



THE HONG KONG
POLYTECHNIC UNIVERSITY

香港理工大學

Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

The Hong Kong Polytechnic University

Department of
Electronic and Information Engineering

Efficient Techniques for Video Retrieval

Sze Kin-Wai

A thesis submitted in partial fulfillment of the requirements for
the Degree of Master of Philosophy

December 2003



Pao Yue-kong Library
PolyU • Hong Kong

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best to my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

SZE KIN WAI (Name of student)

Abstract

Content-Based Video Retrieval (CBVR) is one of the major applications of multimedia signal analysis. Although research on this topic has been conducted for more than twenty years, many problems still remain, and better techniques for CBVR are needed. Therefore, the objectives of this thesis are to devise and develop efficient methods for video parsing and video content representation used in CBVR. In this thesis, different approaches for shot boundary detection and video content representation are reviewed. Shot boundary detection is the first step in analyzing and understanding the structure of a video for CBVR. Their accuracy will directly affect the performance of the retrieval system. However, since there are various types of transitions in a video, and the video may consist of strong motion, sudden change caused by lighting conditions, etc., the detection procedure is difficult. Moreover, video content representation plays an important role in the retrieval process because it affects the retrieval performance. Thus, efficient algorithms for CBVR remain a challenging research topic.

In this research, we have proposed a robust and efficient approach based on the Colored Pattern Appearance Model (CPAM) to represent a frame for shot boundary detection. Instead of using color histogram, CPAM represents a

frame by means of global statistics concerning the local visual appearance, and was originally motivated by studies in human color vision. Then, entropic thresholding is applied to determine the optimal threshold for shot boundary detection. After a video is temporally segmented into shots, a feature vector can be extracted from a shot for video retrieval based on its content. A new video content representation method has been proposed to represent a shot by considering the probability of occurrence of those pixels at the corresponding pixel position among the frames in a video shot. Experimental results show that our representation scheme outperforms the optimal key frame histogram and the alpha-trimmed average histograms. Finally, we have also developed a software library for video retrieval.

Author's Publications

The following technical papers have been published or accepted for publication based on the result generated from this work.

Journal paper:

1. Kin-Wai Sze, Kin-Man Lam and Guoping Qiu, "A New Key Frame Representation For Video Segment Retrieval", accepted to appear in IEEE Transactions on Circuits and Systems for Video Technology.

Conference papers:

2. Kin-Wai Sze and Kin-Man Lam, "An Evaluation Of Low-Level Visual Features for Automatic Video Segmentation", Proceedings, The 4th ACM Postgraduate Research Day, pp. 95 – 98, 2003.
3. Kin-Wai Sze, Kin-Man Lam and Guoping Qiu, "Scene Cut Detection Using The Colored Pattern Appearance Model", Proceedings, IEEE International Conference on Image Processing, Vol 2, pp.14 – 17, Sept. 2003, Barcelona, Spain.
4. Kin-Wai Sze, Kin-Man Lam and Guoping Qiu, "An Optimal Key Frame Representation for Video Shot Retrieval", accepted to appear in Proceedings, IEEE International Symposium on Intelligent Multimedia, Video & Speech Processing, 2004, Hong Kong.

Acknowledgements

I would like to express my sincere gratitude to my Chief Supervisor, Dr. K. M. Lam, as well as Dr. G. Qiu from School of Computer Science, The University of Nottingham, UK. Without their support, this research work would not have been completed. They also offered me many invaluable ideas and suggestions in writing my thesis.

I am also thankful to all the members of the DSP Research Laboratory, past and present. The countless discussions I had with them have been proved to be both fruitful and inspiring.

I would also like to take this opportunity to thank the Centre for Multimedia Signal Processing of the Department of Electronic and Information Engineering.

Finally, it is my pleasure to acknowledge and to thank the Research Office of The Hong Kong Polytechnic University for its generous support over the past two years.

Without the patience and forbearance of my family, the preparation of this research work would have been impossible. I appreciate their constant and continuous support and understanding.

