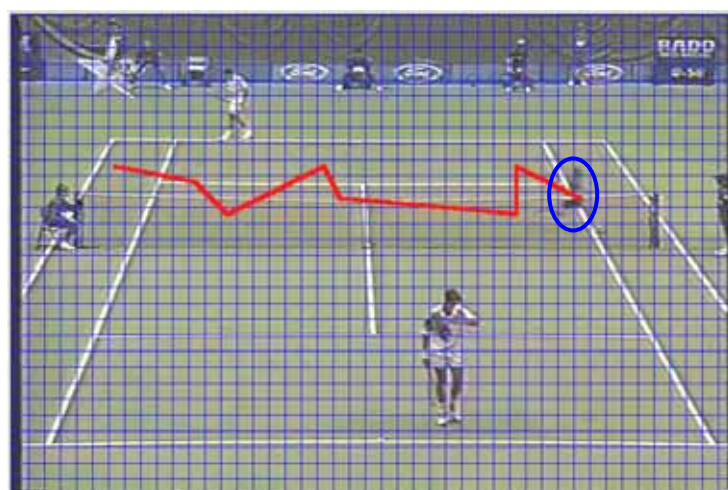


Fig. 2-20. End-frame of a ball boy running clip

The performance metrics used in the experiments are precision and recall, which

are collectively used to measure the effectiveness of a retrieval system. Eq. (2-8) shows the definition of precision and recall, where “*Retrieve(q)*” means the retrieved video sequences corresponding to a query sequence q , “*Relevant(q)*” denotes all the video sequences in the database that are relevant to a query sequence q and $\|\cdot\|$ indicates the cardinality of the set. Recall is defined as the ratio between the number of retrieved relevant video sequences and the total number of relevant video sequences in the video database, and precision is defined as the ratio between the number of retrieved relevant video sequences and the number of total retrieved video sequences.

$$Recall = \frac{\|Retrieve(q) \cap Relevant(q)\|}{\|Relevant(q)\|} \quad (2-8)$$

$$Precision = \frac{\|Retrieve(q) \cap Relevant(q)\|}{\|Retrieve(q)\|}$$

In the first video, the events of 213 clips are correctly detected and identified among the significant 230 clips of tennis court and 17 clips are falsely detected. Therefore, the average recall is about 90% and the average precision is about 89%. In the second video, the events of 85 clips are correctly recognized among 94 clips of tennis court. The average recall is about 89% and the average precision is about 86%. Table 2-2 shows the details of the number of correct detection, false detection, miss detection, the recall and the precision of event inference. The overall performance of precision and recall in the testing video sequences is 87% and 90%, respectively. The precision of detecting the event of baseline rally in these two videos is 95% and the recall is 94%. The inference result of baseline rally is better than the other two events since we assume that the net is located in the middle position in the event inference model. In addition, the camera may pan or tilt and moreover the size and shape of

players would change abruptly while the players approach the net. The factor of abrupt change of object size and shape would lead to the situation that the blocks match the wrong position in the reference frame. In this case, the position of the players may not be correctly located, even though the global motion has been estimated and compensated before the process of object detection.

Table 2-2. Experimental Results of Tennis Event Inference

Results Scene Type	Actual Number of Clips	Correct Detection	Miss Detection	False Detection	Precision	Recall
Baseline Rally (BR)	91	87	4	5	95%	96%
	65	60	5	3	95%	92%
Serve and Volley (SV)	115	106	9	6	95%	92%
	19	16	3	4	80%	84%
Passing Shot (PS)	24	20	4	6	77%	83%
	10	9	1	2	82%	90%
Average					87%	90%

The precision and recall of passing shot detected though are satisfactory but not so high as expected. The number of miss detection of passing shot in the first video is 4 in which 2 clips are classified as baseline rally event and other 2 clips are regarded as serve and volley. However, the occurrence frequency of passing shots is relative low in a tennis game, especially in the games of baseline-rally style such as the second testing video. Usually, the activity area of passing shot event is confined to the area around service line. Since the position of the service line is pre-assumed, while the camera moves too dramatically, it is difficult to keep track of the service line area. After serving the ball, the server may still stays near the baseline while the ball is subsequently hit and passed back very fast by the receiver. In this case, the event would be regarded as baseline rally event.

2.7 Summary

In this chapter, we propose an object-based video content parsing and event

understanding technique in MPEG compressed videos to support semantic content indexing and abstraction. GOP-based video segmentation is exploited to efficiently segment video streams into shots in compressed domain. The shots of tennis court are recognized and selected by the module of scene identification. The insignificant shots of tennis court like the fault service, or so called ball boy running, are removed based on the detected trajectory of the ball boy. The proposed object-tracking algorithm employing the information of motion vectors is utilized to locate the position of moving objects in consecutive P-frames and to generate trajectory of the objects with prominent movement. Furthermore, video events can be inferred from the generated trajectories based on the inference model with specific domain knowledge. Experimental results are convincing and verify that the proposed approach can effectively detect events of tennis games and generate the description of tennis videos automatically. Therefore, by utilizing the proposed mechanism and applying domain knowledge, video streams can be automatically parsed and annotated, and thus the associated metadata of the inferred high-level semantic clips can be used to automatically structure videos, summarize videos and generate the description scheme (DS) and descriptor (D) of video content for MPEG-7 standard.

The proposed mechanism also provides several reusable modules. For example, the module of scene identification can be used to recognize the shots of full or partial view of athletic field of football, soccer, baseball and volleyball when the corresponding domain knowledge is employed. While these kinds of shots are acquired, sports events can be inferred from the spatial-temporal relationship of objects or some active regions in consecutive frames. For example, in baseball games, the striker scene, which is normally composed of a catcher in the middle of the scene, a striker in the left or right and an umpire in the top, can be identified according to the distribution of these objects. Based on the proposed mechanism, we have successfully

applied some of the proposed modules to semantic indexing of volleyball games [22], in which the major events of “service”, “full-court competition” and “close-up” are recognized. In addition, the multi-object tracking algorithm can also be used in the localization and tracking of text captions [23] and in surveillance system to detect suspicious circumstances and the alarm can be triggered according to the detected events.

In the future, we will develop some global edge detection approach to detect the boundary of tennis court in MPEG videos for improving the accuracy of event detections. We will extend the approach of motion-based semantic event detection to other kinds of sports video to extract semantically meaningful video events. Concurrently, the description schemes and descriptors generations for effective content-based query are also the future research.



Chapter 3. Automatic Closed Caption Detection and Filtering in MPEG Videos for Video Structuring

3.1 Introduction

With the increasing digital videos in education, entertainment and other multimedia applications, there is an urgent demand for tools that allow an efficient way for users to acquire desired video data. Content-based searching, browsing and retrieval is more natural, friendly and semantically meaningful to users. The need of content-based multimedia retrieval motivates the research of feature extractions of the information embedded in text, image, audio and video. With the technique of video compression getting mature, lots of videos are being stored in compressed form and accordingly more and more researches focus on the feature extractions in compressed videos especially in MPEG format. For instances, edge features are extracted directly from MPEG compressed videos to detect scene change [24] and captions are processed and inserted into compressed video frames [25]. Features, like chrominance, shape and texture are directly extracted from MPEG videos to detect face regions [26-27]. Videos in compressed form are analyzed and parsed for supporting video browsing [28].

However, textual information is semantically more meaningful and attracts increasing researches on closed caption detection in video frames [29-37]. The researches [31-34] detect closed captions in pixel domain. In [36-37], they proposed to detect closed captions in specific areas. However, it is impractical to localize closed captions in specific areas of a frame since in different video sources closed captions normally do not appear in a fixed position.

A number of previous researches extract closed captions from still images and video frames [33-35][38-39] with a constraint that characters are bounded in size.

Besides, these approaches usually require the property that text has a good contrast from the background. However, text region localization with size constraint is not practical especially for the cases that those captions are small in size but are very significant and meaningful. For example, in sports videos, the superimposed scoreboards show the intermediate results between competitors and present the match as clearly as possible without interference.

There has been very little effort to extract features in compressed domain to detect closed captions in videos. Zhong et al. [29] and Zhang and Chua [30] detect large closed captions frame-by-frame in MPEG videos using DCT AC coefficients to obtain texture information in I-frames without exploiting the temporal information in consecutive frames. However, it is impractical and inefficient to detect closed captions in each frame. Due to the temporal nature of long-term consistency of closed captions over continuous video frames, it would be more robust to detect the closed caption based on its spatial-temporal consistency. Gargi et al. [35] perform text detection by counting the number of intra-coded blocks in P and B frames based on the assumption that the background is static. Hence, it is vulnerable to abrupt and significant camera motion. Besides, this approach is only applied to the P and B frames and does not handle captions that appear in the I-frames.

In this chapter, in order to detect closed captions efficiently and flexibly, we propose an approach for compressed videos to detect caption frames in meaningful shots. Then caption frames instead of every frame are selected as targets for localizing closed captions without size constraint while considering long-term consistency of closed captions over continuous caption frames for removing noise. Moreover, we propose a novel tool – font size detector to identify font size in compressed videos. Using this tool, after the targeted font size is indicated, we can allow users to automatically discriminate captions of interest instead of captions in the presumed

position. It is worth noticing that font size recognition is a critical step in the process of video OCR since a bottleneck for recognizing characters is due to the variation of text font and size [40-43]. Therefore, this tool can be used as a pre-filter to quickly signal the potential caption text and thus reduce the amount of data that needs to be processed.

The proposed system architecture is shown in Fig. 3-1. All the tasks are accomplished in compressed domain. GOP-based video segmentation [16] is exploited to efficiently segment video into shots. The color-based shot identification is proposed to automatically identify meaningful shots. Caption frames in these shots are detected by computing the variation of DCT AC energy both in the horizontal and vertical directions. In addition, we detect closed captions using the weighted horizontal-vertical DCT AC coefficients. To detect closed captions robustly, each candidate closed caption is verified further by computing its long-term consistency that is estimated over the backward shot, the forward shot and the shot itself. After closed captions are obtained, we differentiate the font size of each closed caption based on horizontal projection profile of DCT AC energy in the vertical direction. Captions of interest can then be identified by the font size and size variance. Finally, captions of interest and the meaningful shots can be employed together to construct a high-level concise table of video content.

The rest of the chapter is organized as follows. Section 3-2 describes the color-based shot identification. Section 3-3 presents the proposed approach of closed caption localization. Section 3-4 shows the experimental results and the prototype system of video content visualization. The conclusion and future works are given in section 3-5.

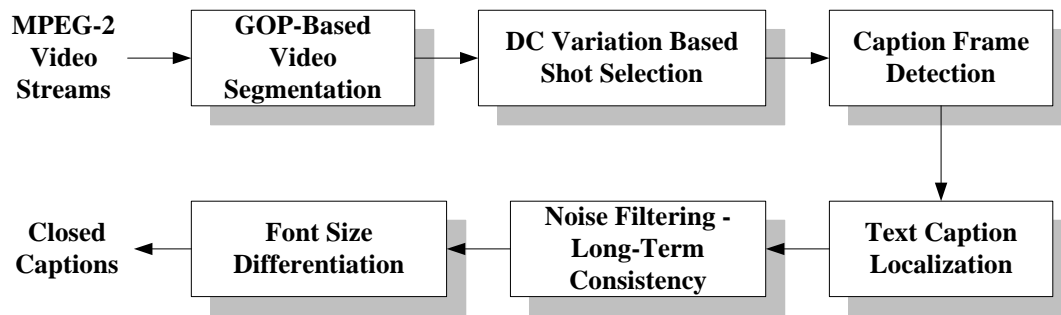


Fig.3-1. Overview of the system architecture

3.2 Shot Identification

3.2.1 Video Segmentation

Video data is segmented into clips to serve as logical units called “shots” or “scenes”. In MPEG-2 format, GOP layer is a random accessed point and contains GOP header and a series of encoded pictures including I, P and B-frame. The size of a GOP is about 10 to 20 frames, which is less than the minimum duration of two consecutive scene changes (about 20 frames). Instead of checking frame-by-frame, we first detect possible occurrences of scene change GOP-by-GOP (inter-GOP). The difference between each consecutive GOP-pair is computed by comparing the corresponding I-frames. If the difference of DC coefficients between these two I-frames is larger than the threshold, then there might exist scene change in between these two GOPs. Hence, the GOP that might contain the scene change frames is located. In the second step – intra GOP scene change detection, we further use the ratio of forward and backward motion vectors to find out the actual frame of scene change within a GOP. By this approach, the experimental results are encouraging and prove that the scene change detection is efficient for video segmentation.

3.2.2 Shot Identification

While the boundary of each shot is detected, the video sequence is segmented into shots consisting of the advertisement, close-up and court-view. Closed captions can then be detected in each video shot. However, it is impractical to detect closed

captions in all video shots. In sports videos, the shots of court-view are our focus since the matches of the sports are primarily shown in the shots of court-view and the scoreboards are presented mostly in these kinds of shots. Therefore, scene identification approach is proposed to identify the shots of court-view.

To recognize the shots of court-view, it is worth noticing that the variation of the intensity in the court-view frames is very small through a whole clip and the value of intensity variation between consecutive frames is very similar. In contrast, the intensity of the advertisement and close-up varies significantly in each frame and the difference of the variance of intensity between two neighboring frames is relatively large. Therefore, the intensity variation within a video shot can be exploited to identify the shots of court view. In order to efficiently obtain the intensity variance of each frames and that of a video shot, DC-images of I-frames are extracted to compute the intensity variance. The frame variance $FVar_{s,i}^{DC}$ and the shot variance $SVar_s$ are defined by

$$FVar_{s,i}^{DC} = \sum_{j=1}^N DC_{i,j}^2 / N - \left(\sum_{j=1}^N DC_{i,j} / N \right)^2, \quad (3-1)$$

and

$$SVar_s = \sum_{i=1}^M (FVar_{s,i}^{DC})^2 / M - \left(\sum_{i=1}^M FVar_{s,i}^{DC} / M \right)^2, \quad (3-2)$$

where $DC_{i,j}$ denotes the DC coefficient of the j th block in the i th frame, N represents the total number of blocks in a frame, and M denotes the total number of frames in shot s .

Based on the fact that the intensity variance of a court-view frame is very small through a whole clip, shots are regarded as the type of court-view $Shot_{Court}$ by

$$Shot_{Court} = \left\{ Shot_s \mid FVar_{s,i}^{DC} < \delta_{frame} \text{ and } SVar_s < \delta_{shot}, \forall i \in [1, N] \right\} \quad (3-3)$$

where δ_{frame} and δ_{shot} are the predefined thresholds.

In order to demonstrate the applicability of the proposed shot identification, the variation of the intensity variance of each I-frame in sports videos including tennis, football and baseball is exhibited in Fig. 3-2.

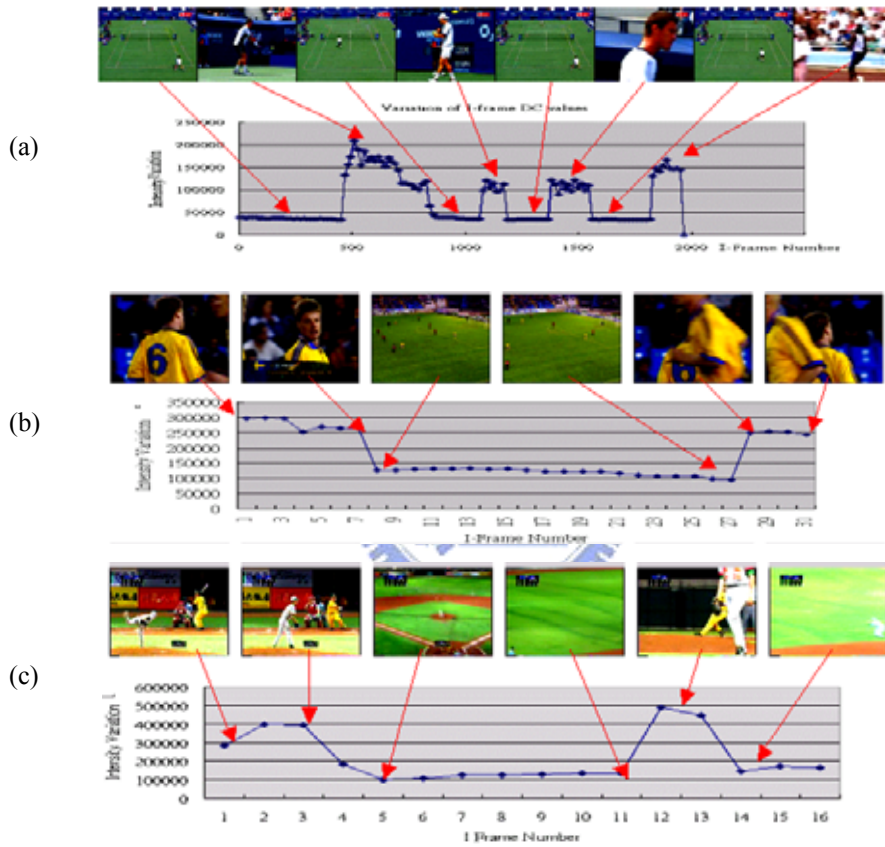


Fig. 3-2. Variation of I-frame DC value (a) tennis; (b) football; (c) baseball

Fig. 3-2(a) shows a tennis video composed of four tennis court shots, three close-up shots and a commercial shot. Fig. 3-2(b) introduces a football sequence consisting of close-up shots and football field shots. A baseball sequence is presented in Fig. 3-2(c) including pitching shots, baseball field shots and close-up shots. From Fig. 3-2, we can observe that the intensity variance of the type of court-view is very small and the value is very similar through a whole clip. Thus, the clips of court-view can be

indicated and selected by the characteristic that the value of intensity variance $FVar_{s,i}^{DC}$ is small in each individual frame and is consistent over the whole shot. Therefore, the proposed approach of shot identification can be applied to identify court-view shots of sports videos, in which the view of a match consists of the intensity-consistent background of a court or athletic field.

3.3 Closed Caption Localization

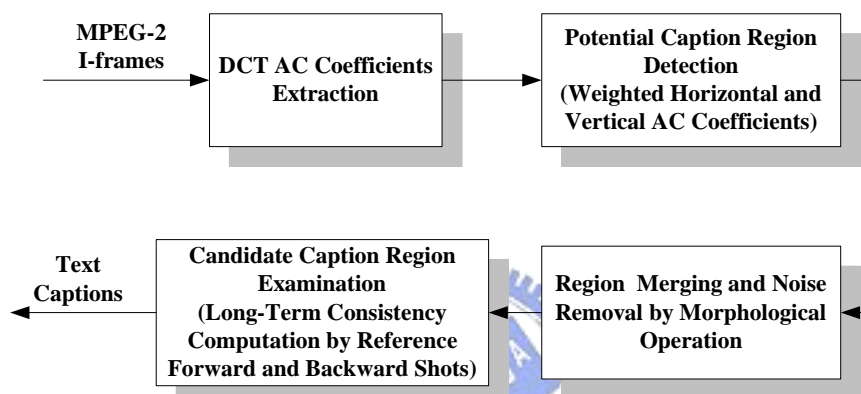


Fig. 3-3. The approach of closed caption localization

In this section, we shall elaborate how to detect caption frames and how to detect closed captions in caption frames. After shots of interest are identified, the closed captions, like the scoreboard, are then detected in these significant shots. However, in general, the scoreboard does not always appear in the frames continuously through a whole clip. It comes up for some while, disappears and comes up again. Therefore, to avoid the time-consuming overhead of closed caption examination frame-by-frame, caption frames should be detected first. The proposed closed caption localization is shown in Fig. 3-3. In the process of caption detection, DCT AC coefficients of I-frames in MPEG-2 video are extracted and are used to determine the energy variation of 8x8 blocks in horizontal and vertical directions, respectively. Potential caption regions are indicated using the weighted horizontal-vertical AC coefficients

and then the fragmented regions are merged using the morphological operations. For more robust localization of closed captions, the spatial-temporal relationship over consecutive frames is exploited to compute the long-term consistency of each candidate caption region by referring certain I-frames in forward and backward shots. Localized closed caption regions may contain the scoreboard, or the logo of a certain channel or some billboard. However, the scoreboard is what viewers are most interested in. Therefore, based on the observation that these different types of closed captions are generally different in font size, we propose an approach to discriminate font size among localized captions. The details of caption frame detection are described in section 3.3.1 and the approach of closed caption localization is shown in section 3.3.2. Section 3.3.3 presents the approach of font size differentiation.

3.3.1 Caption Frame Detection

Caption frame detection is an essential step for closed caption localization because captions may disappear in some frames and then appear subsequently. Therefore, to avoid detecting closed captions frame-by-frame, we first identify the possible frames in which captions might be present. However, the caption size of closed captions in the shots of court-view is usually very small. Under this circumstance, the change of the AC energy of the entire frame with the appearance or disappearance of the small caption would not result in significant variation. It means that the variance of the AC energy obtained from an entire frame cannot be used as a measurement of the possibility of the presence of a small caption.

In order to robustly detect closed captions without size constraint, each I-frame is divided into an appropriate number of regions (say R). However, the size of a region should be moderate to reflect the actual variation of appearance or disappearance of small captions. If the size of a divided region were too small, any slight change of color or texture would incur quite prominent variation of AC energy. Accordingly, in

order to detect the appearance of super-imposed closed captions in four corner areas as well as in the middle of a frame, the number of regions R here can be set to six and its division method is shown in Fig. 4. Based on the frame division method, the variance $RVar_{s,i}^r$ of AC coefficients of each region r in the i th frame of shot s is computed by

$$RVar_{s,i}^r = \sum_{j=1}^N \sum_{h/v} AC_{h/v,j}^2 / N - (\sum_{j=1}^N \sum_{h/v} AC_{h/v,j} / N)^2, r = 1, 2, \dots, R \quad (3-4)$$

where $AC_{h/v,j}$ denotes the horizontal AC coefficients from $AC_{0,1}$ to $AC_{0,7}$ and the vertical AC coefficients from $AC_{1,0}$ to $AC_{7,0}$ in region r and N is the total number of blocks in region r . The DCT AC coefficients used are shown in Fig. 3-5.

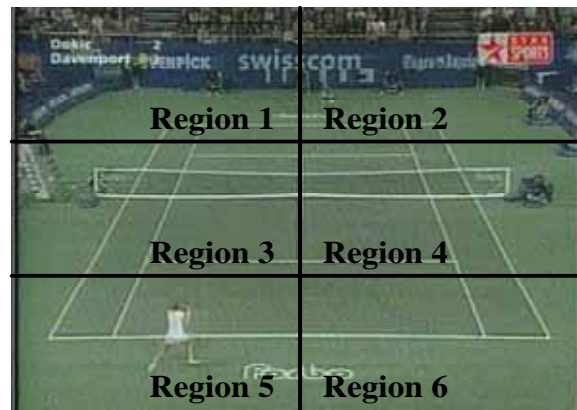


Fig. 3-4. An original frame is divided into R regions (e.g. $R = 6$)

DC	$AC_{0,1}$	$AC_{0,2}$	$AC_{0,3}$	$AC_{0,4}$	$AC_{0,5}$	$AC_{0,6}$	$AC_{0,7}$
$AC_{1,0}$							
$AC_{2,0}$							
$AC_{3,0}$							
$AC_{4,0}$							
$AC_{5,0}$							
$AC_{6,0}$							
$AC_{7,0}$							

Fig. 3-5. DCT AC coefficients used in text caption detection

Using the energy variance $RVar_{s,i}^r$ of each region, the method proposed to determine caption frames is illustrated as follows:

For each region r , (3-5)

If $Diff(RVar_{s,i+1}^r, RVar_{s,i}^r) \leq -\delta$, captions may disappear in frame $(i+1)$

If $Diff(RVar_{s,i+1}^r, RVar_{s,i}^r) \geq \delta$, captions may appear in frame $(i+1)$

where $Diff(RVar_{s,i+1}^r, RVar_{s,i}^r) = RVar_{s,i+1}^r - RVar_{s,i}^r$. In the method, $RVar_{s,i+1}^r$ of region- r in frame $i+1$ is compared with $RVar_{s,i}^r$ of region- r of frame i . If the difference between $RVar_{s,i+1}^r$ and $RVar_{s,i}^r$ is larger than a threshold δ (3000), it means the texture of region- r in frame $i+1$ is more complex than that of region- r in frame i , i.e., closed captions may be superimposed in frame $i+1$. Similarly, if the difference between $RVar_{s,i+1}^r$ and $RVar_{s,i}^r$ is smaller than the threshold $-\delta$, the texture of region- r in frame $i+1$ becomes less complex than that of region- r in frame i , i.e., closed captions in frame i may disappear in frame $i+1$.

Examples of caption frame detection are demonstrated in Fig. 3-6. Fig. 3-6(a) shows the detection of caption frames with small closed captions presented. We can see that the curve of DCT AC variance $RVar_{s,i}^r$ of region-1 drops abruptly in the 18th I-frame and rises in 39th I-frame since the scoreboard disappears from the 18th I-frame to the 38th I-frame in the area of region-1 and then appears again in the 39th I-frame. Similarly, detection of caption frames with a large closed caption presented is demonstrated in Fig. 3-6(b). The text region covering both region-5 and region-6 appears in the 47th I-frame and is presented through the 59th I-frame and disappears in the 60th I-frame. Although the variance of AC energy in region-5 is larger than that in region-6 due to the text of the scoreboard presented in the left side, the variance of AC energy of both regions conforms to Eq. (3-5). Therefore, video frames from the 47th to the 59th I-frames are indicated as caption frames and can be selected for closed caption localization.

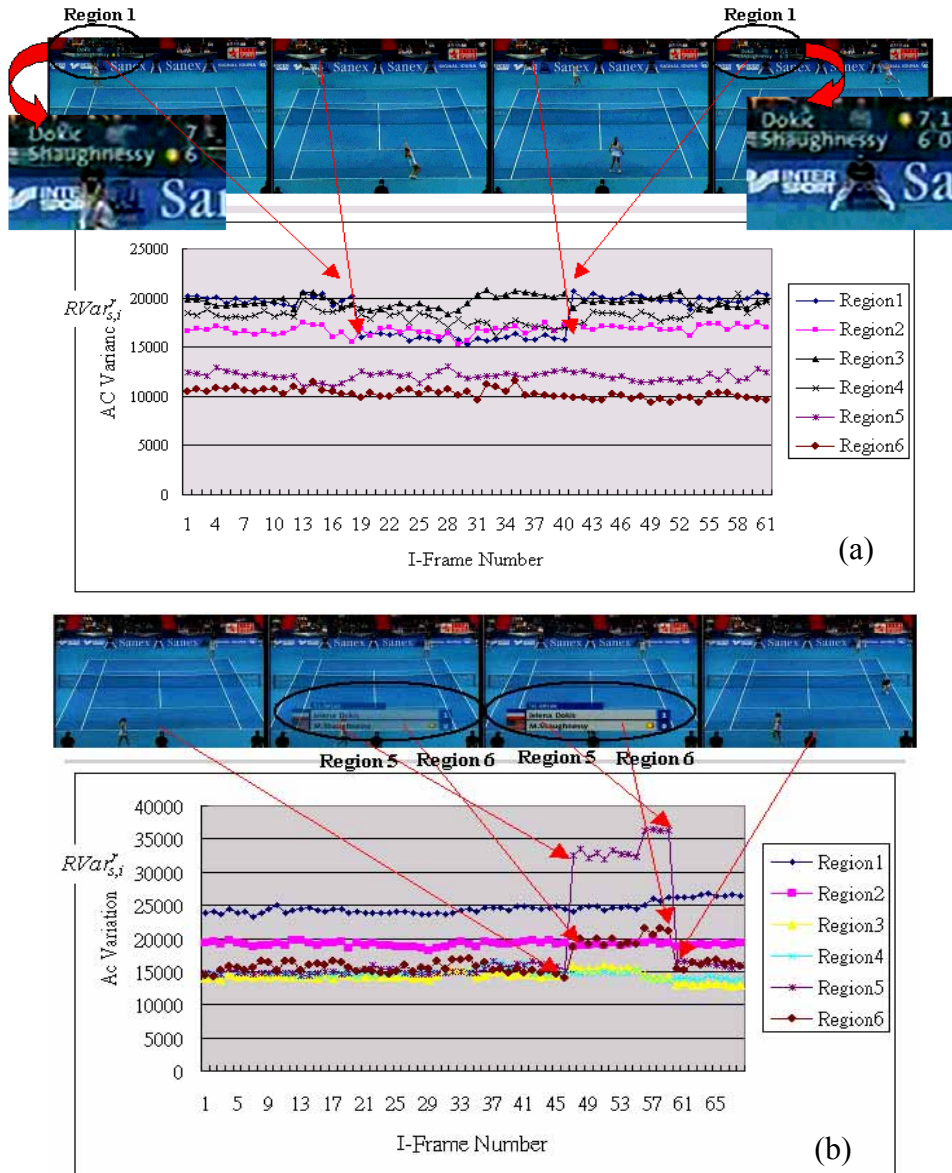


Fig. 3-6. Demonstration of caption frame detection: (a) small closed caption (b) large closed caption

3.3.2 Closed Caption Localization

While the caption frames are identified, we then locate the potential caption regions in these frames by utilizing the gradient energy obtained from the horizontal and vertical DCT AC coefficients. We can observe the fact that closed captions generally appear in rectangular form and the AC energy in the horizontal direction would be larger than that in the vertical direction since distance between characters is fairly small and the distance between two rows of text is relatively large. Therefore, we

assign higher weight to horizontal coefficients than that to vertical coefficients. The weighted gradient energy of an 8x8 block E used as a measurement for evaluating the possibility of a text block can be defined as follows:

$$E = \sqrt{(w_h E_h)^2 + (w_v E_v)^2} \quad (3-6)$$

$$E_h = \sum_{h1 \leq h \leq h2} |AC_{0,h}|, \quad h1 = 1, h2 = 7$$

$$E_v = \sum_{v1 \leq v \leq v2} |AC_{v,0}|, \quad v1 = 1, v2 = 7$$

If the energy E of a block is larger than a predefined threshold, this block is regarded as a potential text block. Otherwise, the block would be considered as a non-text block and be filtered out without further processing. Besides, in order to save computation cost, we select only 3 I-frames (first, middle and last) as representative frames in a shot for closed caption localization.

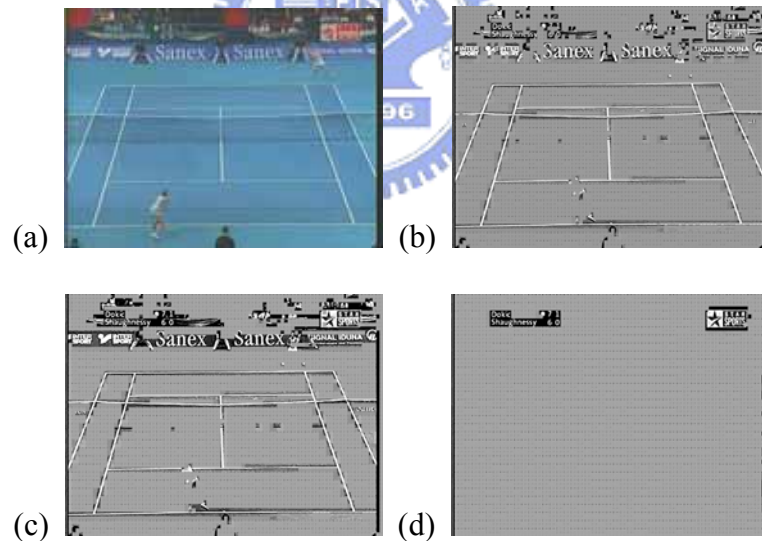


Fig. 3-7. Illustration of intermediate results of closed caption localization (a) Original frame (b) Closed caption detection (c) Result after applying morphological operation (d) Result after long-term consistency verification

The result of closed caption localization is demonstrated in Fig.3-7 with w_h set to 0.7 and w_v to 0.3. Although the scoreboard and the trademark in Fig. 3-7(b) in the upper part of the frame are all located and indicated, caption regions are fragmentary

and some noisy regions remains. Therefore, we adopt a morphological operator in the size of 1x5 blocks to merge fragmentary text regions and the result is demonstrated in Fig.3-6(c). Afterward, the merged text regions are further verified by computing the long-term consistency. For long-term consistency checking, we select another two I-frames as temporal reference, the last I-frame of the forward shot (P_F) and the first I-frame of the backward shot (F_B) as depicted in Fig. 3-8, where T_f , T_m and T_r are the first, middle and the last I-frames of the specific shot. One possible measurement of the long-term coherence of text blocks in potential regions is to check if the text blocks of a potential caption region appear more than half of the time in a shot. That is text blocks appear in more than or equal to three times among the five representative five I-frames.

Here, we exploit the position, intensity and texture information of potential text blocks among these representative I-frames (P_F, T_f, T_m, T_r and F_B) to measure the temporal coherence as defined by

$$C = \frac{\sum_{k=1}^2 (DC_{B_k} - \overline{DC})(E_{B_k} - \overline{E})}{\sqrt{\sum_{k=1}^2 (DC_{B_k} - \overline{DC})^2} \sqrt{\sum_{k=1}^2 (E_{B_k} - \overline{E})^2}}, \quad -1 \leq C \leq 1 \quad (3-7)$$

where DC_{B_k} denotes the value of DC coefficient of B_k , \overline{DC} is the average of DC_{B_k} and $DC_{B_{k+1}}$, E_{B_k} represents the weighted gradient energy E of B_k as defined in Eq. (3-6) and \overline{E} is the average of E_{B_k} and $E_{B_{k+1}}$. A block is characterized by its intensity represented by the DC coefficient and also by its texture obtained from AC coefficients. We compute the correlation C to measure the similarity between two blocks B_k and B_{k+1} , which are in the same corresponding position in their respective frame i and frame $i+1$. If a value C of a block pair is larger than δ_C , these two blocks are regarded as the same. To estimate the temporal coherence of potential text blocks,

we need to compute the pair wise correlation C four times among the 5 representative I-frames as depicted by the arrow-lines in Fig. 3-8. Therefore, a text block is long-term consistent in the specific video shot only when more than half of the times the pair wise I-frames correlation C is larger than δ_c .

The result of long-term consistency checking of text blocks is demonstrated in Fig. 3-7(d). We can see that the scoreboard and the trademark are all successfully localized and most of the noise is removed. The proposed closed caption localization can also be applied to other kinds of videos such as baseball, news and volleyball as demonstrated in Fig. 3-9. In Fig. 3-8(a), we can observe that the closed caption primarily composed of Chinese characters is also localized correctly.

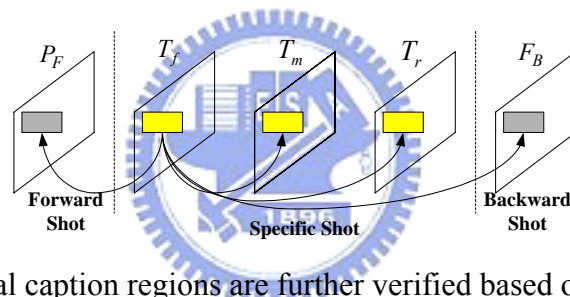


Fig. 3-8. Potential caption regions are further verified based on the long-term consistency

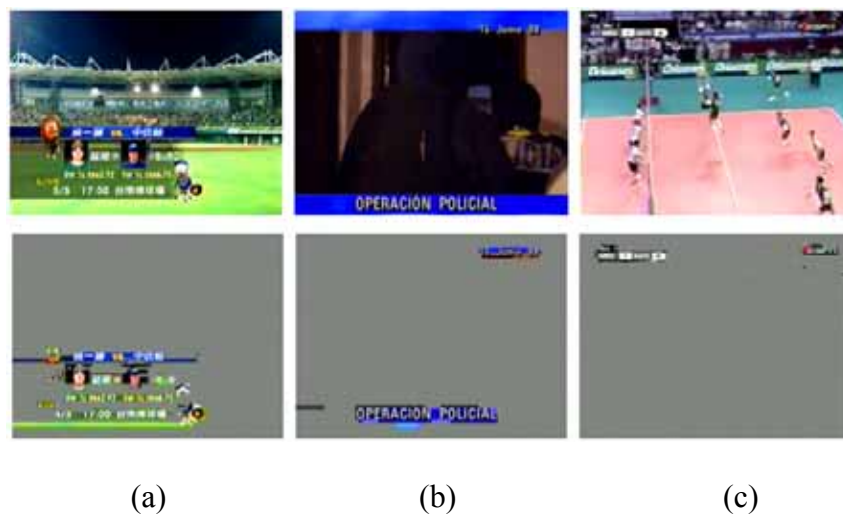


Fig. 3-9. Examples of closed caption localization (a) baseball; (b) news; (c) volleyball

3.3.3 Font Size Differentiation

From Fig. 3-7(d), we can notice that the scoreboard in the left upper corner and the trademark in the right upper corner are all successfully detected. Since scoreboards can be used for the content structuring of sports videos, the issue of separating out the captions in the scoreboard is one of our concerns. Hence, the tool – font size detector is proposed to automatically discriminate the font size as a support in the discrimination of scoreboards. To detect the font size, the gradient energy of each text block is exploited. Since a block consisting of characters will have much larger gradient energy than that of a block consisting of blank space, the distance between two character blocks can thus be determined by evaluating the distance between peak gradient values among blocks in a row or column. It means that the font size can be evaluated by measuring the distance between blocks with peak gradient value (i.e., the periodicity of peak values). The gradient energy in the vertical direction instead of horizontal direction is exploited since the blank space in between two text rows is generally larger than that between two letters and hence the variation of gradient energy in the vertical direction would present in more regular pattern.

In addition, to obtain robust periodicity, we compute the DCT coefficients of the 8x8 overlap-block between two neighboring blocks as defined in Eq. (3-8). A overlap-block $B_{overlap-block}$ shown in Fig. 3-10 comprises lower portion of the top neighboring 8x8 block B_t and upper portion of the bottom neighboring block B_b , where I_{w0} and I_{w1} are the identity matrix in the dimension of $w0 \times w0$ and $w1 \times w1$, respectively. More robust results would be achieved if more overlap-blocks are computed and exploited. For example, $w0$ and $w1$ can be respectively set to 1 and 7, 2 and 6, 3 and 5, etc. to acquire more overlap-blocks for more accurate estimation of font size.

$$B_{\text{overlap-block}} = \begin{pmatrix} 0 & 0 \\ 0 & I_{w_0} \end{pmatrix} B_t + \begin{pmatrix} 0 & 0 \\ I_{w_1} & 0 \end{pmatrix} B_b \quad (3-8)$$

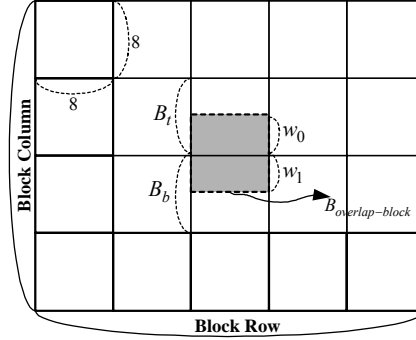


Fig. 3-10. Overlap-block is interpolated from its two neighboring blocks B_t and B_b

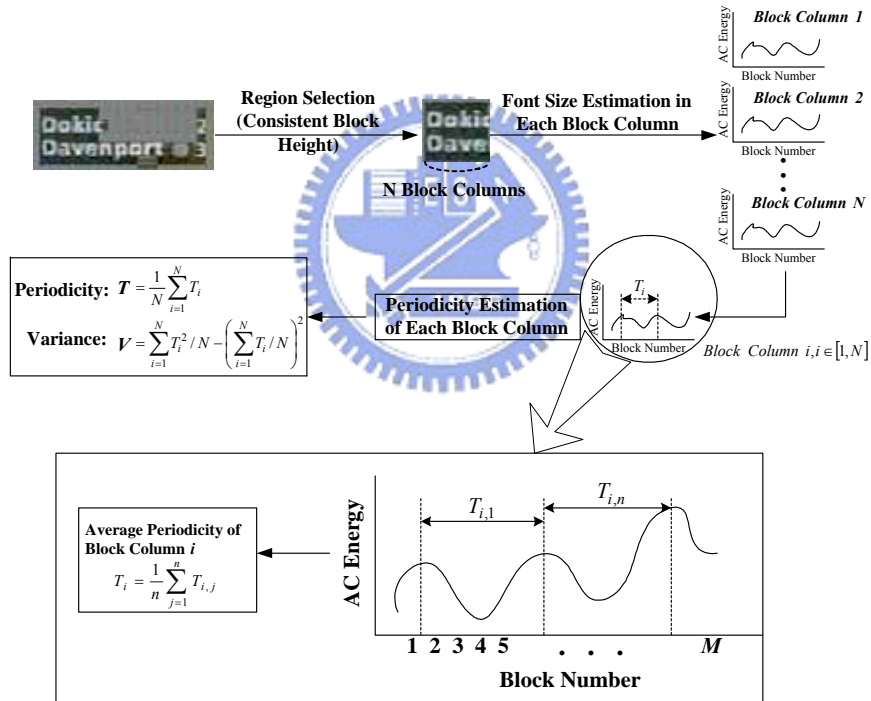


Fig. 3-11. The proposed approach of font size differentiation in compressed domain

Fig. 3-11 shows the proposed approach of font size differentiation, in which the periodicity and variance are estimated for each block column. However, localized closed captions like the example in the top of Fig. 3-11 may not be complete in shape because some pieces with low gradient energy are filtered out. Therefore, to achieve robust font size differentiation, a region that forms a rectangular in the localized

caption is determined for font size computation. Font size differentiation is performed on each block column in the selected region of the closed caption, where a block column depicted in Fig. 3-11 is defined as a whole column of blocks. While the AC energy of each block is extracted, the curve of the variation of AC energy for each block column is checked to locate each local maximum. We can observe that the region containing the boundary of closed captions would have conspicuous texture variation in the vertical direction and the value of the gradient energy would be relatively high. Therefore the local maximum of the curve of vertical AC gradient energy is regarded as the boundary of closed captions. While all local maximums are recognized, we must filter out noise and select reliable curve peaks for further verification. Due to the fact that the first and the last local maximums usually reflect the boundary of closed captions, hence we select the first and the last peaks of the curve and compute the average of the value of these two peaks as the threshold adaptively for noise filtering. If the value of a peak is smaller than the threshold, the peak is filtered out. Otherwise, the peak is kept for font size computation. Therefore, the periodicity of each block column T_i is computed by averaging the distance between two peaks of the curve of AC energy. Finally, the average periodicity T and the periodicity variance V of the closed caption are obtained by

$$T = \frac{1}{N} \sum_{i=1}^N T_i \quad (3-9)$$

and

$$V = \sum_{i=1}^N T_i^2 / N - \left(\sum_{i=1}^N T_i / N \right)^2 \quad (3-10)$$

where N is the total number of block columns in the selected area of the closed caption.

The results of font size analysis of the scoreboard and the trademark in Fig.3-12 are

demonstrated in Fig. 3-13 and Fig. 3-14. In the example, each column of the scoreboard and the trademark consists of 9 blocks, in which 5 blocks are original and 4 overlap-blocks are interpolated. For robustness of font size measurement, we should select some portion of localized closed caption, in which the height of each block column is consistent. Therefore, we compute T for first five block columns because the first part “Doki” of the localized scoreboard consists of five block columns of consistent height and several non-text blocks separate the second part of the scoreboard. Hence, in Fig. 3-13(b), the block columns of the trademark are all selected for font size computation because the height of each block column is consistent.

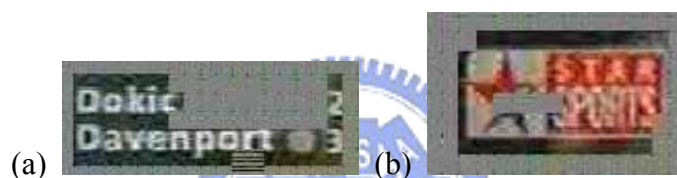


Fig. 3-12. The localized closed captions (a) scoreboard (b) trademark

From Fig. 3-13 and Fig. 3-14, we can see that the average distance T of the scoreboard is about 2.2 which is smaller than 2.9 of the trademark. Besides, the variance V of the row distance of blank space among block columns of the scoreboard is 0.05 which is also smaller than 0.8 of the trademark. Hence, we can correctly discriminate the scoreboard from trademark since the font size of the scoreboard is smaller than that of the trademark and the font size is of better regularity in the scoreboard than that in the trademark.

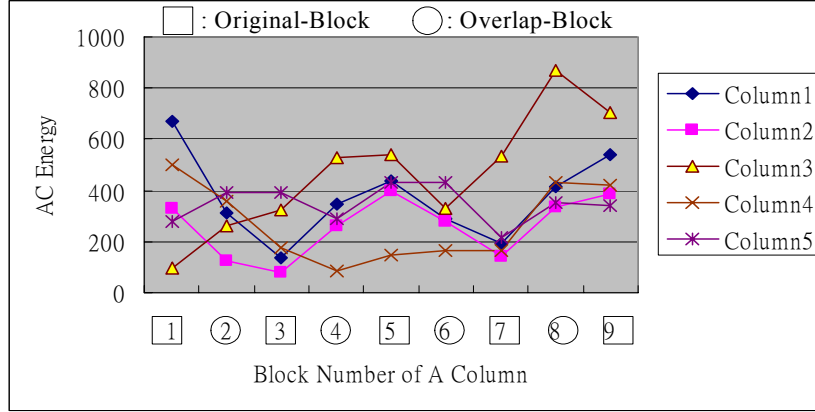


Fig. 3-13. Variation of AC energy of the scoreboard in Fig. 3-12(a) ($T=2.2$, $V=0.05$)

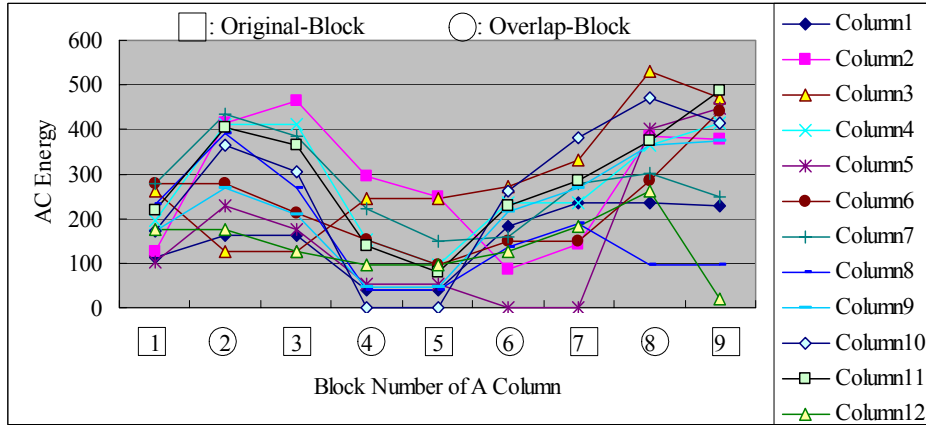


Fig. 3-14. Variation of AC energy of the trademark in Fig. 3-12(b) ($T=2.9$, $V=0.8$)

Furthermore, in order to estimate periodicity of font size more efficiently, we exploit the concept of the projection analysis of a print line [44-45]. Since it can serve for the detection of blank space between successive letters, we thus compute the horizontal projection profile P_H of each block row P_y by summing up the vertical AC coefficients of the blocks. P_H is defined as follows:

$$P_H = \left\{ P_y \mid P_y = \sum_{x=0}^{W-1} \sum_{v=1}^7 |AC_{v,0}|, 0 \leq y < H_T - 1, AC_{v,0} \in B_{x,y} \right\} \quad (3-11)$$

where H_T is the summation of the number of original blocks (H) and the number of

overlap-blocks ($H-l$) of a block column in an $H \times W$ caption region, and $B_{x,y}$ is a block of coordinate (x, y) . By this method, we compute the periodicity T of each localized closed caption once instead of inspection of the periodicity T and of the variance V in each block column. The horizontal projection profile of the scoreboard and the trademark is demonstrated in Fig. 3-15, where the average periodicity T of the scoreboard and the trademark is about 2 and 3, respectively. Using horizontal projection profile, font size can be detected more efficiently since one curve of AC energy variation needs to be computed for a closed caption.