Table of Contents

List of Figures and Tables	1
CHAPTER 1 Introduction	3
1.1 Motivation of Multimedia Signal Analysis	3
1.2 Introduction to Multimedia Signal Analysis	4
1.3 Background	5
1.4 Investigated Approaches	6
1.5 Organization of the Thesis	7
CHAPTER 2 Overview of Shot Boundary Detection and Video Content Feature Extraction	9
2.1 Introduction	9
2.2 Problems of Shot Boundary Detection	12
2.3 Shot Boundary Detection	14
2.3.1 Spatial Matching Approach	14
2.3.2 Twin-Comparison Approach	15
2.3.3 Statistical Approach	17
2.3.4 Histogram-based Approach	19
2.3.4.1 HSV Color Space	19
2.3.4.2 HMMD Color Space	20
2.3.5 Edge-based Approach	21
2.3.6 Compressed Domain Approach	22
2.3.7 Spatio-temporal Slice Approach	23
2.4 Problems of Video Content Feature Extraction	25
2.5 Video Content Feature Extraction	26
2.5.1 Key-frame-based Representations	26
2.5.1.1 Predetermined Temporal Location	26
2.5.1.2 Optimal Key Frame Selection	27
2.5.1.3 Dynamic Key Frame Selection	27
2.5.1.4 Compressed-Domain Approach	28
2.5.2 Shot-based Representations	28
2.5.2.1 Histogram-based Representation	29
2.5.2.2 Temporal Slice Based Representation	30
2.5.2.3 Object-based Representation	31
2.6 Summary of Review	33

CHAPTER 3	An Evaluation of Low-Level Visual Feature for Automatic Shot Boundary Detection	34
3.1	Introduction	34
3.2	Shot Boundary Detection Using Colored Pattern Appearance Model (CPAM)	37
3.2.1	Colored Pattern Appearance Model (CPAM)	37
3.2.2	Frequency Sensitive Competitive Learning (FSCL)	41
3.3	Low-Level Feature Video Content Representation	42
3.3.1	RGB Color Histogram	42
3.3.2	HSV Color Histogram	43
3.3.3	HMMD Color Histogram	43
3.3.4	Opponent Color Histogram	44
3.4	Entropic Thresholding for Shot Boundary Detection	45
3.5	Experimental Results	48
3.6	Conclusion	52
CHAPTER 4	A New Content Representation for Video Segment Retrieval	53
4.1	Introduction	53
4.2	Temporally Maximum Occurrence Frame (TMOF)	55
4.3	Computational Complexity Analysis	65
4.4	Experimental Results	67
4.3.1	Optimum alpha-trimmed average histogram and TMOF	70
4.3.2	Performances of k-TMOF and k-pTMOF	73
4.3.3	Histogram Representation of k-TMOF and k-pTMOF	75
4.5	Conclusion	77
CHAPTER 5	Conclusion and Future Works	78
5.1	Conclusion	78
5.2	Future works	81
	Appendix	82
	References	86

List of Figures and Tables

Figure 2.1 The overall structure of a CBVR system.

Figure 2.2 Examples of cut dissolve and wipe.

Figure 2.3 Block Diagram for Shot Boundary Detection

Figure 2.4 Operation of the Twin-Comparison Approach.

Figure 2.5 Detection of dissolve by searching parabolic curves for variance and approximating constant curve for mean derivative in a video sequence.

Figure 2.6 The space model of HSV color

Figure 2.7 The space model of HMMD color.

Figure 2.8 Spatio-temporal slice patterns generated by various types of camera breaks.

Figure 2.9 The alpha-trimmed average histograms of a video shot.

Figure 2.10 Motion patterns in horizontal and vertical spatio-temporal slices extracted from the center of an image volume.

Figure 3.1 The CPAM algorithm.

Figure 3.2 Block Diagram for Codebook Generation.

Figure 3.3 Block Diagram for Extracting P & C Histogram.

Figure 4.1 An ideal representative frame for a shot with six frames.

Figure 4.2 The construction of the TMOF for a video shot.

Figure 4.3 The video shot, and its TMOF/2-TMOF/2-pTMOF of a home video.

Figure 4.4 The video shot, and its TMOF/2-TMOF/2-pTMOF of a news program.

Figure 4.5 Selections of pixel values using 2-TMOF and 2-pTMOF.

Figure 4.6 The overall average normalized modified retrieval rank of TMOF and the optimum alpha-trimmed average histogram when the number of queries varied from 1 to 1281.