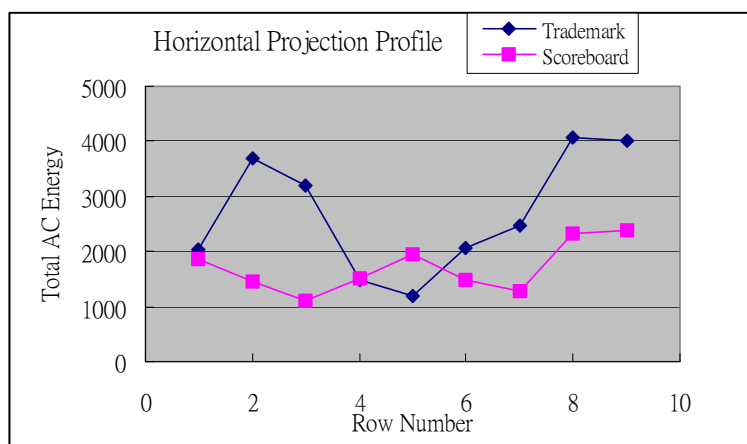


Fig. 3-15. Horizontal projection profile of DCT AC energy of the scoreboard and the trademark in Fig. 3-12(a) and Fig. 3-12(b), respectively

3.4 Experimental Results and Visualization System

3.4.1 Experimental Results

In the experiment, testing dataset consisted of four kinds of videos including tennis, baseball, volleyball and news. Two tennis videos selected from US Open and Australia Open, respectively were recorded from the Star-Sport TV channel. A volleyball video was recorded from ESPN TV channel and a baseball game was recorded from VL-Sport TV channel. A news video was selected from MPEG-7 testing dataset. The testing sequences were encoded in MPEG-2 format with the GOP structure IBBPBBPBBPBBPBB at 30 fps. The length of the first tennis video and the

news video was about 50 minutes, and the length of the second tennis video was about 30 minutes. The length of the volleyball video and the baseball video was about 40 minutes and 60minutes, respectively.

The ground truth of the number of caption I-frames of tennis videos shown in Table 1 was 903 and 414, respectively. In Table 2, there were totally 42183 text blocks in the representative frames of tennis video 1 and totally 25680 text blocks of tennis video 2. The number of text blocks in baseball was larger than other videos due to the large superimposed captions. The results of caption frame detection and closed caption localization were evaluated by estimating the precision and recall. The experimental result of caption frame detection was shown in Table 3-1, and the best performance was achieved in the first tennis video. In tennis video 1, the recall was up to 100% and the precision was about 97%. There were 26 frames of false detection due to the factor that the scoreboard was not presented but some high-texture billboards appear with significant camera movement. In this case, we would detect large variation in the region where billboards were presented. In tennis video 2, the precision of caption frame detection was up to 98% and the recall was about 93%. The number of frames of miss detection was 31 because of the low intensity of the scoreboard in this video sequence. Besides, the color of the scoreboard and that of the tennis court were quite similar and hence it would be more difficult for caption detection in the case of low contrast between closed captions and the background. The worst case in detecting caption frames was presented in the baseball video since the background of several shot types was highly textured, such as the pitching shots and the audience shots. Therefore, when the camera moved, high-textured regions would be considered as the presence of captions. However, recall rate in detecting caption frames in the baseball video remained more than 80%.

Table 3-1. Performance of caption frame detection

Ground Truth of Caption Frames	Frames of Correct Detection	Frames of False Detection	Frames of Miss Detection	Miss Rate	Precision	Recall
Tennis 1	903	26	0	0%	97%	100 %
903						
Tennis 2	383	8	31	7%	98%	93%
414						
Volleyball	578	57	24	4%	91%	96 %
602						
Baseball	1290	407	264	24%	76%	83%
1554						
News	873	113	87	9%	88%	91%
960						
Average					90%	93%

The results of closed caption localization were shown in Table 3-2. In tennis video 1 41030 text-blocks were correctly detected, 347 blocks were falsely detected and 395 text blocks were missed. The precision was about 99% and the recall was about 97%. In tennis video 2, 24624 text blocks were detected, 732 blocks were falsely detected and totally 1056 text blocks were missed. Hence, the precision and recall of tennis video 2 was 97% and 95%, respectively. Some text blocks were missed since the background of the closed caption was transparent and would change with the background while camera moved. In this case, if the texture of the background was similar to the closed caption, the letters of captions cannot reflect the large variation in gradient energy and some text blocks would be missed. The precision rate of the baseball video in detecting text blocks was 81% due to the highly textured background. However, the recall rate was up to 92% since the temporal consistency was exploited to filter noise. Most of the blocks, which appeared for a short duration and their the spatial position were not consistent, were regarded as noise and were thus eliminated. The good performance was due to the reason that the weighted horizontal-vertical AC coefficients were exploited and the long-term consistency of the closed caption over consecutive frames was considered.

Table 3-2. Performance of closed caption localization after caption frame detected

Ground Truth of Text Blocks	Blocks of Correct Detection	Blocks of False Detection	Blocks of Miss Detection	Miss Rate	Precision	Recall
Tennis 1	41030	347	395	1%	99%	97%
42183						
Tennis 2	24624	732	1056	4%	97%	95%
25680						
Volleyball	27353	4087	2250	8%	87%	92%
28122						
Baseball	269792	63270	26676	9%	81%	91%
296405						
News	192016	23732	10080	5%	89%	95%
201616						
Average					91%	94%


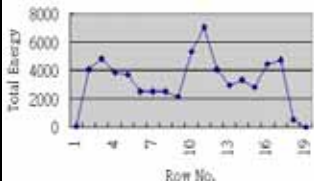

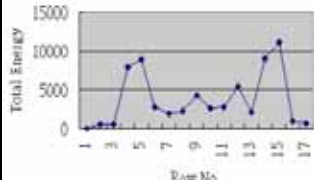


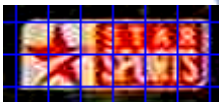
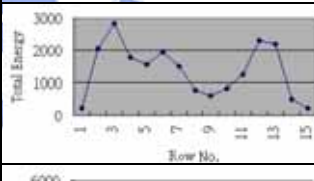
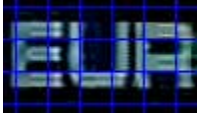
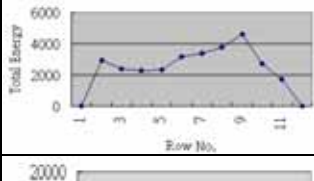
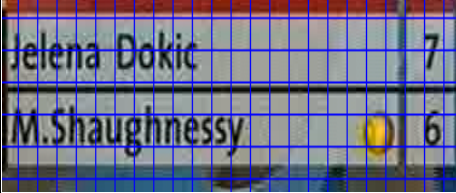
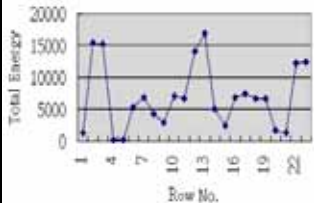

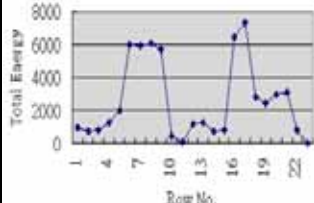
To manifest the feasibility of the approach of font size differentiation, we selected various closed captions with different fonts presented in the testing videos for experiments. The results were illustrated in Table 3-3. The first column of the table introduced the selected closed captions and the second one was the diagram of the curve of vertical AC gradient energy. The average font size of the associated closed caption in terms of the number of blocks was shown in the third column and the last column illustrated the font size of the different kinds of detected fonts. In these closed captions, each covered square represented a macroblock. In the experimental results, we can see that the font size of each closed caption was correctly discriminated. The most complicated case was the G^{th} closed caption since three kinds of fonts appear in the caption, which were all detected as the fonts of size 1, 4 and 2. In the case of bold-faced font shown in the A^{th} and B^{th} closed captions, we can also successfully detect their font size. Closed captions with capital and lowercase letter present together would be more difficult for differentiation of font size. For example, in the H^{th} closed caption, only three capital letters “A”, “S” and “S” were present in two text rows respectively. We cannot find out the regularity of font size with insufficient number of letters of the same typeface. In the H^{th} closed caption, although the font

size of the capitals was not detected, most letters in the caption were lowercase with font size 2 approximately and hence the detected result was still correct.

Moreover, the case of one row text was tested and an example was shown in the J^{th} closed caption of Table 3-3. We can observe that the energy in the top and bottom of the caption was much smaller than the energy in the middle since AC energy was computed in the processed text regions, in which localized text regions were expanded one block row up and down by the morphological operation. Additional block rows were normally the blank non-text regions, and thus the energy in the text blocks would be significantly larger than non-text blocks. Therefore, text-blocks can be successfully detected and font size in one row text can be recognized.

By applying the proposed approach of font size differentiation, we can automatically discriminate the font size either in a closed caption or in different ones. Therefore, this designed tool can be used as the closed caption filter to recognize and select those of interest, once the user indicates the targeted font size of closed captions. Moreover, researches [40-43] focusing on video OCR indicate that a bottleneck for recognizing characters was due to the variation of text font and size. In addition, to make learning data for the filter of character extraction, the size of the filter, which was defined to include a line element of characters, should be determined. Since the size of the line element strongly depends on the font size, it was possible to design a filter that can enhance the line elements dynamically with widely varying font sizes when the font size in the localized captions were known. Consequently, the tool – font size differentiation can be exploited to be a pre-processing tool for video OCR.

Table 3-3. Experimental results of font size differentiation based on horizontal projection profile using vertical DCT AC coefficients

Closed Captions	Analyses		
	AC Energy Variation	Font	
		Font Size (AVG)	Detected Font Size (blocks)
A. 		3.5	4, 3
B. 		5	5
C. 		2	2, 2
D. 		2.3	1.5, 3
E. 		4	4
F. 		5	5, 5
G. 		2.3	1, 4, 2

<p>H.</p> 		2	2
<p>I.</p> 		1.5	1.5
<p>J.</p> 		2	2

3.4.2 The Prototype System of Video Content Visualization

With the successful localization of the super-imposed scoreboard in sports videos, video content can be visualized in a compact form by constructing the hierarchical structure. Taking tennis as an example, the structured contents composed of scoreboards and the related can be combined with the detected tennis semantic events [46], such as baseline rally, serve and volley and passing shot. Each competition shots can be annotated using the type of corresponding event and can be labeled exploiting the scoreboard. Consequently, the information of the type of events, the boundary of events, the key frame of events and the result of the event – the scoreboard can be used in the Highlight Level Description Scheme shown in Fig. 3-16 to support users to efficiently browse videos by viewing the images of scoreboards and the important text information of semantic events. The name of highlight corresponded to the type of tennis event, the descriptor of video segment locator was described by the event boundary and the position of the key frame in the video sequence was used for the key image locator. The key image locator for scoreboard indicates the time point in the

video sequence.

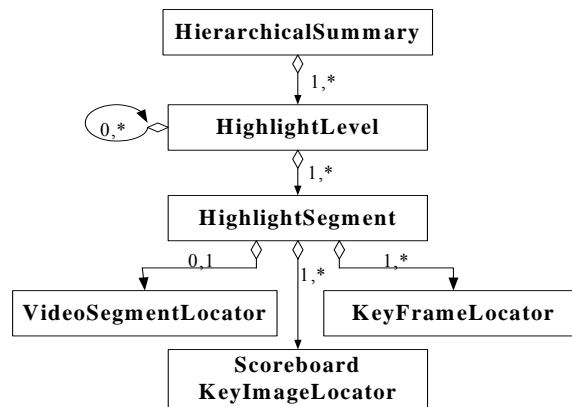


Fig. 3-16. Hierarchical Summary Description Scheme [17]

The table of video content was composed of the original video sequence in the top level, the scoreboard of a set, the scoreboard of a game and the key frame of one point. The user interface of the prototype system was shown in Fig. 3-17 and two areas of “Playback” and “Visualization” were present in the left and the right side, respectively. Initially, the key frame of the original video sequence and the scoreboards of sets were exhibited. While users can click the symbol “+” as the arrow lines indicated, the system would show the scoreboards of the corresponding games. Fig. 3-18 presented more detailed of the hierarchy. Users can select which game they want to watch according to the scoreboards of the games and click the symbol “+” for more detail and the result was shown in Fig. 3-19. Each point of the game was represented by its key frame. Users can view the point by clicking the corresponding key frame and the shot of the point would be displayed in the “Playback Area”. By exploiting the system of video content visualization, users can efficiently browse video sequences. Since the length of a sports video was up to one or two hours generally, the system thus provided a compact and brief overall view of the match for users by exhibiting the textual information of the scoreboards hierarchically.

We believe that the proposed video structuring method can be used in other well-structured sports, such as volleyball and baseball when the corresponding

domain knowledge is applied. In our previous research [22], volleyball videos were automatically structured when the rule of volleyball game was employed.

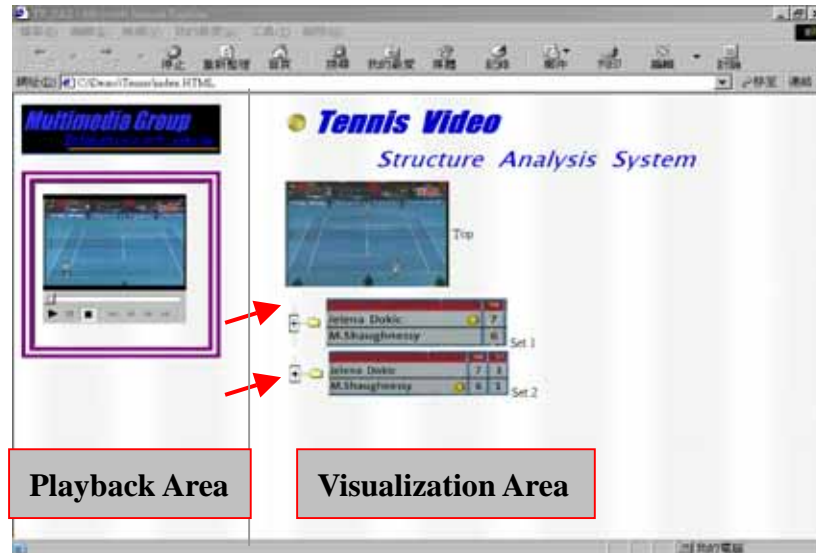


Fig. 3-17. Video Content Visualization System was composed of two areas – “Playback” and “Visualization”

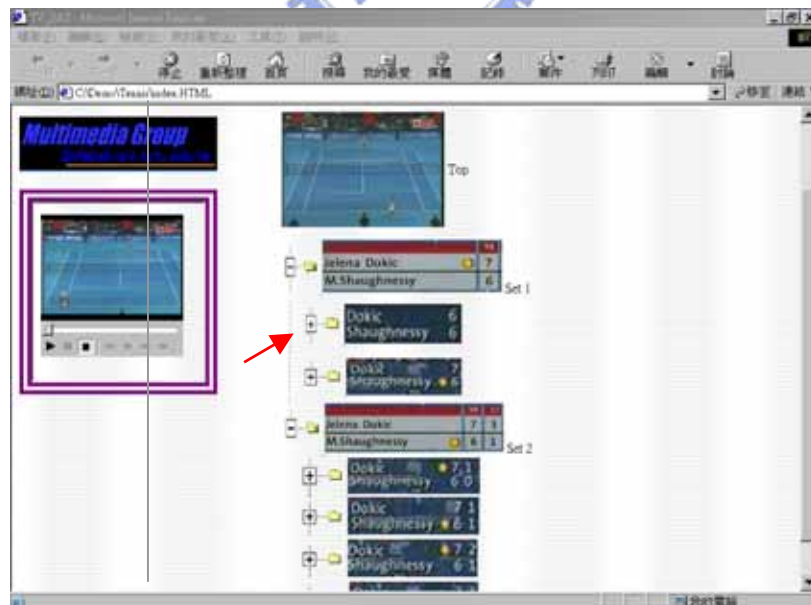


Fig. 3-18. The hierarchical structure of the scoreboards was shown while the user clicks the symbol “+”

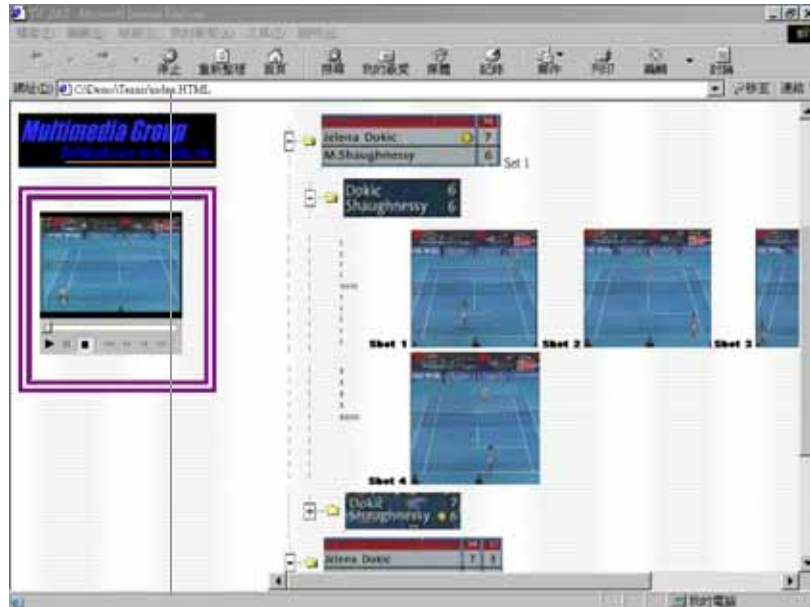


Fig. 3-19. Video shots were presented in the detailed video hierarchy

3.5 Summary

In the chapter, we have proposed a novel mechanism to detect temporal boundaries, identify meaningful shots and then build a compact table of video content. GOP-based video segmentation was used to efficiently segment videos into shots. To efficiently detect closed captions, color-based shot identification was proposed to identify shots of interest, especially for sports videos. Caption frames were detected in the shots of interest using the compressed data in MPEG videos. Then caption frames instead of every frame were selected as targets for detecting closed captions based on the long-term consistency without size constraint. While closed captions were localized, we differentiate the font size of closed captions based on the horizontal projection profile of AC gradient energy obtained from both the original blocks and the interpolated sub-blocks. The proposed tool – font size detector can thus be used as a prefilter to effectively eliminate uninterested closed captions and avoid most of the extremely time consuming post-processing of localized captions. Finally, having the proposed mechanism of high-level video structuring, one can browse videos in an efficient way through a compact table of content.

Chapter 4. Motion Activity Based Shot Identification and Closed Caption Localization for Video Structuring

4.1 Introduction

Tremendous growth in the number of digital videos is driving the need for more effective methods to access and acquire desired video data. Advances in automatic content analysis and feature extraction enable the possibility of effective browsing, searching and filtering of videos. On the other hand, well-developed content-based indexing techniques equip the users with natural and friendly querying, searching, browsing and retrieving tools. For supporting video content representation and indexing, semantic features of higher level must be prepared for achieving more efficient and effective access. The need of representation and indexing for high-level and semantic features underlies the emergence of the MPEG-7, formally called multimedia content description interface. However, the approaches that produce the desired features are a non-normative part of MPEG-7 and are left open for research and future innovation.

Video structuring is a move intending to organize raw video data into a compact, easy-to-access format. Lu and Tang [47] described a video-structuring scheme, which classifies and clusters sports video shots based on low-level features, color and information on global motion. Kwon et al. [48] presented a scene segmentation scheme based on the adaptive weighing of color and motion features. For integrating scene units, they applied an improved overlapping link scheme to achieve the goal. Hanjalic and Lagendijk [49] segmented movies into logical story units based on the global temporal consistency of the color features. Yeung and Yeo [50] proposed a time-constrained and MPEG DC based visual similarity clustering method to segment

a video into logical story units. All the aforementioned research structures videos using low-level features, such as color and motion, and merges shots to generate logical story units based on visual similarity, by applying some time-constrained mechanisms. However, the classification of video shots based on motion activity information of objects or events that little efforts have involved in would be more semantically meaningful. Although visual content is a major source of information in a video, an effective strategy in video structuring is to exploit other valuable information such as text in superimposed closed captions. Therefore, there is an increasing research in localizing superimposed closed captions in video programs either in raw videos [31-37] or in compressed videos [29-30][35]. Li et al. [32] exploited a neural network trained on texture features to obtain text regions and proposed a text-region tracker for tracking of moving text. Shim et al. [33] segmented text areas using chaincodes in the pixel domain and exploited temporal information to refine the segmentation of text. Both Li et al. [32] and Shim et al. examined the similarity among text regions in terms of their positions, intensities and shape features. Chen and Zhang [31] detected text areas using information on vertical edges followed by information on horizontal edges before applying a Bayesian based shape suppression technique for refining the results. Ohya et al. [34] segmented characters by setting a local threshold and merging neighboring regions based on the similarity of gray levels. Kannangara et al. [36] extracted text from specific areas and proposed a method based on the vertical projection profile to segment individual letters. Wu et al. [37] segmented text areas that exploited both multiscale texture segmentation and a spatial cohesion constraint in the pixel domain. Zhong et al. [29], and Zhang and Chua [30] localized text in MPEG videos using DCT AC coefficients to obtain texture information in individual I-frames. Zhang and Chua also identified text regions using a size filter. Gargi et al. [35] detected text by counting the number of intra-coded

blocks in P and B frames based on the assumption that the background is static and a threshold for the size of text segments is also predefined to filter noise.

Previous research made little efforts to localize superimposed closed captions in compressed videos. Besides, the localization of text regions using size filters may not work well, especially in cases in which captions are small but very important and meaningful to viewers. For example, in sports videos, the scoreboard is generally very small but it is significant as it details the competition as clearly as possible. In addition, automatic post-processing in the detected potential text regions is a critical step to speed up following analysis in caption regions, such as video OCR. Previous research made little efforts on filtering captions once potential text regions are localized, such as separating the superimposed captions from the highly textured regions in the background. Therefore, both the identification of text with no size constraints and the filtering of detected caption regions are of concern.

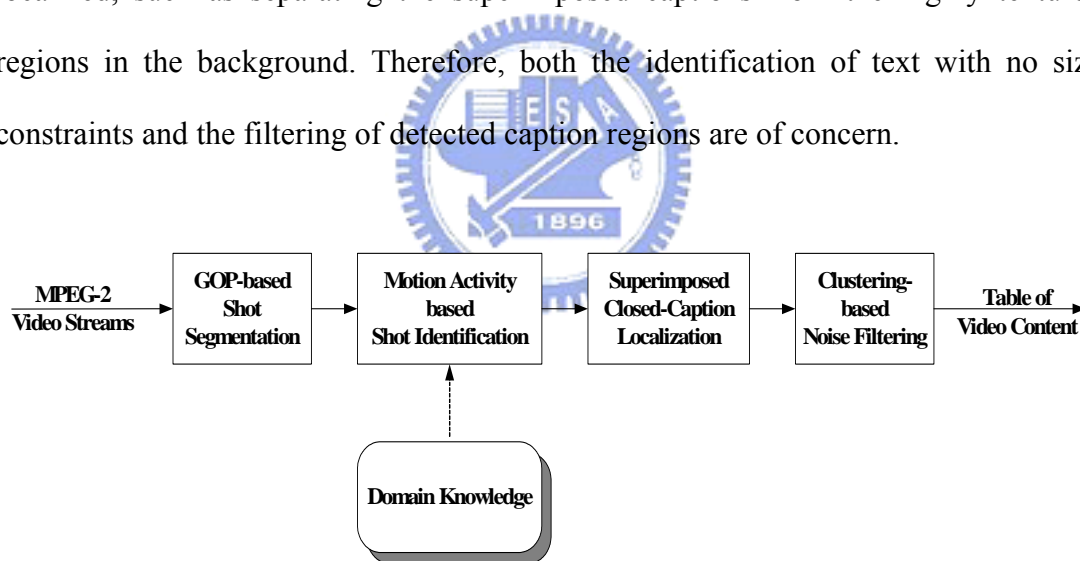


Fig. 4-1. System architecture of motion activity based video structuring

In this chapter, in order to support high-level and semantic-based browsing, we propose a novel approach that structures videos exploiting superimposed closed captions and semantic classes identified by the motion activity descriptor of object-based 2D-histogram. Fig. 4-1 shows the architecture of the proposed system. First, video streams are efficiently segmented into shots using our proposed GOP-based detection of shot changes. This video segmentation module checks video

streams GOP-by-GOP rather than frame-by-frame, and then determines the actual shot change boundaries on the frame level. Video shots are characterized utilizing the proposed motion activity descriptor [51], which represents the spatial distributions of moving objects in a compact form. When the specific domain knowledge is applied, the model of object distributions can be employed to infer the semantic meaning or event in a shot. Accordingly, based on the criterion, video shots are classified into semantic classes. The shots of interests are then selected for localizing superimposed closed captions. Furthermore, the algorithm of clustering-based noise filtering is designed to effectively remove high-textured regions.

The rest of this chapter is organized as follows. Section 4.2 illustrates the GOP-based detection of shot change and Section 4.3 describes identification of shots based on object motion activity. Section 4.4 introduces the approach of localization of superimposed closed captions. Section 4.5 presents the experimental results and Section 4.6 draws conclusions and the future work.

4.2 Video Segmentation

Video data are segmented into meaningful clips to serve as logical units called “shots” or “scenes”. In MPEG-2 format, the GOP layer can be randomly accessed and contains a GOP header and a series of encoded pictures, including I, P and B-frames. A GOP is approximately 10 to 20 frames, normally with duration shorter than two consecutive shot changes (around 20 frames).

Possible occurrences of shot change are examined GOP-by-GOP (inter-GOP). The difference between each consecutive GOP-pair is computed by comparing the I-frames in each consecutive GOP-pair. If the difference between the DC coefficients of these two I-frames exceeds a threshold, then there may have shot change between these two GOPs. Hence, the GOP that contains the shot change frames is identified. In

the second step, detecting intra GOP shot change, the ratio of forward and backward motion vectors are further used to locate the exact frame of the shot change within a GOP. The experimental results obtained using real extensive videos are encouraging and prove that shot changes are efficiently detected for video segmentation.

4.3 Shot Identification

This section introduces the method of shot identification based on object motion activity. Section 4.3.1 describes the method of detecting significant moving objects and section 4.3.2 shows the motion activity descriptor. Section 4.3.3 presents shot identification based on the descriptor.

4.3.1 Moving Object Detection

For computational efficiency, motion information in P-frames is used for the detection of moving objects. In general, consecutive P-frames separated by two or three B-frames are still similar and would not vary too much. Therefore, it is reasonable to only use P-frames as targets for moving objects detection. On the other hand, since the motion vectors estimated in MPEG-2 videos may not be 100% correct, one has to remove noisy motion vector before the motion vectors are clustered. For those motion vectors that are small in magnitude, we consider they are noises and should be removed. For computational efficiency, the average of motion vectors in those inter-coded macroblocks is computed and selected as the threshold for noise removal. After noisy motion vectors are filtered out, motion vectors of similar magnitude and direction are clustered into a group (an object) by applying a region growing process.

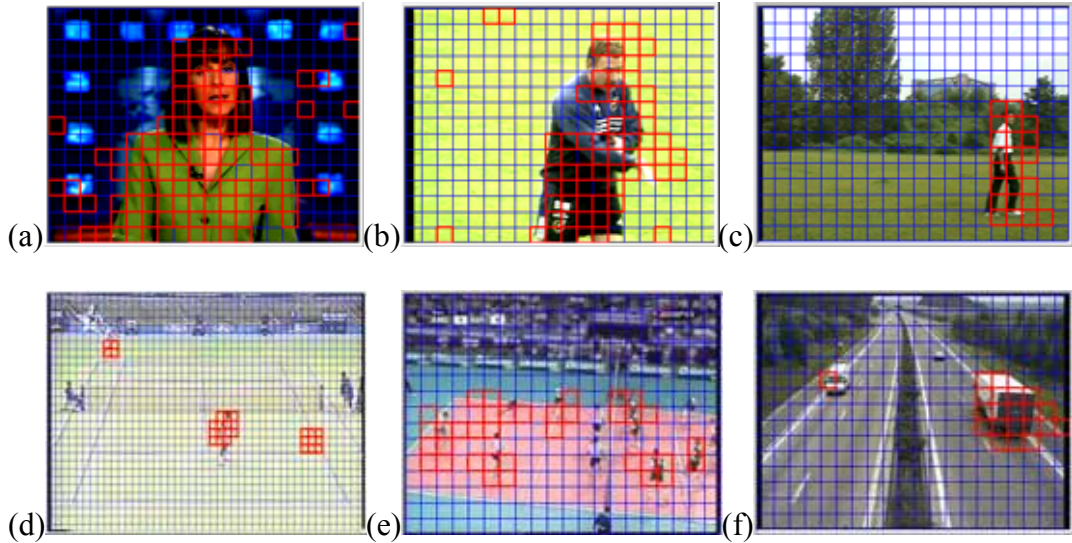


Fig. 4-2. Moving objects detection; (a) anchor person; (b) football; (c) walking person; (d) tennis; (e) volleyball game; (f) traffic monitoring

In our previous works [46] [51], we have successfully detected moving objects in several kinds of videos such as tennis, traffic monitoring, news and football. Moving objects in the environment of static camera are almost detected with both precision and recall higher than 95%. In the videos with moving camera, camera motion such as pan and tilt is estimated by determining the dominant motions before detecting moving objects. Under this circumstance, although the precision is lower than 90%, the recall remains higher than 90%. Examples of moving region detection are demonstrated in Fig. 4-2. Video shots shown in Figs.4-2(a) – 4-2(c) and Fig.4-2(f) are extracted from the MPEG-7 testing dataset – the Spanish News and the traffic monitoring. The tennis shot shown in Fig. 4-2(d) is recorded from the Star-Sports TV-channel. Fig. 4-2(e) shows the shot of volleyball game recorded from the ESPN TV-channel. In the volleyball videos, although several players may be clustered as one moving object such as the example in Fig 4-2(e), the spatial distribution of moving objects can still be characterized when the attributes of object size and object position are employed. The details of the characterization of moving objects using these object attributes in the proposed object-based motion activity descriptor will be

described in the following section. Based on the results shown in Fig. 4-2, it is obvious that most moving objects are successfully detected. Although the detection algorithm detected some noises, usually one can eliminate this kind of noise via tracking moving objects in the forward and backward frames when more precise requirements are needed.

4.3.2 Motion Activity Descriptor – 2D Histogram

In this section, we shall elaborate how to describe object-based motion activity for a video shot considering the attributes of object size and object position. In order to describe the spatial relationships between moving objects in a compact manner while keeping the distinct and recognizable features among video shots, a video shot is characterized using the statistics derived from the object-based 2D-histogram. A 2D-histogram for each P-frame consists of a X-histogram and a Y-histogram, in which the horizontal axis of the X-histogram (Y-histogram) is the quantized into β bins. The workflow of 2D-histogram computation is shown in Fig. 4-3. Initially, size of the object is estimated before it is assigned to a bin. If the object is larger than the predefined unit size ($frame-size/\beta^2$), it is weighted and accumulated according to Eq. (4-1). $Bin_{i,j}^x$ refers to the j^{th} bin of the X-histogram in frame i . $Acc_{i,j,\alpha}^x$ means the accumulated value in the j^{th} bin of *object* α in *frame* i for the X-histogram, and Obj represents the number of objects in frame i . Fig. 4-4 provides an example of the 2D-histogram. In the example, the frame includes two objects of size of three units and four units. The size of each object is assigned to a histogram bin according to the position of its centroid on the horizontal axis to obtain the X-histogram. The football player of size three is assigned to the Bin 1 and the basketball player of size four is assigned to Bin 3 in the X-histogram. Similarly, in the Y-histogram, the Bin 2 is increased by three and the Bin1 is increased by four.

$$Bin_{i,j}^x = \sum_{\alpha=1}^{Obj} Acc_{i,j,\alpha}^x, \quad (4-1)$$

$$where \quad Acc_{i,j,\alpha}^x = \begin{cases} 1, & \text{if object size} \leq \frac{1}{\beta^2} \text{ frame size} \\ \frac{\text{size of object } \alpha}{\text{frame size}} * \beta^2, & \text{otherwise} \end{cases}$$

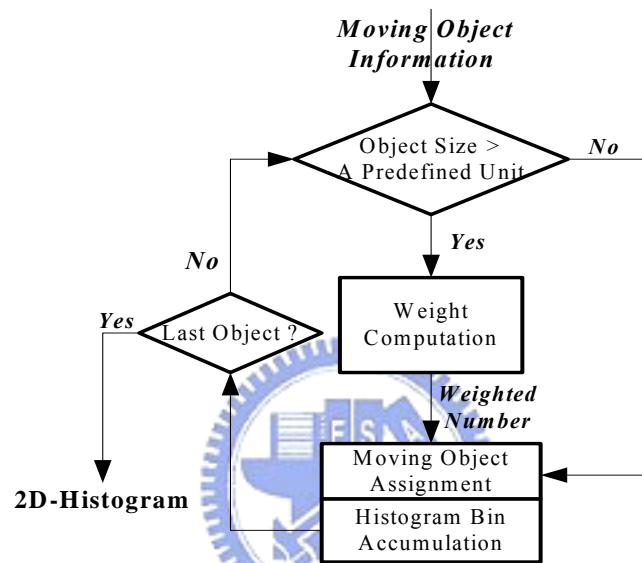


Fig. 4-3. Workflow of motion activity descriptor

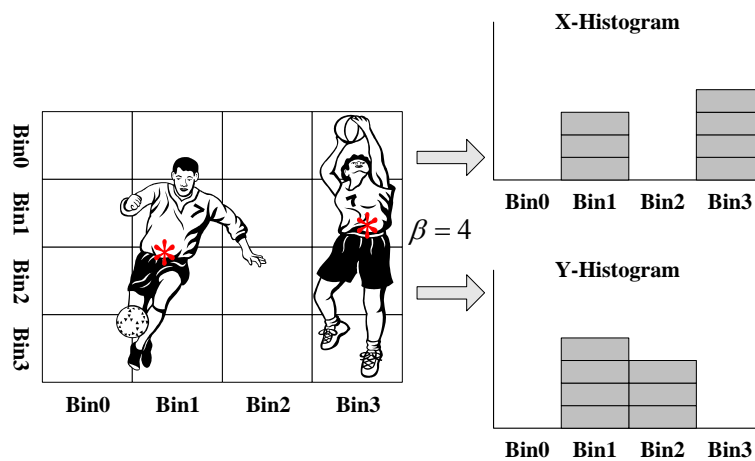


Fig. 4-4. An example of 2D-histogram computation

Using the proposed 2D-histogram, the spatial distribution of moving objects in

P-frames is characterized in a concise form. The X-Y histogram pair shows the spatial relationships among the moving objects since each moving object is assigned to a histogram bin according to the X-Y coordinate of its centroid. Objects belong to the same coordinate interval are grouped into the same bins. Hence, the distance between object groups can be specified as the distance between the associated bins.

4.3.3 Shot Identification Algorithm

Automatically identifying video shots or events is a domain-specific problem, because it requires higher-level content analysis. For sports videos, interesting shots are specific to a particular model of visual features, such as the spatial distribution of moving objects. Therefore, employing domain knowledge in sports videos to recognize specific video shots is indispensable. In this chapter, we select the volleyball game as the case study. In general, a volleyball game mainly consists of three shot types - the “service”, “full-court view” and close-up”. Fig. 4-5 presents the typical frames of these three shot types. The service shots have the characteristic that one or few objects appear in the left or right sides of the frame and more objects appear in the other side of the frame. In full-court view shots, the number of objects on the left is similar to the number of objects on the right and the difference between the numbers of objects is smaller than that in the service shots. In the close-up shots, a large object is near the middle of the frame. Accordingly, these main types of shots in volleyball videos can be distinguished according to the distribution of moving objects. Although description using both X-histogram and Y-histogram would be more detailed and complete than using one of them, it is reasonable to use the X-histogram only to distinguish these main shot types because most players in the volleyball games move along the horizontal axis. The algorithm of shot identification is based on K-means clustering and here K is set to four according to the number of shot types in the volleyball games (two for the type of “Service”, one for “Full-court view” and one

for “Close-up”). The algorithm is detailed as follows. After shots clustering, four clusters are obtained with their centroids, μ_1 , μ_2 , μ_3 and μ_{middle} , and the shot type of them can be determined by comparing the variances of feature subspaces Var^1 , Var^2 , Var^3 and Var^{middle} . If Var^1 or Var^3 is larger than others, the shot type is considered as “Service” because the server may serve in the right or left side of the court. If Var^2 is the largest among the four variances, the shot type is regarded as “Close-up”. Otherwise, the shot type is identified as “Full-court view”.

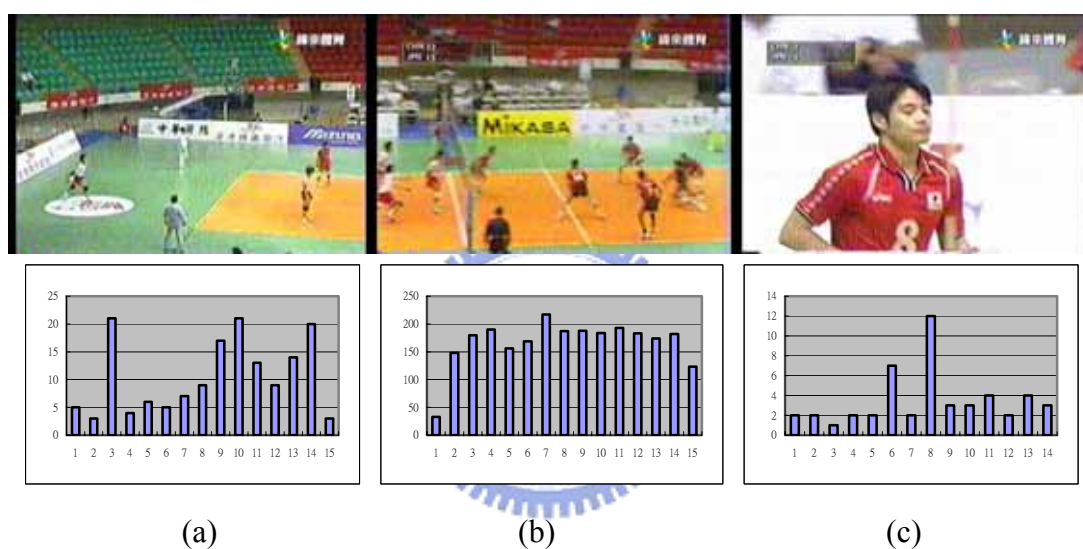


Fig. 4-5. Histograms of shots; (a) Service; (b) Full-court view; (c) Close-up

Shot Identification Algorithm

Input: Segmented shots $\{Shot_1, Shot_2, \dots, Shot_s\}$

Output: Shot types: $\{ST_1, ST_2, \dots, ST_s\}$, where the type of shot i $ST_i \in \{S, F, C\}$ (S:

Service, F: Full-court view, C: Close-up)

1. X-coordinate is divided into β bins.
2. For each shot, compute the representative motion activity descriptor.

$$Xhist_j = \frac{1}{|Shot_j|} \sum_{i=1}^{|Shot_j|} H_i, \text{ where } H_i = [b_1, b_2, b_3, \dots, b_\beta], j \in [1, s]$$

3. Choose the initial cluster centroids, $\mu_1, \mu_2, \dots, \mu_K$

3.1 Divide each X-histogram $Xhist$ into K subspaces in which each subspace is of $m = \beta/K$ dimensions.

$$Xhist_j^\rho = [b_{(\rho-1)m+1}, b_{(\rho-1)m+2}, b_{(\rho-1)m+3}, \dots, b_{\rho m}], \text{ where } \rho \in [1, K]$$

3.2 For each subspace, compute the variance Var_j^ρ within its m elements

$$Var_j^\rho = \left(\sum_{\omega=(\rho-1)m+1}^{\rho m} (b_\omega)^2 / m \right) - \left(\left(\sum_{\omega=(\rho-1)m+1}^{\rho m} b_\omega \right) / m \right)^2$$

3.3 Choose the initial cluster centroids $\mu_1, \mu_2, \dots, \mu_K$ by

$$\mu_\rho = \arg \max_{1 \leq j \leq s} Var_j^\rho,$$

and choose a particular cluster centroid μ_{middle} by

$$\mu_{middle} = \arg \min_{1 \leq j \leq s} Var_j^{middle}, \text{ where index } middle \text{ is determined by } \lceil \beta/2 \rceil$$

4. Classify each feature vector $Xhist_j$ of shot j to the cluster ρ with the smallest distance

$$C_\rho \leftarrow \left\{ Xhist \mid \arg \min_{1 \leq \rho \leq K+1} |Xhist_j - \mu_\rho| \right\}, \text{ where } |\cdot| \text{ means the distance between two feature vectors by summing up the absolute difference of two histogram bins.}$$

5. Update cluster centroids $\mu_\rho = \frac{1}{|C_\rho|} \sum_{n=1}^{|C_\rho|} Xhist_n^\rho$

6. If any cluster centroid changes its value, go to step4.

7. For each cluster, determine its shot type ST

If $\arg \max_{1 \leq \rho \leq K+1} Var^\rho \in \{1, 3\}$, then $\forall Shot_i \in \{\mu_1, \mu_3\}, ST_i \in S$

Else if $\arg \max_{1 \leq \rho \leq K+1} Var^\rho = 2$, then $\forall Shot_i \in \mu_2, ST_i \in C$

Otherwise, $\forall Shot_i \in \mu_{middle}, ST_i \in F$

4.4 Closed Caption Localization

In this section, we shall elaborate how to localize superimposed closed captions in I-frames with directly exploiting the compressed data. The proposed approach is shown in Fig. 4-6. First, the horizontal gradient energy is computed to filter out noise using DCT AC coefficients. The next step is to remove noisy regions by applying the morphological operation. When the candidate caption regions are detected, the clustering-based algorithm is then employed to filter out high-textured non-caption regions. Section 4.4.1 details the detection of the closed captions and section 4.4.2 shows the algorithm of SOM-based filtering.



Fig. 4-6. Closed caption localization in video frames

4.4.1 Localization of Superimposed Closed Captions

After shots of interest are identified, the approach of closed caption detection is proposed to localize the superimposed closed captions in these shots, such as the scoreboard and the channel trademark. To efficiently localize captions in compressed videos, several DCT AC coefficients shown in Fig.4-7 are used to compute the horizontal and vertical gradient energy. The horizontal gradient energy defined by Eq.(4-2) is computed using the AC coefficients from $AC_{0,1}$ to $AC_{0,7}$. Due to the fact that some blank space appears between consecutive letters in closed captions, the variation of the gradient energy in the horizontal direction would be more frequent and larger than that in the vertical direction. Hence, it is reasonable to filter out non-caption regions using the horizontal gradient energy. For each 8×8 block, the horizontal gradient energy E_h is exploited to determine the block type. If the E_h of a block exceeds a predefined threshold, then the block is regarded as a potential

caption block. Otherwise, if the E_h of a block is below the threshold, then the block is removed.

DC	$AC_{0,1}$	$AC_{0,2}$	$AC_{0,3}$	$AC_{0,4}$	$AC_{0,5}$	$AC_{0,6}$	$AC_{0,7}$
$AC_{1,0}$							
$AC_{2,0}$							
$AC_{3,0}$							
$AC_{4,0}$							
$AC_{5,0}$							
$AC_{6,0}$							
$AC_{7,0}$							

Fig. 4-7. DCT AC coefficients used in localizing superimposed closed captions

$$E_h = \sum_{j=1}^7 |AC_{0,j}| \quad (4-2)$$

However, various shots may have different lighting conditions, which are reflected in the contrast in frames. Besides, the contrast impacts the determination of the threshold and the detection of the closed captions might fail for this reason. Therefore, the threshold is determined adaptively according to contrast that is evaluated using horizontal gradient energy. The threshold T_s is computed by Eq. (4-3), where γ is an adjustable factor; $SVar_s$ represents the average of horizontal gradient energy of shot s , $FVar_{s,i}^{AC}$ represents the horizontal gradient energy of frame i in shot s , AC_h is the horizontal DCT ac coefficient from $AC_{0,1}$ to $AC_{0,7}$, M denotes the number of P-frames in a shot and N means the number of blocks in a frame. Due to the fact that a higher $FVar_{s,i}^{AC}$ implies a higher E_h contrast in frame i , noisy regions can be more easily removed from a frame of higher gradient energy. Therefore, a lower weight is assigned to the frame with a higher contrast and a higher weight is assigned to one with a lower contrast. Accordingly, using this method, most of the non-caption regions can be removed. Fig. 4-8(b) demonstrates the results filtered using E_h .

$$T_s = \gamma \times SVar_s, \quad \gamma = \begin{cases} 3.2, & \text{when } FVar_{s,i}^{AC} < SVar_s \\ 2.4, & \text{when } FVar_{s,i}^{AC} \geq SVar_s \end{cases} \quad (4-3)$$

$$SVar_s = \frac{1}{M} \sum_{i=1}^M FVar_{s,i}^{AC}$$

$$FVar_{s,i}^{AC} = \sum_{j=1}^N \sum_{h=1}^7 AC_{h,j}^2 / N - \left(\sum_{j=1}^N \sum_{h=1}^7 |AC_{h,j}| / N \right)^2$$



(a)	
(b)	(c)
(d)	(e)

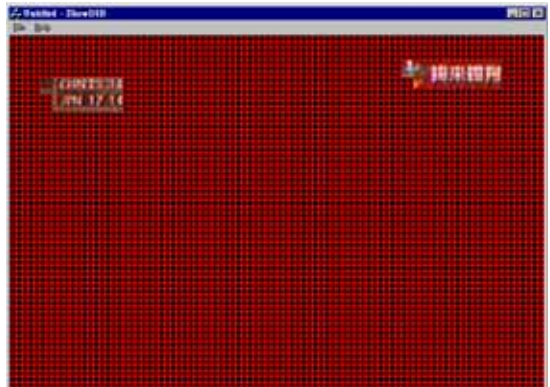
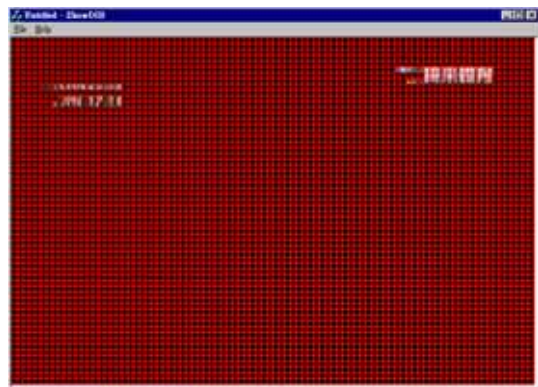
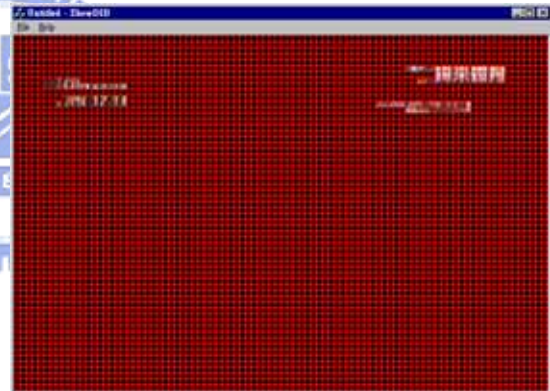
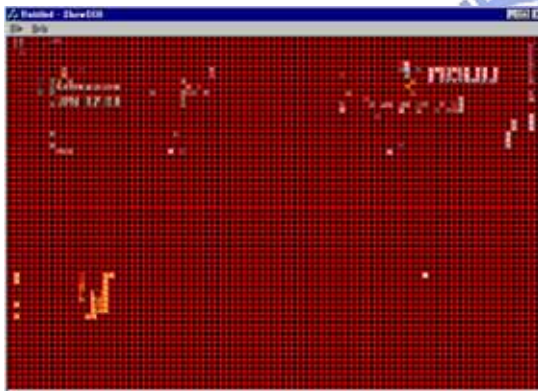


Fig. 4-8. Demonstration of the localization of superimposed closed captions (a) original I-frame; (b) result after filtering by using horizontal gradient energy; (c) result after morphological operation; (d) result after filtering using SOM-based algorithm; (e) result after dilation

After most of the non-caption regions are eliminated, several small separate regions remain and are either very close or faraway from each other. Some regions are supposed to be connected, such as the scoreboard and the channel trademark. Hence, neighboring regions must be merged and isolated ones should be removed. Therefore, a morphological operator of 1x3 blocks is used to merge regions separated by a distance of under three blocks. Fig. 4-8(c) shows the result of applying a morphological operation. Several small and isolated regions are filtered out and the caption regions are merged. However, some background regions with large horizontal gradient energy remain present after morphological operation. Hence, an algorithm based on the concept of SOM (Self-Organization Map) is proposed to differentiate the foreground captions from the background highly textured regions.