Figure 4.7 The overall average recall of TMOF and the optimum alpha-trimmed average histogram when the number of queries varied from 1 to 1281.

Figure 4.8 The performances of k -TMOF and k -pTMOF using the minimum distance measure with different values of k .

Table 3.1 Performances of different low-level representations for scene cut detection based on video sequence 1.

Table 3.2 Performances of different low-level representations for scene cut detection based on video sequence 2.

Table 4.1 The respective performances of the optimal key frame histogram, the alpha-trimmed average histograms with $\alpha = \{0, 0.10, 0.15, 0.20, 0.25, 0.5\}$, and the TMOF, the 3-TMOF, and the 3-pTMOF.

Table 4.2 The performances of the k -TMOF and k -pTMOF represented as 256-bin histograms, and that of the optimum alpha-trimmed average histogram.

CHAPTER 1

Introduction

The objective of this chapter is to introduce the general concepts of Multimedia Signal Analysis and its applications. The state-of-the art technology for multimedia signal analysis will be presented. An overview of the techniques for multimedia signal analysis, especially for the Content-Based Video Retrieval System (CBVRS) [1, 6, 7, 46] will be presented, and the shot boundary detection algorithm and the video content representation method for CBVRS proposed in this thesis will also be given.

1.1 Motivation of Multimedia Signal Analysis

Due to the rapid increase in the amount of multimedia documentation generated and the wide range of multimedia applications, efficient and effective management of the data is indispensable. Manual annotation of multimedia documentation is also an arduous task. Therefore, a number of initiatives have been undertaken whose aims to efficiently store, access, digest, and retrieve multimedia information from the past two decades. However, the semantics of a multimedia document are embedded in multiple forms, such as audio, image and video, which increase its complexity. Hence,

multimedia signal analysis still is a challenging research topic.

1.2 Introduction to Multimedia Signal Analysis

Multimedia signal analysis [7], which refers to the computerized understanding of the semantic meanings of a multimedia document, appears to be a natural extension (or merging) of image signal analysis and audio signal analysis. However, there are a number of factors that are ignored when dealing with images which should be dealt with when using videos. These factors are primarily related to the temporal information available from a multimedia document. While these factors may complicate the analysis process, they may also help in characterizing information useful to the analysis. A video sequence can be viewed as a document organized by time, and can be parsed into logical units at different levels [1, 7, 39]. The basic unit is a frame. The concept of a video shot as a group of related consecutive frames in a video also provides a new idea for handling a video document. Then a story is composed of a set of scenes, and each scene contains a set of shots [58, 59]. A generic approach for managing video data is video parsing, which segments a video into shots by means of shot boundary detection. The temporal information also induces the concept of motion, which is an important attribute of video motion features such as moments of the motion field, motion histogram, or global motion parameters.

By the way, visual signals also have many features for their characterization [40],

which have been initialised in image signal analysis. Basically, features for image signals can be categorized into three groups: color [5, 29], texture [29, 38] and shape [24]. Color is an important attribute for image representations such as colour histogram. Texture is an important feature of a visible surface where repetition or quasi-repetition of a fundamental pattern occurs. Co-occurrence matrix representation and Tamura representation are popular forms of texture representation. Shape features can be represented using traditional shape analysis techniques such as moment invariants, Fourier descriptors, autoregressive models, and geometry attributes.

There are many features for the characterization of audio signal analysis [7], such as total spectrum power, subband powers, brightness, bandwidth, pitch frequency, mel-frequency cepstral coefficients, linear prediction coefficients, etc. These features are currently used for content-based audio classification and retrieval.

1.3 Background

Considering the vast amount of video data, the development of a means for the quick and relevant access of multimedia signals is critical. Video data should be structured and indexed. A video clip is a sequence of image frames, so indexing each of the frames as still images will introduce extremely high computation. Video is a structured medium in which the actions and events in a video program should be viewed as a document rather than a non-structured sequence of frames.