4.4.2 Clustering-Based Noise Filtering

The Self-Organizing Map-based algorithm [52] has been applied to segment and recognize textures, and is well suited to the task of classifying textures. A SOM-based noise-filtering algorithm is proposed to further differentiate the foreground captions from the background highly textured regions. The details of the algorithm are described as follows.

SOM-Based Noise Filtering Algorithm

Input: Candidate regions after morphological operation $\Psi = \{R_1, R_2, \dots, R_n\}$

Output: Closed caption regions

1. Initially, the cluster number is set to *zero* ($j=0$).
2. For each candidate region R_i , the average horizontal-vertical gradient energy E_i , weighted by w_h and w_v , is computed. Here, w_h is set to 0.6 and w_v is set to 0.4. n is the number of regions in Ψ .

$$E_i = \frac{1}{n} \sum_{j=1}^n \left(w_h \sum_{u=1}^7 |AC_{0,u}| + w_v \sum_{v=1}^7 |AC_{v,0}| \right) \quad (4-4)$$

3. For each region $R_i \in \Psi$

If $i = 1, j = j + 1$; assign R_i to cluster C_j

Else if there is a cluster C such that $D_k \leq T$ and D_k is minimal among $\{D_k\}$, where $k \in [1, j]$ and D_k is defined in Eq. (5)

assign R_i to C

$$D_k = \frac{2}{|C_k|(|C_k| - 1)} \sum_{i=1}^{|C_k|} \sum_{j=i+1}^{|C_k|} |E_i - E_j| \quad (4-5)$$

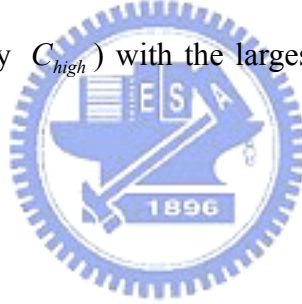
Else

$j = j + 1$; create a new cluster C_j and assign R_i to C_j

4. Set $T = T - \delta$

Select the cluster C_k (say C_{high}) with the largest mean gradient energy $E_{avg,k}$, computed by Eq. (6)

$$E_{avg,k} = \frac{1}{|C_k|} \sum_{i=1}^{|C_k|} E_i \quad (4-6)$$



5. If D_{high} of C_{high} exceeds T , then reset $\Psi = C_{high}$.

Go to step 3.

Else

Go to step 6.

6. The cluster C_{high} is the set of closed captions.

In the algorithm, more weight is assigned to the horizontal DCT AC coefficients than the vertical ones because closed captions generally appear in rectangular form and the AC energy in the horizontal direction would then exceeds that in the vertical direction because the letters of each word are fairly close to each other whereas the distance between two rows of text is relatively large. Furthermore, the SOM-based

candidate region clustering is iterated until the gradient energy $E_{avg,k}$ of the cluster C_{high} is smaller than the threshold T . Based on the experiments, T is set to 70; δ is set to 11 empirically, and the number of iterations is two or three. Using this method, the set of closed captions can be automatically determined. This method is based on the fact that closed captions are in the foreground and are superimposed after filming. Consequently, the closed captions are clearer and their gradient energy exceeds that of the background. After SOM-based noise filtering, each closed caption region is dilated by one block row. The result is shown in Fig. 4-8(e) and we can see that regions belonging to the same closed caption are merged.

4.5 Experimental Results and Analysis

Two video sequences are recorded from the VL Sports and ESPN TV channels respectively and encoded in the MPEG-2 format in which the GOP structure is IBBPBBPBBPBBPBB and the frame rate is 30 fps. Since testing videos, Video I and Video II demonstrated in Fig.4-9, are recorded from different games, the background color, background texture, object color and lighting effect in these videos are thus different. The length of the Video I is about one hour and 163 shots of services, competition of the full-court views and close-up are obtained and the length of the Video II is around one and half hour and it consists of 199 shots. To measure the performance of the proposed scheme, precision and recall for the approach of shot identification and the algorithm for detecting closed captions are evaluated. Table 4-1 and Table 4-2 show the experimental results of the shot identification in Video I and Video II, respectively. The precision of identification of all three kinds of shots in Video I and II are both higher than 92%. The values of recall in close-up shots of both videos are up to 98%. The recall values of full-court view shots is just 87% in Video I and 89 in Video II since the camera zooms in to capture shots in which players spike

near the net. In such a case, the shot would include a large portion of the net and a large object would be detected; the shot is thus regarded as a close-up shot. Additionally, when a team is defending, several players may run to save the ball. In this situation, the number of objects on the left might not be similar to the number of objects on the right and thus the shot is classified as a service shot. Although the recall value of the full court shot does not exceed 90%, the overall accuracy of shot identification is still very favorable.



Fig. 4-9. Demonstration of testing videos: (a) Video I (b) Video II

Table 4-3 presents the results of closed caption localization. In Video I, 107 potential captions are detected in which 98 localized regions are the real closed captions including the scoreboard and the trademark. In Video II, there are 125 closed captions containing the scoreboard and the trademark and 128 potential captions are detected in which 118 localized regions are the real closed captions. The recall value reaches 100% and the precision is around 92% in Video I and the recall value is about 94% and the precision is about 92% in Video II. The number of false detection in Video I is nine and that the number of false detection in Video II is ten because the background may include an advertising page whose gradient energy is similar to that of the scoreboard and the channel trademark. In such a case, this high-textured region is falsely detected as the closed caption. Fig. 4-10 presents an example. In Fig.

4-10(d), the billboard is not filtered out since its gradient energy is stronger than the superimposed scoreboard and the trademark.

Table 4-1. Result of shot identification (Video I: 163 shots)

Ground Truth	Number of Detection	Number of Correct Detection	Number of False Detection	Number of Miss Detection	Precision	Recall
Closed-up	62	57	5	1	92%	98%
58						
Service	52	49	3	4	94%	92%
53						
Full Court	49	45	4	7	92%	87%
52						

Table 4-2. Result of shot identification (Video II: 199 shots)

Ground Truth (video 2)	Number of Detection	Number of Correct Detection	Number of False Detection	Number of Miss Detection	Precision	Recall
Closed-up	73	70	3	1	96%	98%
71						
Service	65	60	5	4	92%	94%
64						
Full Court	61	57	4	7	94%	89%
64						

Table 4-3. Result of closed caption localization

Ground Truth	Number of Detection	Number of Correct Detection	Precision	Recall
Video 1	107	98	91.59%	100%
98				
Video 2	128	118	92.18%	94.4%
125				

Fig. 4-11 shows the initial graphical user interface of the video browsing system. The table of video content is provided, in which the scoreboard at each game point is in the “Closed Caption” field and the representative frames of the three types of shot are shown in the “Service Shot”, “Full-Court Shot” and “Close-Up Shot” fields, respectively. Semantic high-level video structuring provides users an overall view of the competition as textual information in the scoreboard, and allows users to select the

point to watch, browsing through the video sequences in the different levels of detail. Additionally, when users want to see smashes, defense or offense, they can select full court view shots. Fig. 4-12 depicts all full-court view shots when users click the option “show all shots” in the “F shot” field. Moreover, when users want to see their favorite players, they can watch close-up view shots. Fig. 4-13 shows all “one-point” close-up shots obtained by selecting the “show other shots” option in the close-up shot field.



Fig. 4-10. Closed caption localization; (a) original I-frame; (b) result after filtering by horizontal gradient energy; (c) result after morphological operation; (d) result after filtering by SOM-based algorithm; (e) result after dilation



Fig. 4-11. Video structure of caption frames as well as service, full-court view, and close-up shots



Fig. 4-12. The bottom of the interface shows full-court shots



Fig. 4-13. The bottom of the interface presents close-up shots

4.6 Summary

In this chapter, we propose a novel mechanism to automatically structuring volleyball videos in the MPEG compressed domain and construct the table of video content employing both the localized scoreboard and the semantic classes of shots. GOP-based video segmentation is used to efficiently segment videos into shots. The spatial distribution of moving objects is characterized using the object-based motion activity descriptor. Experimental results indicate that the proposed descriptor effectively identified several shot types in volleyball videos. Additionally, experimental results in localizing superimposed closed captions also show that the target captions are successfully localized and differentiated from the high-textured background regions. These target captions and the shots in semantic classes are well organized in a compact form. Therefore, users are allowed to browse videos nonlinearly in an efficient manner through the table of video content following either

the scoreboards or the semantic classes of shots. Although only volleyball games are used in the experiments, the proposed mechanism provides several reusable modules like the descriptor of motion activity and the method of closed caption detection. Once the spatial distribution model of moving objects is obtained from employing specific domain knowledge, shots of interest such as the full or partial view of athletic field with particular player distribution can be automatically identified using the proposed object-based motion activity descriptor.

In the future, with the successful identification of shots in volleyball games in this chapter and the effective classification of video shots of MPEG-7 testing dataset in our previous research, we would like to apply the proposed system architecture for the motion activity shot identification/classification to other videos, including movies, documentaries and other sports. In addition, we will investigate video OCR to recognize the localized closed captions and thereby to support the automatic generation of meta-data, like the names of teams in sports videos, the names of leading characters in movies, or important people in other kinds of videos.

Chapter 5. Robust Video Sequence Retrieval Using A Novel Object-Based T2D-Histogram Descriptor

5.1 Introduction

The tremendous growth in the number of digital videos has become the main driving force for developing automatic video retrieval techniques. Among different types of tools that can push the advancement of retrieval techniques, an efficient automatic content analyzer that can help execute correct browsing, searching and filtering of videos is a must. In order to achieve this goal, one has to make use of high-level semantic features to represent video contents. The need of representing high-level semantic features has motivated the emergence of MPEG-7, formally called the multimedia content description interface [53]. However, the methods that produce the specific features and the corresponding similarity measures represent the non-normative part of MPEG-7 and are still open for research and future innovation. Usually, the high-level semantic features of video sequences can be inferred from low-level features. The low-level features can be color distribution, texture composition, motion intensity and motion distribution. Among different types of features that can be extracted from a video, motion is considered as a very significant one due to its temporal nature. In the literature, Divakaran et al. [54] used a region-based histogram to compute the spatial distribution of moving regions. The run-length descriptor in MPEG-7 [55] is used to reflect whether moving regions occurred in a frame. Aghbari et al. [56] proposed a motion-location based method to extract motion features from divided sub-fields. Peker et al. [57] calculated the average motion vectors of a P-frame and those of a video sequence to be the overall motion features. In addition to the above mentioned local motion features, Wang et al. [58] proposed to use some global motion features to describe video content.

In contrast to the motion-based features of individual frames, another group of researchers proposed to use spatio-temporal features between successive frames because these types of features are more abundant in the amount of information. Wang et al. [59] extracted features of color, edge and motion, and measured the similarity between temporal patterns using the method of dynamic programming. Lin et al. [60] characterized the temporal content variation in a shot using two descriptors - dominant color histograms of group of frames and spatial structure histograms of individual frames. Cheung and Zakhor [61] utilized the HSV color histogram to represent the key-frames of video clips and designed a video signature clustering algorithm for detecting similarities between videos. Dimitrova et al. [62] represented video segments by color super-histograms, which are used to compute color histograms for individual shots. Other works that fall into this category can be found in [63-67].

There are several drawbacks associated with the key-frame based matching process. First, the features selected from key-frames usually suffer from the high dimensionality problem. Second, the features chosen from a key-frame is in fact local features. For a matching process that is targeting at measuring the similarity among a great number of video clips, the key-frame based matching method is not really feasible because the information used to characterize the relationships among consecutive frames is not taken into account. In order to overcome these drawbacks, we propose an object-based motion activity descriptor, which can exploit the spatio-temporal information of a video clip in the matching process. Basically, the proposed spatio-temporal features can support high-level semantic-based retrieval of videos in a very efficient manner. We make use of some spatio-temporal relationships among moving objects and then use them to support the retrieval task. In the retrieval process, we use the DCT to reduce the dimensionality of the extracted

high-dimensional feature. Using DCT, we can maintain the local topology of a high-dimensional feature. In addition, the energy concentration property of DCT allows us to use only a few DCT coefficients to represent the moving objects and their variations. Therefore, the transformation can make an accurate and efficient retrieval process possible.

The rest of the chapter is organized as follows. Section 5.2 presents an overview of the proposed scheme. Section 5.3 illustrates the methods used to characterize video segments. Section 5.4 describes the representation and matching of video sequences. Section 5.5 presents the experimental results. Section 5.6 draws conclusions and suggests avenues for future work.

5.2 Overview of the Proposed Scheme

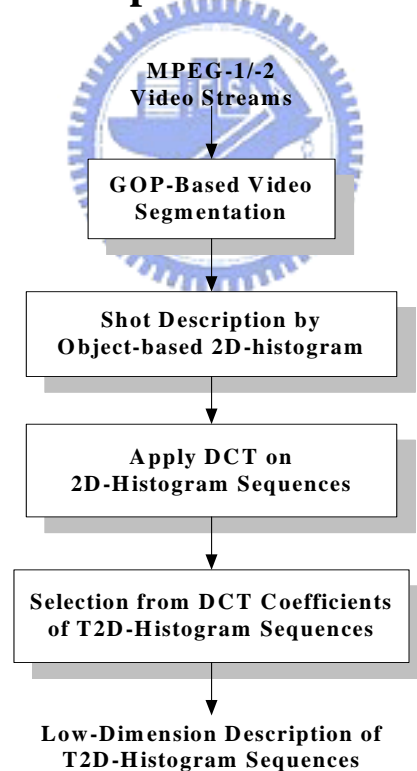


Fig. 5-1. An overview of extracting the proposed T2D-Histogram descriptor – compressed videos are parsed semantically and represented by reduced low-dimensional DCT coefficients

In this section, we shall provide an overview of the proposed video retrieval system.

Fig. 5-1 shows the flowchart of the proposed system. MPEG videos are efficiently segmented into shots using our previously proposed GOP-based video segmentation algorithm. This video segmentation algorithm checks video streams GOP-by-GOP rather than frame-by-frame. The actual shot boundaries are then determined at the frame level. After the process of shot segmentation, the next step is to execute an algorithm, which can generate an object-based motion activity description. The motion activity descriptor is able to describe moving objects in compressed videos. The features used by this motion descriptor are statistically computed by spatial and temporal distributions along the horizontal and vertical directions, respectively. The function of the descriptor is basically an encoder, which can encode video contents into high-level relational features. In order for maintaining high computational efficiency, we choose P-frames for motion activity analysis. Under these circumstances, a video clip can be represented by a set of motion activity descriptions of consecutive frames in the time domain. However, it is impractical to search a large video database using the time domain features. Therefore, we propose to apply DCT on the target frames and make them become lower dimensional in the frequency domain. Finally, we conduct an indexing process on the transformed DCT coefficients. As we mentioned before, due to the energy concentration property of DCT, we are able to represent the original moving objects in a most accurate and efficient way.

5.3 Characterization of Video Segments

In this section, we shall describe how to characterize a video segment so that it can be used to perform efficient video retrieval. We shall describe how to detect moving objects in a video segment in Section 5.3.1 and then discuss how to describe motion activity of a video segment in Section 5.3.2.

5.3.1 Moving Object Detection

For computational efficiency, motion information in P-frames is used for the detection of moving objects. In general, consecutive P-frames separated by two or three B-frames are still similar and would not vary too much. Therefore, it is reasonable to use P-frames as targets for moving objects detection. On the other hand, since the motion vectors estimated in MPEG-2 videos may not be 100% correct, one has to remove the noisy part before they can be used. For those motion vectors that are small in magnitude, we consider they are noises and should be removed. For the sake of computation speed, the average of motion vectors in those inter-coded macroblocks is computed and selected as the threshold for noise removal. After noisy motion vectors are filtered out, the motion vectors with similar magnitude and direction are clustered into a group by applying a region growing process with an morphological operator of 2x2 macroblocks. Thus, moving areas with size smaller than 4 macroblocks would be recognized as noises and be removed. Fig. 5-2 illustrates some examples of moving object detection in MPEG videos.

In our previous works [10][51], we have successfully detected moving objects in several kinds of videos such as tennis, traffic monitoring, news and football. Moving objects can be detected with an over 90% success rate when the camera is stationary. When the camera moves, camera motion such as pan or tilt should be estimated in advance before detecting moving objects. In our previous work, the precision is about 83% when the camera moves. However, the recall is still higher than 90%. Examples of moving object detection using our previous algorithm are demonstrated in Fig. 5-2. Video shots shown in Figs. 5-2(a) – 5-2(c) are extracted from an MPEG-7 testing dataset, and the shot of tennis competition in Fig. 5-2(d) is recorded from the Star-Sports TV-channel. Based on the results shown in Fig. 5-2, it is obvious that all moving objects are successfully detected.

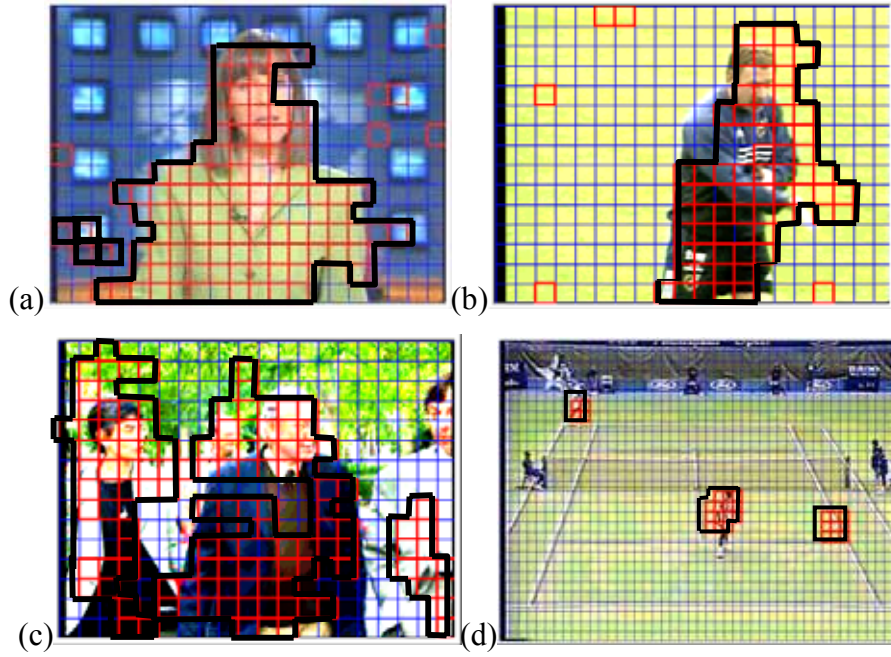


Fig. 5-2. Demonstration of moving object detection (a) anchor person (b) football (c) walking person (d) tennis competition.

5.3.2 Describing Motion Activity in a Video Segment

In this section, we shall elaborate how to describe object-based motion activity in a video segment. After moving objects are detected, the spatial distribution of them is characterized using the statistics derived from the 2D-histogram. A 2D-histogram for each P-frame consists of an X-histogram and a Y-histogram. The horizontal axis of the X-histogram (Y-histogram) is the quantized X-coordinate (Y-coordinate) in a frame. The X- and Y-coordinates are quantized into β bins, which should be moderate and be adaptive to various content types of MPEG videos. Thus, β should be related to the frame resolution and the threshold of object size based noise filtering, and is defined by

$$\beta = \min\left(\frac{R_{row}}{\sqrt{S}}, \frac{R_{column}}{\sqrt{S}}\right), \quad (5-1)$$

where $R_{row} \times R_{column}$ is the resolution of frame size in terms of macroblocks and S is the size of morphological operator in noise filtering. The decision of β will be verified by the simulated results in Section 5.3. Initially, the object size is estimated before bin

assignment. If an object is larger than the predefined unit size ($frame-size/\beta^2$), then it is normalized and accumulated according to the following equation:

$$Bin_{i,j}^X = \sum_{r=1}^{Obj} Acc_{i,j,r}^X, \quad (5-2)$$

$$where \ Acc_{i,j,r}^X = \begin{cases} 1, & \text{if object size} \leq \frac{1}{\beta^2} \text{ frame size} \\ \frac{\text{size of object } \gamma}{\text{frame size}} * \beta^2, & \text{otherwise} \end{cases}$$

where $Bin_{i,j}^X$ denotes the j^{th} bin of an X-histogram in frame i , $Acc_{i,j,r}^X$ means the accumulated value of the j^{th} bin of *object r* in *frame i* for an X-histogram, and Obj is the number of objects in frame i . Fig. 5-3 shows how a 2D-histogram is computed, with the number of histogram bins set to four. In the example, two objects with sizes of three units and four units are present in the frame. To obtain the X-histogram, the size of each object is assigned to a histogram bin based on its centroid (indicated by the symbol “*”) on the horizontal axis. For example, the football player of size three is assigned to Bin 1 and the basketball player of size four is assigned to Bin 3 in the X-histogram. Similarly, in the Y-histogram, Bin 2 is increased by 3 and that of Bin1 is increased by 4.

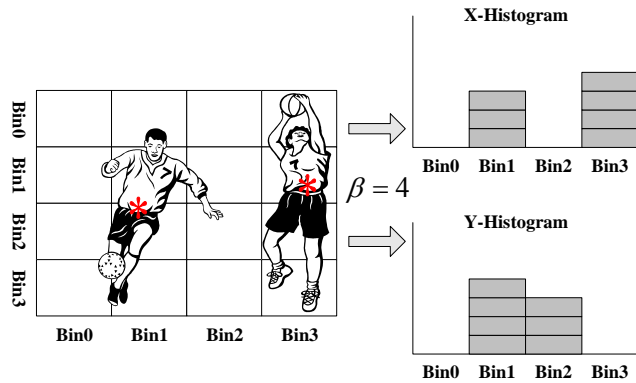


Fig. 5-3. Demonstration of the computation of 2D-histogram

Using the proposed 2D-histogram, the spatial distributions among moving objects are approximately described since each moving object is assigned to the histogram bin

based on its centroid. Objects that belong to the same coordinate interval are grouped into the same bin, and thus the distance between object groups can be specified as the distance between the associated bins.

5.4 Video Sequence Matching

After video segments are characterized by the descriptor of object-based 2D-histogram, temporal relationships among the moving objects have to be described. In order to characterize the temporal relationships among moving objects, a few DCT coefficients of the transformed time sequence are used to represent the variations of original objects among consecutive frames. A brief review of DCT will be elaborated in Section 5.4.1. Section 5.4.2 will describe how to represent a video sequence. The similarity metric that can be used to measure the degree of similarity will be discussed in Section 5.4.3.

5.4.1 Discrete Cosine Transform

The DCT (Discrete Cosine Transform) is a powerful tool that has been extensively used in many data compression applications. The DCT of a finite length sequence often has its coefficients more highly concentrated at low indices than other transforms do [67]. It has been proven in [68] that the approximation capability of DCT is much better than that of other approximation methods. Therefore, we shall use the DCT to characterize the temporal variations among moving objects in a video sequence.

5.4.2 Representation of Video Sequences

In this section, we shall describe how to characterize the temporal variations among moving objects exploiting the DCT. The algorithm that can be exploited to generate video sequence representation is as follows:

Video Sequence Representation Algorithm

Input: Consecutive P-frames $\{P_1, P_2, P_3, \dots, P_N\}$

Output: Sequences of representative DCT coefficients $[Z_{f,j}]$, where $f \in [1, \alpha]$ and $j \in [1, \beta]$

Procedure:

1. For each P-frame P_i

Detect moving objects by clustering macroblocks that have similar motion vector magnitudes and similar motion directions.

2. For each object $Obj_{i,r}$, where i and r denote the r th object in the i th P-frame;

Compute the centroid and the object size in the unit of macroblocks.

3. Set the number of histogram bins to β

4. For each P-frame P_i ,

Compute the X-histogram and the Y-histogram according to the horizontal and vertical position of the objects, respectively.

5. For each sequence of histogram bins $[Bin_{t,j}^Z]$, where $t \in [1, N]$, $j \in [1, \beta]$ and $Z \in \{X, Y\}$

Compute the transformed sequence $[Z_{f,j}]$ using the Discrete Cosine Transform

$$Z_{f,j} = C(f) \sum_{t=1}^N Bin_{t,j}^z \cos\left(\frac{(2t+1)f\pi}{2N}\right), \text{ where } f \in [1, N]$$

6. Set the number of DCT coefficients to α .

7. For β transformed sequences $[Z_{f,j}]$ of DCT coefficients,

Select the DC coefficient and $(\alpha-1)$ AC coefficients to represent a transformed sequence.

8. Generate the β reduced low-dimensional sequences $[Z_{f,j}]$, where $f \in [1, \alpha]$ and $j \in [1, \beta]$

Fig. 5-4 is the graphical representation of the above algorithm. For each P-frame, the feature of the object-based motion activity is described by a 2D-histogram, in which the spatial distribution of moving objects in horizontal and vertical direction are characterized by the bin values of the X-histogram and the Y-histogram, respectively. Therefore, a video sequence can be represented by a sequence of 2D-histogram with $2N\beta$ dimensions, where N is the number of P-frames in a video sequence and β is the number of bins in X-histogram and Y-histogram. In order to reduce the dimensionality of the feature space, DCT is exploited to transform the 2D-histogram of the original video sequence into the frequency domain. The value of the j^{th} bin $Bin_{i,j}^X$ of X-histogram ($Bin_{i,j}^Y$ of Y-histogram) in the i th P-frame is considered to be a signal in time i , and thus the corresponding j^{th} X-histogram bin in the consecutive N P-frames is regarded as a time signal $x_j = [Bin_{t,j}^X]$ ($y_j = [Bin_{t,j}^Y]$ of the Y-histogram), where $t = 1, 2, 3, \dots, N$. The N -point DCT of a signal x_j is defined as a sequence $X = [X_{f,j}]$, $f = 1, 2, 3, \dots, N$ as follows:

$$X_{f,j} = C(f) \sum_{t=1}^N Bin_{t,j}^X \cos\left(\frac{(2t+1)f\pi}{2N}\right), \quad (5-3)$$

$$C(0) = \sqrt{\frac{1}{N}} \quad \text{and} \quad C(f) = \sqrt{\frac{2}{N}}, \quad f = 1, 2, \dots, N-1$$

where N is the number of P-frames and $j \in [1, \beta]$. Eq. (5-3) indicates that a video sequence is represented by β sequences of DCT coefficients restricted by the number of bins in the histogram. It means that temporal variations among original objects in the successive P-frames are characterized by β sequences of DCT coefficients in frequency domain.

It is well known that the first few low-frequency AC terms together with the DC term will suffice for the need. Therefore, for easy computation we only choose these

terms to represent a video sequence instead of selecting all coefficients. However, to select an appropriate amount of AC coefficients is always a crucial issue. Since the selection of coefficients is an ill-posed problem, we shall discuss this problem in the experiments.

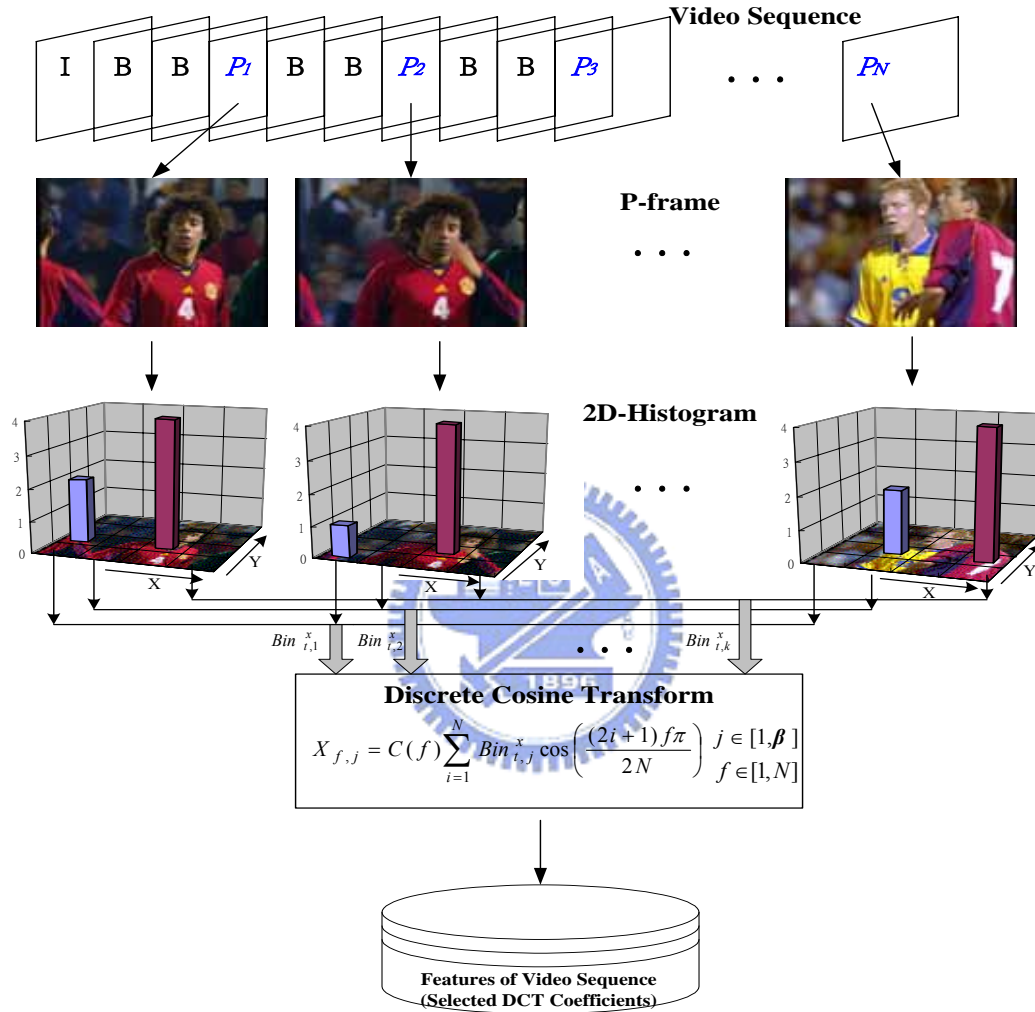


Fig. 5-4. Video sequences are characterized by the object-based T2D-Histogram descriptor and further represented by reduced low-dimensional DCT coefficients

5.4.3 Choice of Similarity Measure

A very important property of Parseval's theorem is that the Euclidean distance between DCT transformed signals is able to maintain the local topology. Therefore, for matching between video sequences we employ the modified Euclidean distance as the metric. Let $[W_f^X]$ and $[H_f^X]$ be two finite point sets of X-histogram ($[W_f^Y]$ and

$[H_f^Y]$ of the Y-histogram). Then the modified Euclidean distance between two video sequences w and h is defined as

$$\begin{aligned} Dist_x(w, h) &= Min \left(\begin{array}{l} Dist_x(W, H), Dist_x(W, shr(1, H)), \\ Dist_x(W, shr(2, H)), \dots, Dist_x(W, shr(\beta - 1, H)) \end{array} \right) \\ Dist_y(w, h) &= Min \left(\begin{array}{l} Dist_y(W, H), Dist_y(W, shr(1, H)), \\ Dist_y(W, shr(2, H)), \dots, Dist_y(W, shr(\beta - 1, H)) \end{array} \right) \end{aligned} \quad (5-4)$$

where $Dist_x(W, H) = \sum_{j=1}^{\beta} \sum_{f=1}^{\alpha} (W_{f,j}^X - H_{f,j}^X)^2$, $Dist_y(W, H) = \sum_{j=1}^{\beta} \sum_{f=1}^{\alpha} (W_{f,j}^Y - H_{f,j}^Y)^2$ and W and H are the transformed signals of w and h , respectively. In Eq. (5-4), j denotes the j th histogram bin, f represents the f th coefficient and α denotes the number of selected DCT coefficients. $shr(n, H)$ is a bin-rotating function which rotates the β histogram bins to the right n times in a cyclic way. For example, $shr(1, H)$ shifts the first $(\beta-1)$ bins 1 time to the right and the last bin rotates from the β th bin to the 1st bin. Using the distance metric with function $shr(n, H)$, two video sequences will be regarded as similar when they are spatially and temporally similar. If the function $shr(n, H)$ were not employed in the distance function, a shot A with objects positioned in the left and a shot B with objects positioned in the right would be regarded as dissimilar because the peak bins of Shots A and B are in the left and right, respectively and thereby the distance between A and B would be very large.

To further address the overall moving trend of objects within a video sequence, $Dist_x(w, h)$ and $Dist_y(w, h)$ are weighted adaptively based on the average motion vector magnitudes derived from the x- and y-directions. Under these circumstances, the total distance $Dist_{total}(w, h)$ between two video sequences w and h can be defined as

$$Dist_{total}(w, h) = WT_H \cdot Dist_x(w, h) + WT_V \cdot Dist_y(w, h) \quad (5-5)$$

$$WT_H = \frac{1}{N} \sum_{i=1}^N \frac{MV_{i,H}}{MV_{i,H} + MV_{i,V}}, \quad WT_V = 1 - WT_H,$$

where WT_H is the weight of the X-histogram (WT_V of Y-histogram), N is the number

of P-frames, and $MV_{i,H}$ and $MV_{i,V}$ are the average motion vector magnitudes of the X-component and Y-component, respectively, of the inter-coded macroblocks in the i^{th} P-frame. The reason why the analysis on object motion is split into two independent directions is as follows. It is well known that a camera would normally pan or tilt to catch moving objects in a scene. This act will in fact result in the situation that the global motion is mainly horizontal (vertical) when most active regions move in the horizontal (vertical) direction. Therefore, it is feasible to use the dominant moving trend to measure the video similarity. For example, we can discriminate between baseball and football videos using the above mentioned similarity metric because most players in a baseball game run vertically and the camera tilts to track them or the baseball, while players in a football game primarily run horizontally and the camera pans to track significant events.

5.5 Experimental Results and Discussions

In order to show the effectiveness of the proposed method, we simulated the color video sequence matching algorithm by MPEG-7 test dataset [69], which includes various programs such as documentaries, news, sports, entertainment, education, scenery, interview, etc and consists of 1173 shots. In the test dataset, the degree of strength of the motions in these shots ranged from low, medium to high, and the size of moving objects were classified as either small, medium or large. The anchorperson shots and interview shots (API shots) are typical low activity shots with small-range motions of mouth and head. The close-up tracking shots (CUT shots) are medium or large activity shots with medium or large-area moving foreground objects. The walking person shots (WP shots) are typical medium activity shots with medium or large motion areas. The aims of the experiments were to (1) evaluate the retrieval performance using different number of DCT coefficients; (2) analyze the degree of

accuracy when distinct number of histogram bins was used in the retrieval process; and (3) evaluate the retrieval performance of the proposed object-based motion activity descriptor. To evaluate the performance of the above three issues, precision and recall were used as the metrics to measure the performance of the proposed retrieval system. Recall and precision were defined as follows:

$$Recall = \frac{\|Retrieve(q) \cap Relevant(q)\|}{\|Relevant(q)\|}, \quad Precision = \frac{\|Retrieve(q) \cap Relevant(q)\|}{\|Retrieve(q)\|}, \quad (5-6)$$

where “*Retrieve(q)*” means the retrieved video sequences that corresponded to a query sequence q ; “*Relevant(q)*” denotes all video sequences in the database that were relevant to a query sequence q and $\|\cdot\|$ indicates the cardinality of the set. *Recall* was defined as the ratio of the number of retrieved relevant video sequences to the total number of relevant video sequences in the video database, and *Precision* was defined as the ratio of the number of retrieved relevant video sequences to the total number of retrieved video sequences. In the following subsections, we shall elaborate on how to determine some important thresholds that will be used in the experiments and report the retrieval performance of the proposed system.

5.5.1 Selecting Appropriate Number of DCT Coefficients

In the experiments, we used four shot classes to test the performance of our algorithms. Among these test videos, the shots of the Close-Up Tracking (CUT) and the Walking Person (WP) were with high degree of motion. The shots covered in the Bicycle Racing (BR) and the Anchor Person (API) were with medium degree of motion and low degree of motion, respectively. Figs. 5-5(a) – 5-5(d) show the examples of these four shot types, with key-frames sampled per 40 frames. To evaluate the effect when different number of DCT coefficients was used in the retrieval process, the number of DCT coefficients, α , including the DC and the first $(\alpha-1)$ AC coefficients, was varied and tested under the condition that the number of

histogram bins, β , was set to 8. β was set to 8 because in the test dataset the resolution of frame size in terms of macroblocks was 20x15 in SIF 320x240 format. The descriptors D , the X-histogram, the Y-histogram, the 2D-histogram and the weighted 2D-histogram were independently used.



Fig.5- 5. Examples of the Close-Up (CUT), Bicycle Racing (BR), Walking Person (WP) and Anchorperson and Interview (API) shots

Figs. 5-6(a) – 5-6(d) show the retrieval performance using four different types of shots, CUT, BR, WP and API, respectively. The four curves shown in the figures corresponded to four descriptors, which had distinct number of DCT coefficients ($\alpha=1$, $\alpha=2$, $\alpha=3$ and $\alpha=5$). The horizontal axis denotes recall and the vertical axis denotes precision. Table 5-1 compared the performance among distinct settings of α .

“Rank” refers to the order of retrieval performance of recall-precision pairs and the first two ranks were listed for each descriptor measured by using different setting of α . The retrieval performance in the recall-precision pair with $\alpha=2$ in the CU and BR shots was better than that obtained with other settings. Although the setting of $\alpha=1$ yielded better retrieval than $\alpha=2$ in the WP shot, the performance obtained by setting $\alpha=2$ was still in the second best. For the API shots, the setting $\alpha=5$ was the best in terms of retrieval and the settings $\alpha=3$ and $\alpha=2$ were the second best as shown in Figs. 5-6(a) – 5-6(b) and Figs. 5-6(c) – 5-6(d), respectively.

Table 5-1. Performance using distinct α and four feature descriptors ($\beta = 8$)

Shot Type Descriptor		Close-Up Tracking (CUT)	Bicycle Racing (BR)	Walking Person (WP)	Anchor Person (API)
X- Histogram	Rank #1	2	2	1	5
	Rank #2	3	3	2	3
Y- Histogram	Rank #1	2	2	1	5
	Rank #2	1	3	2	3
2D - Histogram	Rank #1	2	2	1	5
	Rank #2	3	3	2	2
Weighted 2D - Histogram	Rank #1	2	2	1	5
	Rank #2	3	3	2	2

To evaluate the overall performance obtained using different numbers of DCT coefficients, the retrieval performance $P_{\lambda_{NDC}}$ for different α was determined by

$$P_{\lambda_{NDC}} = \sum_{i=1}^{|D|} \sum_{j=1}^{|Clips|} \frac{\rho}{Rank_{i,j}^{\lambda_{NDC}}} \quad (5-7)$$

where “NDC” denotes the “Number of DCT Coefficients”; ρ is the total number of different α settings in the experiment and $Rank_{i,j}^{\lambda_{NDC}}$ is the ranking of the retrieval performance for the shot of type j with $\alpha = \lambda_{NDC}$, using descriptor i . When $P_{\lambda_{NDC}}$ was larger, the performance obtained with $\alpha = \lambda_{NDC}$ was better. From the curves shown in Figs. 5-6(a) – 5-6(d), it is clear that P_2 can be computed and its value was larger

than other P values. This outcome means when $\alpha=2$, the retrieval result was the best. Hence, the experimental results imply that two DCT coefficients are enough for similarity measurement of video segments. This indicates the DC coefficient and the lowest-frequency AC coefficient will suffice.

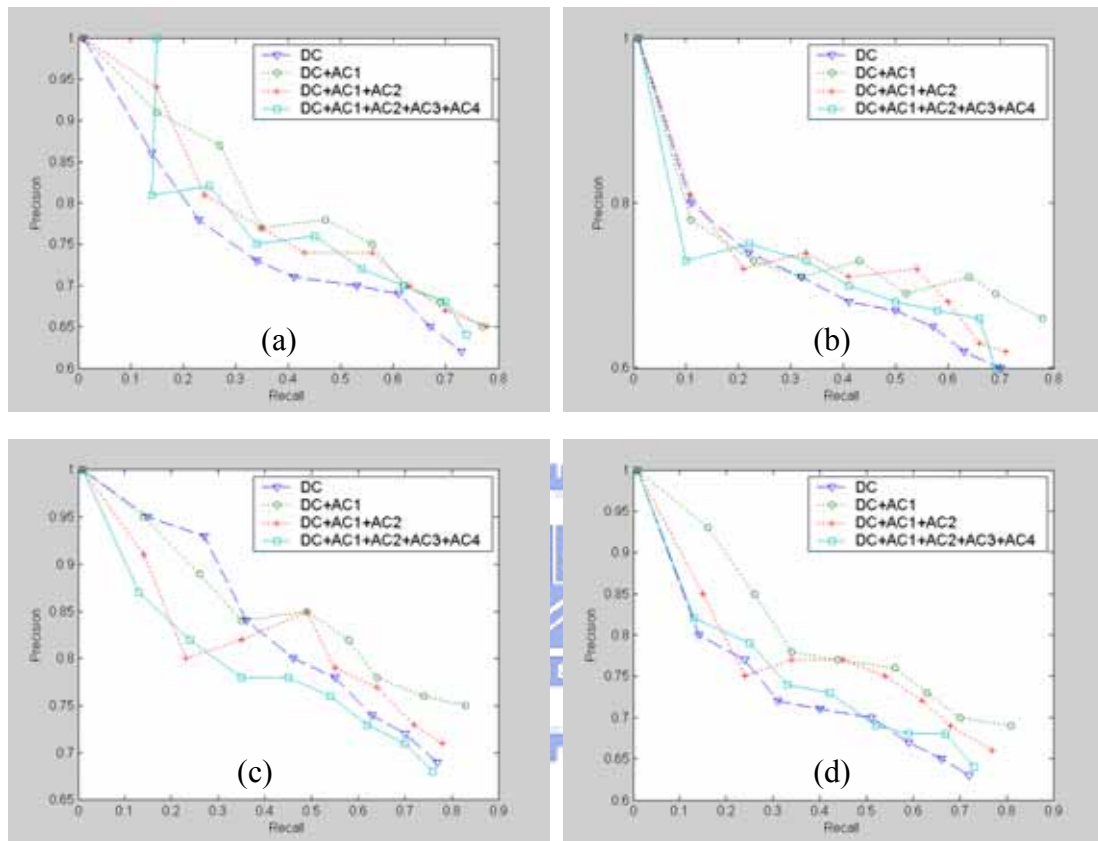


Fig. 5-6. Average retrieval performance with different descriptors ($\beta = 8$, $\alpha \in [1, 5]$)
 (a) X-histogram (b) Y-histogram (c) 2D-histogram (d) Weighted 2D-histogram

5.5.2 Choosing an Appropriate Motion Activity Descriptor

In order to determine an appropriate motion activity descriptor, we changed the value of β from 4 to 10, each time with an increment of 2. Figs. 5-7(a)-5-7(d) show, respectively, the performance of the recall-precision pair corresponding to $\beta=4$, $\beta=6$, $\beta=8$, and $\beta=10$. Table 5-2 illustrates the performance calculated by using four different number of histogram bins ($\beta=4, 6, 8$, and 10). In most cases, the descriptor adopted weighted 2D-histogram outperformed other types of descriptors. In order to

quantitatively compute the performance, we used a metric, P_{λ_D} , to measure the retrieval results,

$$P_{\lambda_D} = \sum_{i=1}^{|\beta|} \sum_{j=1}^{|Clips|} \frac{|D|}{Rank_{i,j}^{\lambda_D}} \quad (5-8)$$

where $|\beta|$ denotes the total number of distinct settings of β ; $|D|$ represents the number of testing descriptors; $Rank_{i,j}^{\lambda_D}$ is the retrieval performance ranking of the shot of type j with the i th β parameter setting and the descriptor $D = \lambda_D$. Based on the results calculated by Eq. (5-8), we chose the weighted 2D-histogram descriptor as the motion activity descriptor for all the experiments conducted in this work.

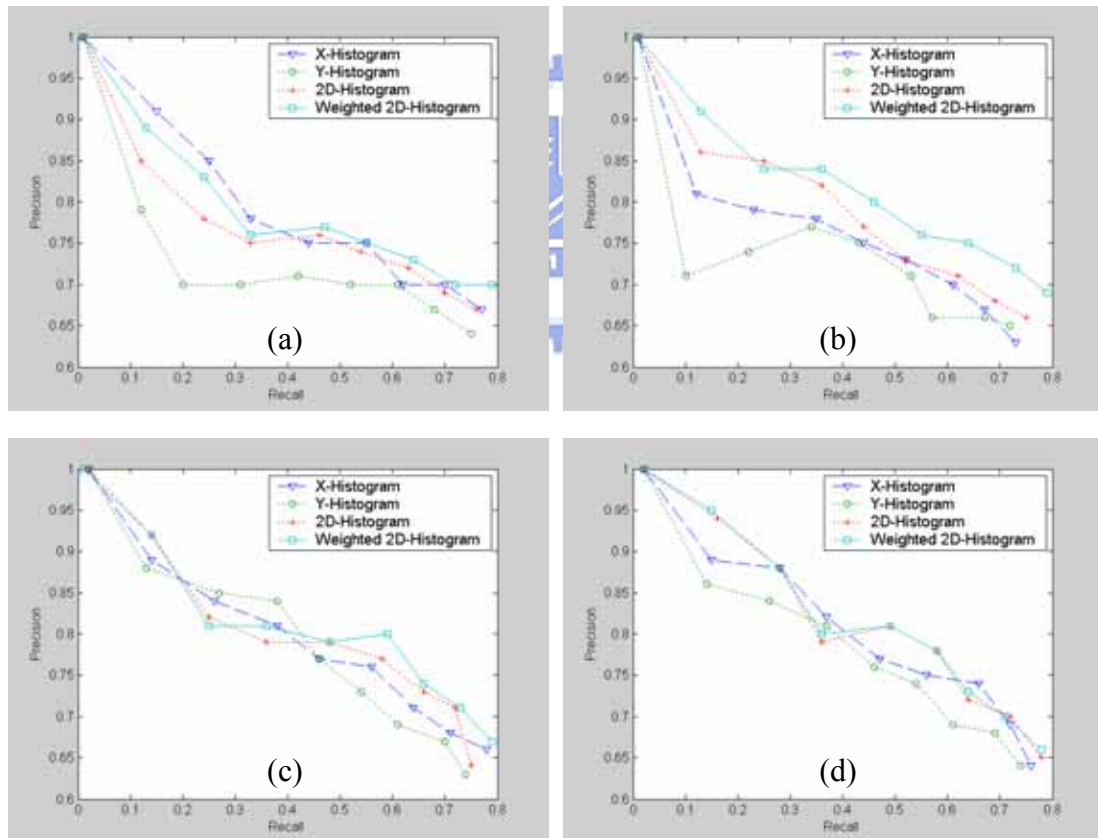


Fig. 5-7. Average retrieval performance ($\alpha=2$) with different number of bins (β)
(a) $\beta = 4$ (b) $\beta = 6$ (c) $\beta = 8$ (d) $\beta = 10$

Table5- 2. The performance obtained of four descriptors with different β ($\alpha = 2$)

Shot Type β Setting		Close-Up Tracking (CUT)	Bicycle Racing (BR)	Walking Person (WP)	Anchor Person (API)
$\beta = 4$	Rank #1	X	X	W-2D	W-2D
	Rank #2	W-2D	W-2D	X	2D
$\beta = 6$	Rank #1	W-2D	Y	X	W-2D
	Rank #2	X	W-2D	W-2D	2D
$\beta = 8$	Rank #1	X	W-2D	W-2D	W-2D
	Rank #2	W-2D	2D	2D	2D
$\beta = 10$	Rank #1	W-2D	W-2D	W-2D	X
	Rank #2	2D	2D	2D	2-2D

X: X-Histogram Y: Y-Histogram 2D: 2D-Histogram W-2D: Weighted 2D-Histogram

5.5.3 Determining the Best Number of Histogram Bins

In this section, we shall verify the decision of the number of histogram bins β . Therefore, we evaluated the performance by using different number of histogram bins, which ranged from 4, 6, 8 to 10. The recall-precision pair corresponding to each β setting was depicted in Fig. 5-8, and the ranking of retrieval performance for each shot type was illustrated in Table 5-3.

It is obvious that the retrieval performance at $\beta = 8$ decided by Eq.(5-1) was better than other settings and the worst case was when $\beta = 4$. The experimental results reveal that the number of histogram bins should be moderate, because fewer histogram bins correspond to a less precise description of the variation in spatial distribution. In contrast, when the number of histogram bins was too large, the descriptor would be extremely responsive to the slight changes. Under this circumstance, the distance obtained from excessive number of bins between two similar shots is relatively high such that these two shots would be regarded as dissimilar.

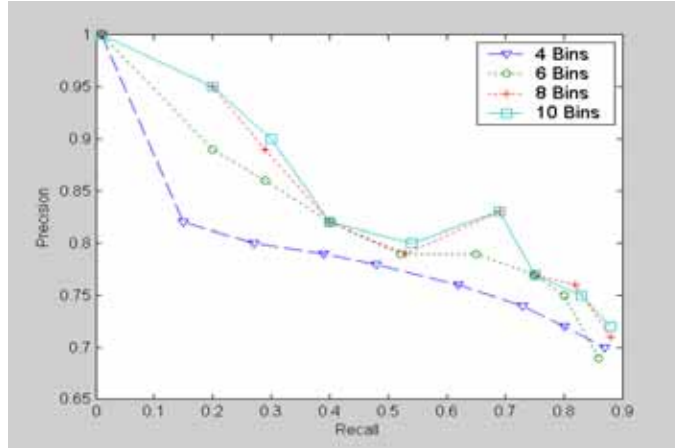


Fig. 5-8. Average retrieval performance with parameters:
 $\alpha=2$, D : weighted 2D-histogram, $\beta \in \{4,6,8,10\}$

Table 5-3. Comparison of performance using different numbers of histogram bins (β)

Shot Type Performance	Close-Up (CUT)	Bicycle Racing (BR)	Walking Person (WP)	Anchor Person (API)
Rank #1	6	8	8	8
Rank #2	10	10	10	10
Rank #3	8	6	6	6
Rank #4	4	4	4	4

5.5.4 Evaluation of Retrieval Performance

After the number of DCT coefficients, the number of histogram bins and the descriptor type are determined, we shall evaluate the overall retrieving accuracy of the proposed system. The ground truth and the overall performance corresponding to the four shot classes are shown in Table 4. In the experiment, each shot in these four classes was used as a query shot. The top 30 similar shots were returned as a query result for evaluating retrieval performance. Finally, the respective average recall and precision for each class were computed. The recall of these four kinds of shots exceeded 80% in which the recall of BR, CUT and API were higher than 86%. The worst result was obtained by testing the API shots, with the precision of 78%. On the other hand, although the precision of the API shots was under 80%, the precision of the CUT, BR, and WP all exceeded 80%. From Table 5-4, the overall average recall

and average precision were 86% and 81%, respectively.

For performance comparison, we have performed the same experiments using the algorithms of motion-based run-length descriptor (RLD) and shot activity histogram (SAH) provided by MPEG-7 [70]. Fig. 5-9 shows the precision versus recall performance of RLD, SAH and T2D-Histogram. The T2D-Histogram descriptor had performance gain over RLD of 45% in API shots, 30% in the CUT shots, 34% in the WP shots and 35% in the BR shots. Also, the T2D-Histogram had performance gain over SAH of 11% in the API shots, 7% in the CUT shots, 20% in the WP shots and 21% in the BR shots. In average, the T2D-Histogram descriptor had 35% and 15% performance gains over the RLD and SAH, respectively. The experimental results using extensive test videos show that the proposed T2D-Histogram outperforms RLD and SAH in MPEG-7 in the performance of video similarity retrieval.

Table 5-4. Retrieval performance using the T2D-Histogram descriptor

Clips Performance	Close-Up Tracking (CUT)	Bicycle Racing (BR)	Walking Person (WP)	Anchor Person (API)
Ground-Truth Video shots	162	47	239	152
Recall	88%	87%	80%	86%
Precision	80%	84%	81%	78%
Average Recall		Average Precision		
86%		81%		

Examples of the query results were demonstrated in Figs. 5-10 – 5-14, in which the top 20 similar shots for CUT, BR, WP and API shots were listed, respectively. In Fig. 10, most retrieved shots included large objects with significant motion belonged to the CUT shots. However, due to camera motion, some shots were mistakenly detected. For example, the full-court shots of the football game like (4), (8) and (12) of Fig. 5-10 were retrieved due to the panning effect of the camera. As to the relevant shots, it

is worth noticing that the major objects in these shots, such as (3), (18) and (19) of Fig. 5-10, had similar size with the object covered in the query although they had different colors. The reason why these shots could still be detected was due to their similarity with the objects in the query visually and semantically. When comparing with color-based methods such as color histogram, these shots with distinct dominant colors but semantically related cannot be retrieved.

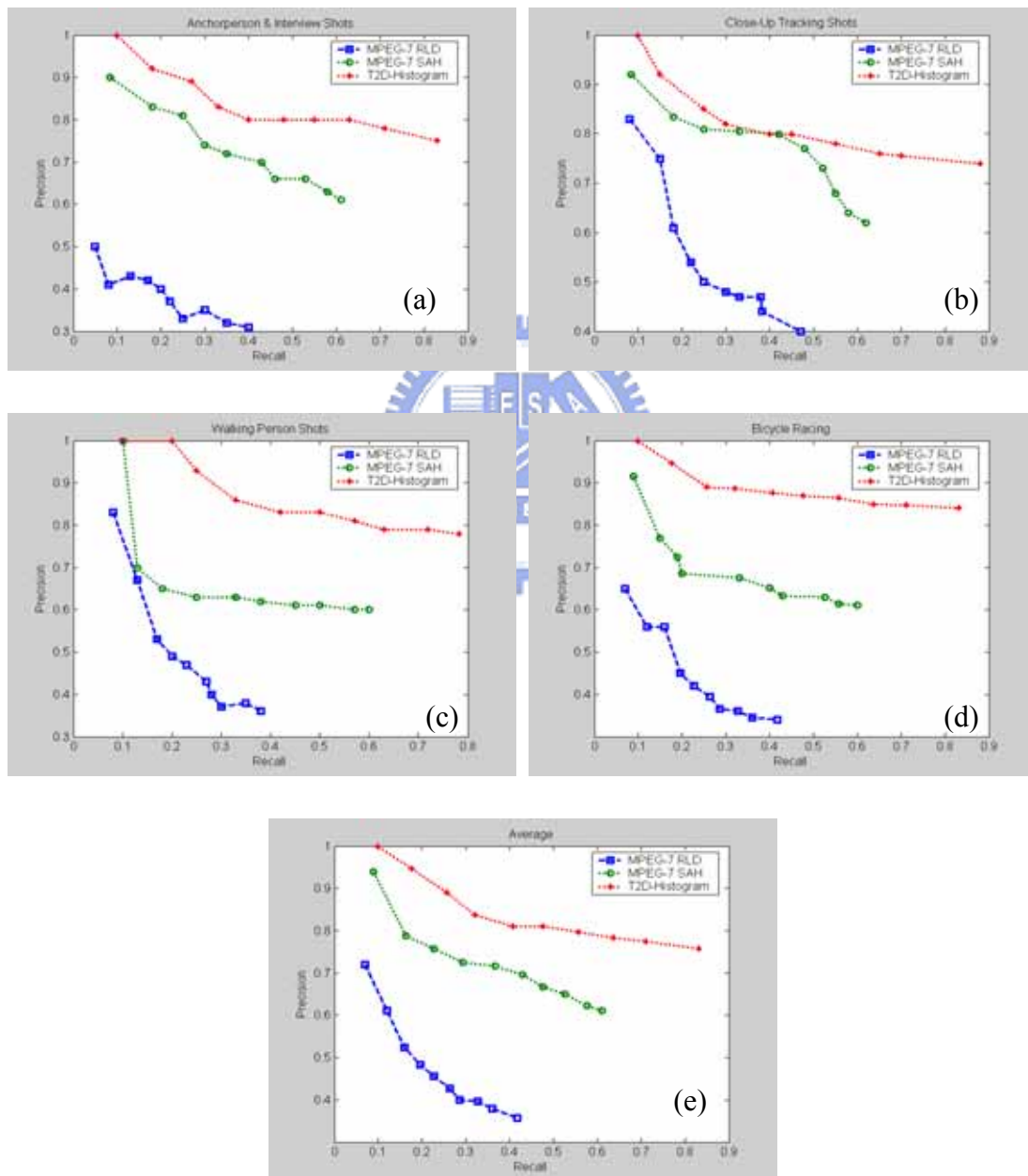


Fig. 5-9. Retrieval performance of the four shot classes (a) API Shots (b) CUT Shots (c) WP Shots (d) BR Shots and (e) Average

In Fig. 5-11, the retrieval performance of the BR shot was quite good and most retrieved video segments were similar to the query due to the particular motion of the rider(s). In Fig. 5-12, most retrieved video segments had a few medium-size moving objects. Some video segments were mistakenly detected, such as (10) and (14) of Fig. 5-12. These shots were retrieved due to the reason that the complex background was detected as several medium-size objects with a moving camera. In Fig. 5-13, most retrieved video segments included one large object with low motion, and so interview shots were also retrieved such as the shots (6), (8), (13), (16) and (20) of Fig.5-13. An example of false detection can be found in (12) of Fig.5-13, wherein some medium-size objects moved near to each other and so were incorrectly detected as a single large moving object.

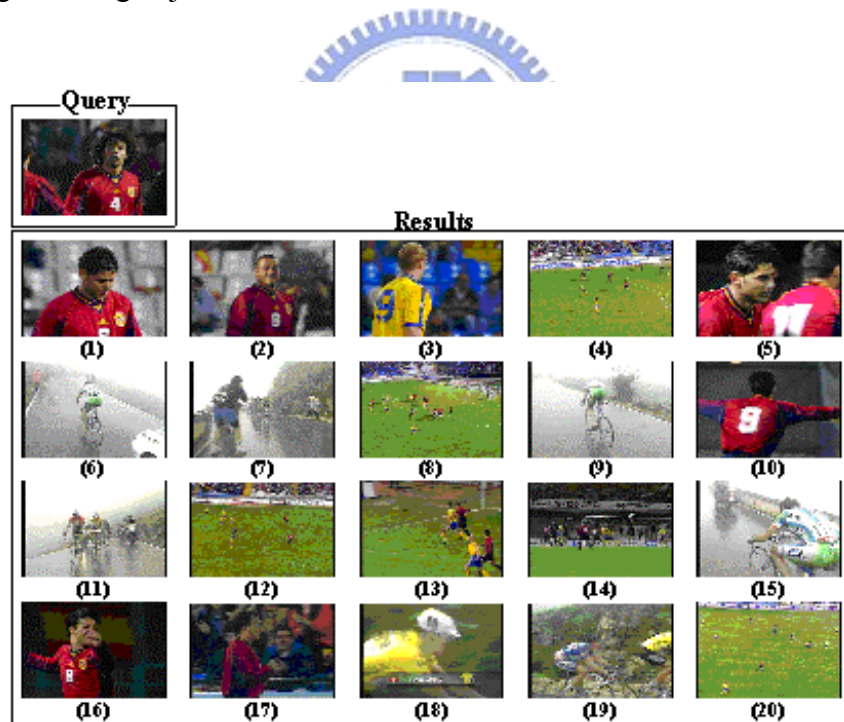


Fig. 5-10. Demonstration of the query result for a CUT shot

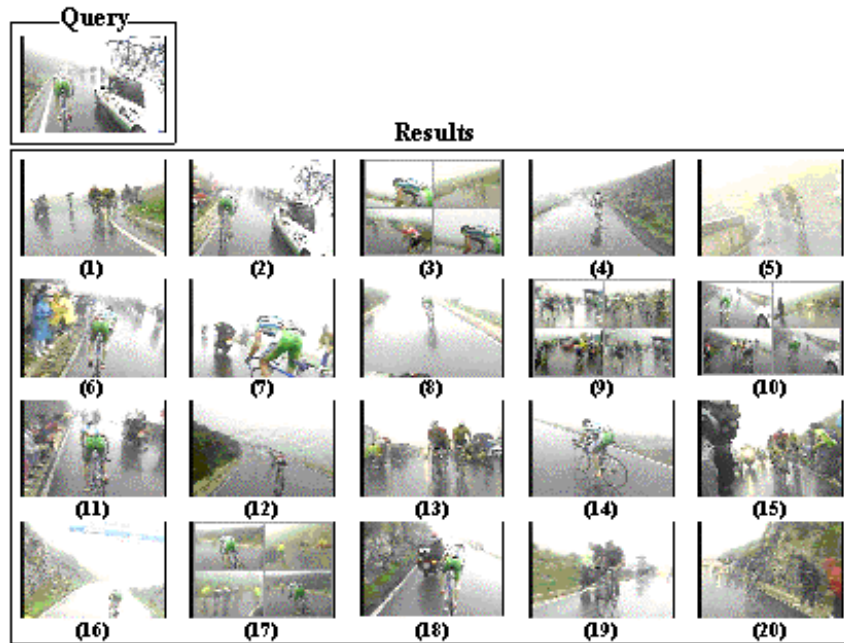


Fig. 5-11. Demonstration of the query result for a BR shot



Fig. 5-12. Demonstration of the query result for a WP shot

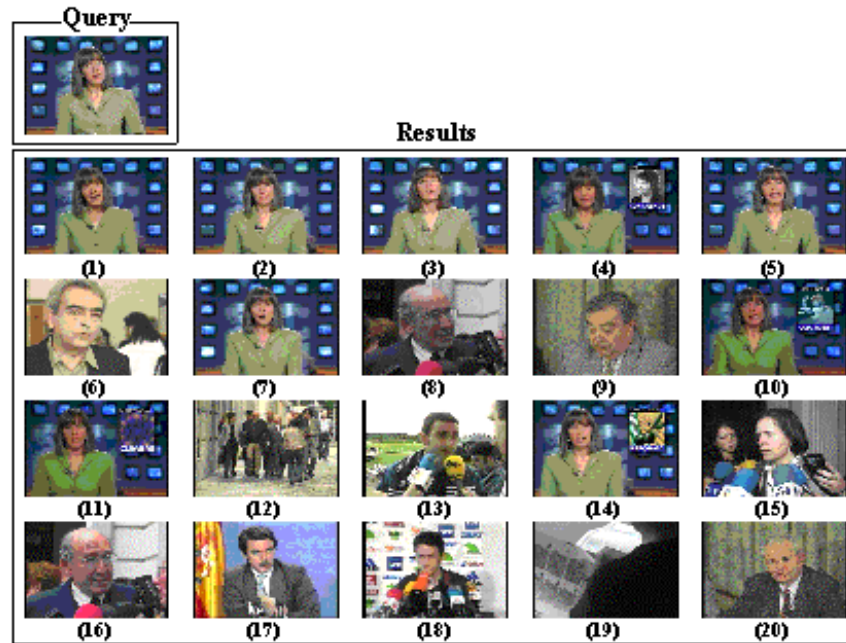


Fig. 5-13. Demonstration of the query result for an API shot

5.6 Summary

A novel framework of high-level video representation for video sequence matching has been proposed in this work. The proposed framework has two special features: 1) the proposed descriptor of object-based T2D-Histogram has exploited both spatial and temporal features of moving objects and characterized video sequences in a semantics-based manner; 2) the dimensionality of feature space has been reduced using DCT while characterizing the temporal variations among moving objects. Experimental results obtained using the extensive test dataset of MPEG-7 have demonstrated that a few DCT coefficients could suffice for representing a video sequence and also shown that the proposed T2D-Histogram descriptor was quite robust. Using this novel motion activity descriptor of object-based T2D-Histogram, one can perform video retrieval in an accurate and efficient way.

Chapter 6. Robust Video Similarity Retrieval Using Temporal MIMB Moments

6.1 Introduction

The tremendous growth in the number of digital videos has become the main driving force for developing automatic video retrieval techniques. Among different types of tools that can push the advancement of retrieval techniques, an efficient automatic content analyzer that can help execute correct browsing, searching and filtering of videos is a must. In order to achieve this goal, one has to make use of high-level semantic features to represent video contents. The need of representing high-level semantic features has motivated the emergence of MPEG-7, formally called the multimedia content description interface [53]. However, the methods that produce the specific features and the corresponding similarity measures represent the non-normative part of MPEG-7 and are still open for research and future innovation.

Usually, the high-level semantic features of video sequences can be inferred from low-level features. The low-level features can be color distribution, texture composition, motion intensity and motion distribution. Among different types of features that can be extracted from a video, motion is considered as a very significant one due to its temporal nature. In the literature, Divakaran et al. [54] used a region-based histogram to compute the spatial distribution of moving regions. The run-length descriptor in MPEG-7 [55] is used to reflect whether moving regions occurred in a frame. Aghbari et al. [56] proposed a motion-location based method to extract motion features from divided sub-fields. Peker et al. [57] calculated the average motion vectors of a P-frame and those of a video sequence to be the overall motion features. In addition to the above mentioned local motion features, Ngo et al. [58] and Tang et al. [15] proposed to use some global motion features to describe

video content.

In contrast to the motion-based features of individual frames, another group of researchers proposed to use spatio-temporal features between successive frames because these types of features are more abundant in the amount of information. Wang et al. [59] extracted features of color, edge and motion, and measured the similarity between temporal patterns using the method of dynamic programming. Lin et al. [60] characterized the temporal content variation in a shot using two descriptors - dominant color histograms of group of frames and spatial structure histograms of individual frames. Cheung and Zakhor [61] utilized the HSV color histogram to represent the key-frames of video clips and designed a video signature clustering algorithm for detecting similarities between videos. Dimitrova et al. [62] represented video segments by color super-histograms, which are used to compute color histograms for individual shots. Other works that fall into this category can be found in [63-66].

There are several drawbacks associated with the key-frame based matching process. First, the features selected from key-frames usually suffer from the high dimensionality problem. Second, the features chosen from a key-frame is in fact local features. For a matching process that is targeting at measuring the similarity among a great number of video clips, the key-frame based matching method is not really feasible because the information used to characterize the relationships among consecutive frames is not taken into account. In order to overcome these drawbacks, we propose a motion pattern descriptor, which can exploit the spatio-temporal information of moving blobs in a video shot in the matching process. Basically, the proposed spatio-temporal features can support high-level semantic-based retrieval of videos in a very efficient manner. We make use of some spatio-temporal relationships among moving blobs and then use them to support the retrieval task. In the retrieval

process, we use the DCT to reduce the dimensionality of the extracted high-dimensional feature. Using DCT, we can maintain the local topology of a high-dimensional feature. In addition, the energy concentration property of DCT allows us to use only a few DCT coefficients to represent the moving blobs and their variations. Therefore, the transformation can make an accurate and efficient retrieval process possible.

The rest of the chapter is organized as follows. Section 6.2 illustrates the methods used to characterize video segments. Section 6.3 presents the experimental results. Section 6.4 draws conclusions.

6.2 Characterization of Video Segments

6.2.1 Detecting Moving Blobs in MPEG Videos

For computational efficiency, motion information in P-frames is used for the detection of moving blobs. In general, consecutive P-frames separated by two or three B-frames are still similar and would not vary too much. Therefore, it is reasonable to use P-frames as targets for detecting moving blobs. On the other hand, since the motion vectors estimated in MPEG videos is for the purpose of compression and thus may not be 100% correct, one has to remove the noisy part before they can be used. In our previous work [71], a cascaded filter that is composed of a Gaussian filter followed by a median filter is exploited for noise removal. An example of noise filtering in MVF is demonstrated in Fig.6-1. The experimental results show that the precision is higher than 70% and the recall is higher than 80% and thus prove that the proposed spatial filter is effective to remove the noise in motion vector fields. To detect moving blobs in the filtered MVFs, macroblocks of similar MV magnitude and direction are clustered together by employing a region-growing method with an operator of 3x3 macroblocks.

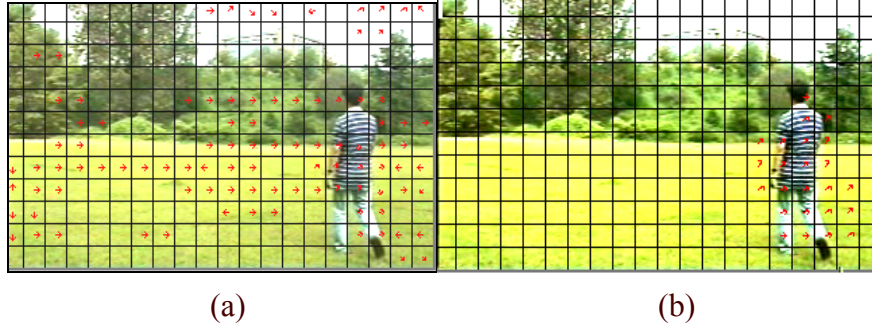


Fig. 6-1. Demonstration of MVF noise reduction (a) MVF without filtering; (b) MVF smoothing with a cascaded filter

6.2.2 MIMB Moments

The motion intensity of moving blobs (MIMB) is a descriptor for describing sketch features in a frame that contain moving regions with motion intensity. Rather than directly employing the MIMB obtained in a P-frame, a temporal filter using Gaussian filter with temporal window size of 5 frames is exploited to smooth MIMBs. To represent the spatial feature of MIMBs in a compact meaningful form, the moment invariants of MIMBs are computed. The use of moments for image analysis and object representation was inspired by Hu[72]. According to Hu's Uniqueness Theorem, the moment set $\{\mu_{pq}\}$ is uniquely determined by $MIMB(x,y)$ and conversely, $MIMB(x,y)$ is uniquely determined by $\{\mu_{pq}\}$. The central moment μ_{pq} computed from MIMB is defined by

$$\mu_{pq} = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} (x - \bar{x})^p (y - \bar{y})^q MIMB(x, y) \quad (6-1)$$

where $(p,q) = \{(0,2), (1,1), (2,0), (0,3), (1,2), (2,1), (3,0)\}$ and CxR is the frame size in terms of macroblocks. To select a meaningful subset of moment values that contain sufficient information to uniquely characterize the MIMBs, the seven moment invariants defined by Hu are employed and defined by

$$M_1 = \mu_{20} + \mu_{02} \quad (6-2)$$

$$M_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad (6-3)$$

$$M_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \quad (6-4)$$

$$M_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \quad (6-5)$$

$$M_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \quad (6-6)$$

$$M_6 = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \quad (6-7)$$

$$M_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \quad (6-8)$$

6.2.3 Representing Temporal Variations of MIMB Moments

In this section, we shall describe how to characterize the temporal variations among moving blobs exploiting the DCT. The algorithm that can be exploited to generate video sequence representation is as follows:

Video Sequence Representation Algorithm

Input: Consecutive P-frames $\{P1, P2, P3, \dots, PN\}$

Output: Representative truncated DCT coefficients $[X_{\Lambda,m}]$, where $\Lambda \in [1, \alpha]$.

Procedure:

1. For each P-frame P_i ,

Detect moving blobs using a cascaded filter followed by using morphological operations.



2. For each P-frame P_i ,

Compute Hu's seven moment invariants $\{M_{m,i}\}$ in the filtered MVF, where $m \in [1,7]$.

3. Compute the transformed sequence $[X_{f,m}]$ using the Discrete Cosine Transform

$$X_{f,m} = C(f) \sum_{t=1}^N M_{m,t} \cos\left(\frac{(2t+1)f\pi}{2N}\right), \text{ where } f \in [1, N]$$

4. For m transformed sequences, $[X_{f,m}]$ of DCT coefficients,

Truncate the number of DCT coefficients to α , which is composed of the DC coefficient and $(\alpha-1)$ AC coefficients to represent a transformed sequence.

5. Generate a feature vector $F(X_{\Lambda,1}, X_{\Lambda,2}, X_{\Lambda,3}, X_{\Lambda,4}, X_{\Lambda,5}, X_{\Lambda,6}, X_{\Lambda,7})$ for each

video segment, where $\Lambda \in [1, \alpha]$.

For each P-frame, the spatial feature of moving blobs in P-frames is represented by Hu's seven moment invariants. In order to characterize the temporal variations of moving blobs within successive frames, DCT is exploited to transform the MIMB moments of the original video sequence into the frequency domain. The value of the MIMB $M_{m,i}$ in the i th P-frame is considered to be a signal in time i , and thus the corresponding MIMB $M_{m,i}$ in the N P-frames is regarded as a time signal $x_m = [M_{m,t}]$, where $t = 1, 2, 3, \dots, N$. The N -point DCT of a signal x_m is defined as a sequence $X = [X_{f,m}]$, $f = 1, 2, 3, \dots, N$ as follows:

$$X_{f,m} = C(f) \sum_{t=1}^N M_{m,t} \cos\left(\frac{(2t+1)f\pi}{2N}\right), C(0) = \sqrt{\frac{1}{N}}, \text{ and } C(f) = \sqrt{\frac{2}{N}}, f = 1, 2, \dots, N-1, \quad (6-9)$$

where N is the number of P-frames and $m \in [1, 7]$. Eq. (6-9) indicates that a video sequence is represented by 7 sequences of DCT coefficients. It means that temporal variations among original objects in the successive P-frames are characterized by 7 sequences of DCT coefficients in frequency domain. It is well known that the first few low-frequency AC terms together with the DC term will suffice for the need. Therefore, for considering computation cost we only choose these terms to represent a video sequence instead of selecting all coefficients. However, to select an appropriate amount of AC coefficients is always a crucial issue. The experimental results imply that two DCT coefficients are enough for similarity measurement of video segments. This indicates the DC coefficient and the lowest-frequency AC coefficient will suffice.

6.3 Experimental Results

6.3.1 Choice of Similarity Measure

The similarity measure is for computing the similarity between a feature vector of a query video shot and a feature vector of a target video shot. To choose a similarity measure, in statistics we prefer a distance that for each of the components takes the variability of that variable into account when determining its distance from the center. Components with high variability should receive less weight than components with low variability. Therefore, a Mahalanobis distance is used as a similarity measure, which is defined as

$$D(F^q, F^t) = \left(\sum_{k=1}^n \left| \frac{F_k^q - F_k^t}{\sigma_k} \right|^2 \right)^{1/2}, \quad (6-10)$$

where F_k^q and F_k^t denote the k th components of a query feature vector F^q and a target feature vector F^t , respectively and n denotes the dimension of a feature vector. σ_k denotes the standard deviation of the k th component for feature vectors in the testing dataset.

6.3.2 Evaluation of Retrieval Performance

In order to show the effectiveness of the proposed method, we simulated the algorithm of video sequence matching by using MPEG-7 testing dataset [69] which includes various programs such as news, sports, entertainment, education, etc and consists of 1173 shots. The degree of strength of the motions in these shots ranged from low, medium to high, and the size of moving objects were classified as either small, medium or large. To evaluate the performance, precision and recall were used as the metrics to measure the performance of the proposed retrieval system. Recall and precision were defined as follows:

$$Recall = \frac{\|Retrieve(q) \cap Relevant(q)\|}{\|Relevant(q)\|}, \quad Precision = \frac{\|Retrieve(q) \cap Relevant(q)\|}{\|Retrieve(q)\|} \quad (6-11)$$

where “ $Retrieve(q)$ ” means the retrieved video sequences that corresponded to a query sequence q ; “ $Relevant(q)$ ” denotes all video sequences in the database that were

relevant to a query sequence q and $||\cdot||$ indicates the cardinality of the set.

In the experiments, we used three classes of shots to test the performance of our algorithms. Among these test videos, the shots covered in the Close-Up Tracking (CUT) and the Walking Persons (WP) were with high degree of motion and medium degree motion, respectively. The Anchorperson and Interview (IV) shots were with low degree of motion. Considering the sensitivity of the proposed descriptor to the size of moving blobs, the blob size ranges between small blobs of 2x2 macroblocks and large blobs of half or larger frame size. The 30 most relevant shots corresponding to every query were selected out of 1173 shots. In order to give a comparison, we also do the same experiments using the algorithm of motion-based run-length descriptor (RLD) and shot activity histogram (SAH) provided by MPEG-7.

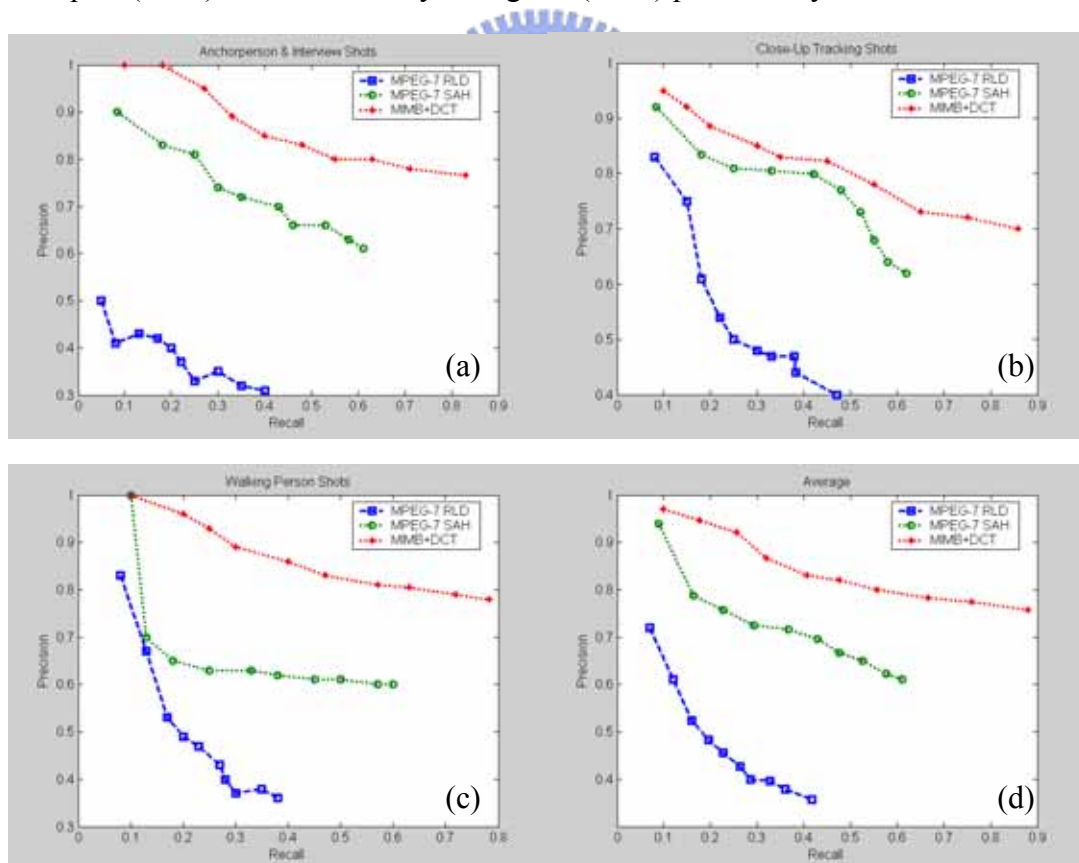


Fig. 6-2. Recall versus precision performance of the three shot classes (a) Interview Shots (b) Close-Up Tracking Shots (c) Walking Person Shots (d) Average

Fig. 6-2 shows the precision versus recall performance of the combination of RLD

in MPEG-7 and the proposed MIMB+DCT descriptor. The proposed descriptor yielded 49% average performance gain in the IV shots, 29% in the CUT shots and 37% in the WP shots over the RLD. Also, the proposed descriptor yielded 40% and 21% average performance gain in all the testing classes over the RLD and SAH in MPEG-7, respectively.

6.4 Summary

A novel framework of high-level video representation for video sequence matching has been developed in this work. The proposed framework has two special features: 1) the proposed temporal MIMB moments has exploited both spatial and temporal features of moving blobs and characterized video sequences in a high-level manner; 2) the dimensionality of feature space has been reduced using DCT while characterizing the temporal variations among moving blobs. Experimental results obtained using MPEG-7 testing dataset have demonstrated that a few DCT coefficients could suffice for representing a video sequence and also shown that the proposed motion-pattern descriptor was quite robust and efficient. Using this framework, one can perform video retrieval in an accurate and efficient way.

Chapter 7. Conclusions and Future Work

7.1 Contributions

In this thesis, we have proposed several object-based approaches in compressed videos for detecting semantic events, characterizing video shots, localizing superimposed closed captions, and structuring video content. We have several contributions as follows:

1. An effective object-based mechanism is proposed to detect semantic events in sports videos.
2. Two novel object-based motion pattern descriptors are proposed to characterize the spatio-temporal variations of moving objects in a video shot. These descriptors are not only in a compact form but are effective for video similarity retrieval.
3. A transformed feature employing DCT that is low dimensional but semantically meaningful is proposed for efficient content-based video retrieval.
4. A novel algorithm for detecting superimposed closed captions in compressed videos is proposed.
5. A font size filter is designed as the support for users to automatically select the desired closed captions.

All the proposed descriptors are verified by extensive test dataset of various characteristics. In the experiments comprising comprehensive comparisons, the proposed descriptors outperform several related motion activity descriptors.

7.2 Future Work

In the future, in order to allow users to browse and to search a video sequence in a short time, a more compact semantic form representing video content is indispensable. Therefore, with the video characterization capabilities of the proposed text and motion

activity features, there are several interesting extensions on extractions of high-level semantic features, as listed below.

- Design of New Visual Features

Although several visual descriptors are proposed in MPEG-7, such as color, texture and motion, MPEG-7 standardizes only a number but not nearly all useful features. It is necessary to design and implement additional descriptors for symmetry detection of objects (e.g., face detection), object-based description in video streams (e.g., structure recognition from motion), and semantic high-level video event analysis from uncompressed as well as compressed video streams. Additionally, we plan to describe object-based 3D features. For example, in compressed videos we can generate 3D objects in two steps (1) detect moving blobs in B- or P-frames (2) locate the corresponding blobs in I-frames. Subsequently, new features in the 3D blobs should be investigated such as the feature point and shape of the 3D volume.

- Employment of Video Context Information

In an article, we can often tell the meaning of a word from its context. Similarly, in the video content, human can realize what the shot means from its neighboring shots – the video context. In the research issue of video context, neighboring shots related with the target should be identified. However, how to identify related neighboring shots with least time constraint is the first critical problem. Once related shots are identified, how to compute the similarity between the query and target shot sets is another challenge.

- Similarity Measurement

The goal is the design of methods for query definition that are flexible enough to

satisfy the different ways how humans can perceive and judge similarity and are applicable in different querying environment. In this thesis, we have proposed motion-based weighted distance metric, which can be used to effectively distinguish between video clips that are of dominant motion either in horizontal or vertical directions. In the future, camera motion can be estimated and be considered in the similarity measurement because camera motion is not only the content of human perception but it is an important cue for classifying general videos.



References

- [1] ISO/IEC JTC1/SC29/WG11/N3913, "Study of CD 15938-3 MPEG-7 Multimedia Content Description Interface – Part 3 Visual," Pisa, January 2001.
- [2] N. Babaguchi, Y. Kawai and T. Kitahashi, "Event Based Indexing of Broadcasted Sports Video by Intermodal Collaboration," IEEE Transactions on Multimedia, Vol. 4, No. 1, March 2002.
- [3] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang and M. Sakauchi, "Automatic Parsing of TV Soccer Programs," Proc. International Conference on Multimedia Computing and Systems," pp. 167-174, May 1995.
- [4] G. Sudhir, John C. M. Lee and Anil K. Jain, "Automatic Classification of Tennis Video for High-Level Content-based Retrieval," Proc. IEEE International Workshop Content-Based Access of Image and Video Database, 1998, pp. 81-90.
- [5] G. S. Pingali, Y. Jean and I. Carlbom, "Real Time Tracking for Enhanced Tennis Broadcasts," Proc. IEEE Computer Society Conference Computer Vision and Pattern Recognition, 1998, pp. 260-265.
- [6] H. Miyamori and S. I. Iisaku, "Video Annotation for Content-based Retrieval using Human Behavior Analysis and Domain Knowledge," Proc. Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000, pp. 320–325.
- [7] N. Haering, R. J. Qian, and M. I. Sezan, "A Semantic Event-Detection Approach and Its Application to Detecting Hunts in Wildlife Video," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, No. 6, September 2000, pp. 857-868.
- [8] H. L. Eng, and K. K. Ma, "Bidirectional Motion Tracking for Video Indexing," Proc. 3rd IEEE Workshop on Multimedia Signal Processing, 1999, pp. 153-158.

- [9] L. Favalli, A. Mecocci, and F. Moschetti, "Object Tracking for Retrieval Applications in MPEG-2," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 3, April 2000, pp. 427-432.
- [10] D. Y. Chen and Suh-Yin Lee, "Motion-Based Semantic Event Detection for Video Content Descriptions in MPEG-7," *Proc. 2nd IEEE Pacific Rim Conference on Multimedia*, Beijing, China, Oct. 2001, pp. 110-117.
- [11] M. Lee, and B. Ahn, "Robust Algorithm for Scene Analysis on Compressed Video," *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1999, pp. 103-106.
- [12] J. Nang, S. Hong, and Y. Ihm, "An Efficient Video Segmentation Scheme for MPEG Video Stream using Macroblock Information," *Proc. ACM Multimedia Orlando, USA*, 1999, pp. 23-26.
- [13] S. C. Pei, and Y. Z. Chou, "Efficient MPEG Compressed Video Analysis Using Macroblock Type Information," *IEEE Transactions on Multimedia*, Vol. 1, No. 4, December 1999, pp. 321-333.
- [14] R. Wang, and T. Huang, "Fast Camera Motion Analysis in MPEG domain," *Proc. IEEE International Conference on Image Processing*, 1999, Vol. 3, pp. 691-694.
- [15] Y. P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid Estimation of Camera Motion from Compressed Video with Application to Video Annotation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 1, February 2000, pp. 133-146.
- [16] S. Y. Lee, J. L. Lian and D. Y. Chen, "Video Summary and Browsing Based on Story-Unit for Video-on-Demand Service," *Proc. 3rd International Conference on Information, Communications, and Signal Processing*, Singapore, Oct. 2001.
- [17] ISO/IEC JTC1/SC29/WG11/N3964, "MPEG-7 Multimedia Description Schemes XM (v7.0)," Singapore, March 2001.

- [18] <http://www.itftennis.com/html/rule/>
- [19] Coding of Moving Pictures and Associated Audio-for Digital Storage Media at up to about 1.5Mbit/s, Committee Draft of Standard ISO11172: ISO/MPEG 90/176, November 1991.
- [20] J. L. Mitchell, W. B. Pennebaker, Chad E.Fogg, and Didier J. LeGall, "MPEG VIDEO COMPRESSION STANDARD," Chapman&Hall, NY, USA, 1997.
- [21] J. Meng, Y. Juan, S.F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence," Proc. IS&T/SPIE, Vol. 2419, pp.14-25, 1995.
- [22] D. Y. Chen, S. J. Lin and Suh-Yin Lee, "Motion Activity Based Shot Identification and Closed Caption Detection for Video Structuring," Proc. 5th International Conference on Visual Information System, pp. 288-301, March, 2002.
- [23] D. Y. Chen, M. H. Hsiao and Suh-Yin Lee, "Automatic Closed Caption Detection and Font Size Differentiation in MPEG Videos," Proc. 5th International Conference on Visual Information System, pp. 276-287, March 2002.
- [24] S. W. Lee, Y. M. Kim and S. W. Choi, "Fast Scene Change Detection using Direct Feature Extraction from MPEG Compressed Videos," IEEE Transactions on Multimedia, Vol. 2, No. 4, pp. 240-254, Dec. 2000.
- [25] J. Nang, O. Kwon and S. Hong, "Caption Processing for MPEG Video in MC-DCT Compressed Domain," Proc of ACM Multimedia Workshop, pp. 211-214, 2000.
- [26] H. Wang and S. F. Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 4, pp. 615-628, August 1997.
- [27] H. Luo and A. Eleftheriadis, "On Face Detection in the Compressed Domain," Proc. of ACM Multimedia, pp. 285-294, 2000.

- [28] H. J. Zhang, C. Y. Low, S. W. Smoliar and J. H. Wu, "Video Parsing and Browsing Using Compressed Data," *Multimedia Tools and Applications*, pp. 89-111, 1995.
- [29] Y. Zhong, H. Zhang and A. K. Jain, "Automatic Caption Localization in Compressed Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 4, pp. 385-392, April 2000.
- [30] Y. Zhang and T. S. Chua, "Detection of Text Captions in Compressed Domain Video," *Proc. of ACM Multimedia Workshop*, pp. 201-204, 2000.
- [31] X. Chen and H. Zhang, "Text Area Detection from Video Frames," *Proc. of 2nd IEEE Pacific Rim Conference on Multimedia*, pp. 222-228, Oct. 2001.
- [32] H. Li, D. Doermann and O. Kia, "Automatic Text Detection and Tracking in Digital Video," *IEEE Transactions on Image Processing*, Vol. 9, No. 1, pp. 147-156, Jan. 2000.
- [33] J. C. Shim, C. Dorai and R. Bollee, "Automatic Text Extraction from Video for Content-Based Annotation and Retrieval," *Proc. 14th International Conference on Pattern Recognition*, pp. 618-620, 1998.
- [34] J. Ohya, A. Shio and S. Akamatsu, "Recognizing Characters in Scene Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 2, pp. 214-220, February 1994.
- [35] U. Gargi, S. Antani and R. Kasturi, "Indexing Text Events in Digital Video Databases," *Proc. 14th International Conference on Pattern Recognition*, pp. 916-918, 1998.
- [36] S. Kannangara, E. Asbun, R. X. Browning and E. J. Delp, "The Use of Nonlinear Filtering in Automatic Video Title Capture," *Proc. IEEE/EURASIP Workshop on Nonlinear Signal and Image Processing*, 1997.
- [37] V. Wu, R. Manmatha and E. M. Riseman, "TextFinder: An Automatic System to

- Detect and Recognize Text in Images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 11, pp. 1224-1229, November 1999.
- [38] A. K. Jain and B. Yu, “Automatic Text Location in Images and Video Frames,” Patter Recognition, Vol. 31, No. 12, pp. 2055-2076, 1998.
- [39] S. W. Lee, D. J. Lee and H. S. Park, “A New Methodology for Grayscale Character Segmentation and Recognition,” IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 18, No. 10, pp. 1045-1050, Oct. 1996.
- [40] W. Qi, L. Gu, H. Jiang, X. R. Chen and H. J. Zhang, “Integrating Visual, Audio and Text Analysis for News Video,” Proc. International Conference on Image Processing, Vol. 3, pp. 520-523, 2000.
- [41] D. Chen, K. Shearer and H. Boulard, “Text Enhancement with Asymmetric Filter for Video OCR,” Proc. 11th International Conference on Image Analysis and Processing, pp. 192-197, Sep. 2001.
- [42] T. Sato, T. Kanade, E. K. Hughes and M. A. Smith, “Video OCR for Digital News Archive,” Proc. IEEE International Workshop on Content-Based Access of Image and Video Database, pp. 52-60, Jan. 1998.
- [43] Y. Ariki and K. Matsuura, “Automatic Classification of TV News Articles Based on Telop Character Recognition,” Proc. IEEE International Conference on , pp. 148-152, 1999.
- [44] S. W. Lee and D. S. Ryu, “Parameter-Free Geometric Document Layout Analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 11, pp. 1240-1256, November 2001.
- [45] R. G. Casey and E. Lecolinet, “A Survey of Methods and Strategies in Character Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence,” Vol. 18, No. 7, pp. 690-706, July 1996.
- [46] W. Qi, L. Gu, H. Jiang, X. R. Chen and H. J. Zhang, “Integrating Visual, Audio

- and Text Analysis for News Video,” Proc. International Conference on Image Processing, Vol. 3, pp. 520-523, 2000.
- [47] H. Lu and Y. P. Tan, “Sports Video Analysis and Structuring,” Proc. IEEE 4th Workshop on Multimedia Signal Processing, pp.45-50, 2001.
- [48] Y. M. Kwon, C. J. Song and I. J. Kim, “A New Approach for High Level Video Structuring,” Proc. IEEE International Conference on Multimedia and Expo., Vol. 2, pp. 773-776, 2000.
- [49] A. Hanjalic and R. L. Lagendijk, “Automated High-Level Movie Segmentation for Advanced Video-Retrieval Systems,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 4, pp. 580-588, June 1999.
- [50] M. M. Yeung and B. L. Yeo, “Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 5, pp. 771-785, Oct. 1997.
- [51] D. Y. Chen, H. T. Chen and S. Y. Lee, “Motion Activity Based Semantic Video Similarity Retrieval,” Proc. IEEE 3rd Pacific Rim Conference on Multimedia, pp. 319-327, Hsinchu, Taiwan, Dec 2002.
- [52] T. Kohonen, “The Self-Organizing Map,” Proceedings of IEEE, 78: 1464-1480, 1990.
- [53] T. Sikora, “The MPEG-7 Visual Standard for Content Description – An Overview,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, pp. 696 –702, June 2001.
- [54] A. Divakaran, K. Peker and H. Sun, “A Region Based Descriptor for Spatial Distribution of Motion Activity for Compressed Video,” Proc. International Conference on Image Processing, Vol. 2, pp. 287-290, Sep. 2000.
- [55] S. Jeannin and A. Divakaran, “MPEG-7 Visual Motion Descriptors,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, pp.

720-724, June 2001.

- [56] Z. Aghbari, K. Kaneko and A. Makinouchi, "A Motion-Location Based Indexing Method for Retrieving MPEG Videos," Proc. 9th International Workshop on Database and Expert Systems Applications, pp. 102-107, Aug. 1998.
- [57] K. A. Peker, A. A. Alatan and A. N. Akansu, "Low-Level Motion Activity Features for Semantic Characterization of Video," Proc. IEEE International Conference on Image Processing, Vol. 2, pp 801-804, Sep. 2000.
- [58] R. Wang, H. J. Zhang and Y. Q. Zhang, "A Confidence Measure Based Moving Object Extraction System Built for Compressed Domain," Proc. IEEE International Symposium on Circuits and Systems, Vol. 5, pp. 21-24, May 2000.
- [59] R. Wang, M. R. Naphade, and T. S. Huang, "Video Retrieval and Relevance Feedback in The Context of A Post-Integration Model," Proc. IEEE 4th Workshop on Multimedia Signal Processing, pp. 33-38, Oct. 2001.
- [60] T. Lin, C. W. Ngo, H. J. Zhang and Q. Y. Shi, "Integrating Color and Spatial Features for Content-Based Video Retrieval," Proc. IEEE International Conference on Image Processing, Vol. 2, pp. 592-595, Oct. 2001.
- [61] S. S. Cheung and A. Zakhor, "Video Similarity Detection with Video Signature Clustering," Proc. IEEE International Conference on Image Processing, Vol. 2, pp. 649-652, Sep. 2001.
- [62] L. Agnihotri and N. Dimitrova, "Video Clustering Using SuperHistograms in Large Archives," Proc. 4th International Conference on Visual Information Systems, pp. 62-73, Lyon, France, November 2000.
- [63] M. Roach, J. S. Mason and M. Pawlewski, "Motion-Based Classification of Cartoons," Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 146-149, Hong Kong, May 2001.
- [64] L. Zhao, W. Qi, S. Z. Li, S. Q. Yang and H. J. Zhang, "Content-based Retrieval

- of Video Shot Using the Improved Nearest Feature Line Method,” Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 3, pp. 1625-1628, 2001.
- [65] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan and A. Yamada, “Color and Texture Descriptors,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, pp. 703-715, June 2001.
- [66] R. Mohan, “Video Sequence Matching,” IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 6, pp. 3697-3700, May 1998.
- [67] M. M. Yeung and B. Liu, “Efficient Matching and Clustering of Video Shots,” Proc. IEEE International Conference on Image Processing, Vol. 1, pp. 338-341, Oct. 1995.
- [68] Y. Q. Shi and H. Sun, Image and Video Compression for Multimedia Engineering. CRC Press, New York, 2000.
- [69] ISO/IEC JTC1/SC29/WG11/N2466, “Licensing Agreement for the MPEG-7 Content Set,” Atlantic City, USA, October 1998.
- [70] ISO/IEC JTC1/SC29/WG11/N4547, “Extraction and Use of MPEG-7 Descriptions,” Pattaya, December 2001.
- [71] Ahmad, A.M.A., D. Y. Chen and Suh-Yin Lee, “Robust Object Detection Using Cascade Filter in MPEG Videos,” Proc. IEEE 5th International Symposium on Multimedia Software Engineering, pp. 196-203, Taichung, Taiwan, Dec 2003.
- [72] M. Hu, “Visual Pattern Recognition by Moment Invariants,” IRE Transactions on Information Theory, Vol. IT-8, pp. 179-187, Feb. 1962.