Content-based video retrieval system analyzes a video in terms of its content by means of three primary processes: video parsing, content analysis, and abstraction. Parsing is the process of extracting temporal structure of a video, which involves the detection of temporal boundaries and identification of story units. The content feature extraction process is the extraction of visual features that describe pattern, color object motions, events and actions in video sequences. Abstraction is a process which extracts or constructs a subset of video data from the original video, such as key-frames or key-sequences as entries for shots, scenes or stories. Based on the content features or meta-data obtained from these three processes, indexes for the video can be built through a clustering process, which classifies sequences or shots, into different visual categories or indexing structures. In this research, efficient algorithms contributing for video parsing, content analysis and abstraction will be investigated and developed.

1.4 Investigated Approaches

The objective of this research is to investigate and develop efficient algorithms of multimedia signal analysis for CBVRS. With multimedia signals, both audio and visual features represent the characteristics of the signal. By analysing the correlation between audio and visual features, algorithms for accurate and efficient multimedia signal accessing, digesting and retrieving can be devised. However, visual content

contributes more in multimedia signal analysis, while audio content is used to assist in the analysis process [7 - 9].

For video retrieval, it is almost impossible to use keywords to describe video sequences. The reasons are that this process requires tremendous manpower, and the keywords to be used are subjective. Therefore, a content-based retrieval technique is a solution. A content-based video retrieval system can provide the efficient indexing, retrieval and browsing of a video sequence.

A generic approach for managing video data is to analyze the temporal structure of a video [16, 28, 39, 49]. This involves video parsing, which partitions a video into story, scene or shot levels by identifying their boundary and analyzing their structures. The shot level is the bottom level, which can be obtained by shot boundary detection. Scene detection [30, 51] and story segmentation algorithms [7, 55] are based on analyzing the synthetic structure of a video. Basically, a shot is represented by a feature vector, which contains the most important visual content of the shot. The feature vector is then used to represent the video shots for analyzing the synthetic structure [57], indexing [55, 56] and retrieval.

1.5 Organization of the Thesis

The rest of this thesis will give an overview of existing techniques for multimedia signal analysis for content-based video retrieval system, of our proposed algorithms

for automatic scene break detection, and of video shot representation for video indexing.

Chapter 2 will present the state-of-the-art technology for shot boundary detection and video content representation for content-based video retrieval. In Chapter 3, we propose a robust approach for shot boundary detection using Colored Pattern Appearance Model (CPAM) [60] as a content representation and an adaptive thresholding technique, namely entropic thresholding [11], which determines the optimal threshold values for locating scene breaks in a video. This approach provides more reliable results for automatic scene break detection to segment a video into shot. In Chapter 4, we propose a new content representation of a video shot based on the concept of the probability of the occurrence of corresponding pixels from all frames in a video segment. Finally, a summary of the major developments and the conclusion of this research work are provided in Chapter 5.

Overview of Shot Boundary Detection and Video Content Feature Extraction

2.1 Introduction

Content-based Image Retrieval Systems (CBIRS) [1] have started flourishing on the Web. Their performances are continuously improving and their basic principles are very diverse. Content-based Video Retrieval Systems (CBVRS) are less common, and seem at a first glance to be a natural extension of CBIRS. However, a number of factors that are ignored when dealing with images should be dealt with for videos. These factors are primarily related to the temporal information available from a video document. While these factors may complicate the querying system, they may also help in characterizing useful information for querying video.

To analyze a video document, the primary process is to perform video parsing which is the process of extracting the temporal structure of a video. The basic unit of temporal structure of a video signal is a shot, which is defined as a sequence of frames that are continuously captured from the same camera. However, the definition of a shot change is difficult to make. Pronounced object or camera motions may change

the content of successive video frames drastically. Ideally, a shot can encompass pans, tilts, or zooms. Therefore, shot boundary detection is not only a challenging process, but also the first step in the content-based analysis of a video document. Content analysis [3] is a process that extracts content features from a shot, scenes or stories. Figure 2.1 illustrates the overall structure of a CBVR system. Feature vectors extracted from video shots are then used to represent the content information about the shots for clustering and indexing. Meta-data refers to the information of the semantic structure of the video documents and their content features for retrieval and browsing. The performance of CBVR is quite dependent on the accuracy of shot boundary detection and the representative power of the feature used to represent the video shots. In other words, a reliable method for detecting the shot boundaries and an efficient feature for representing the video shots are indispensable to such applications. In this chapter, various approaches for shot boundary detection and video content feature extraction are reviewed.

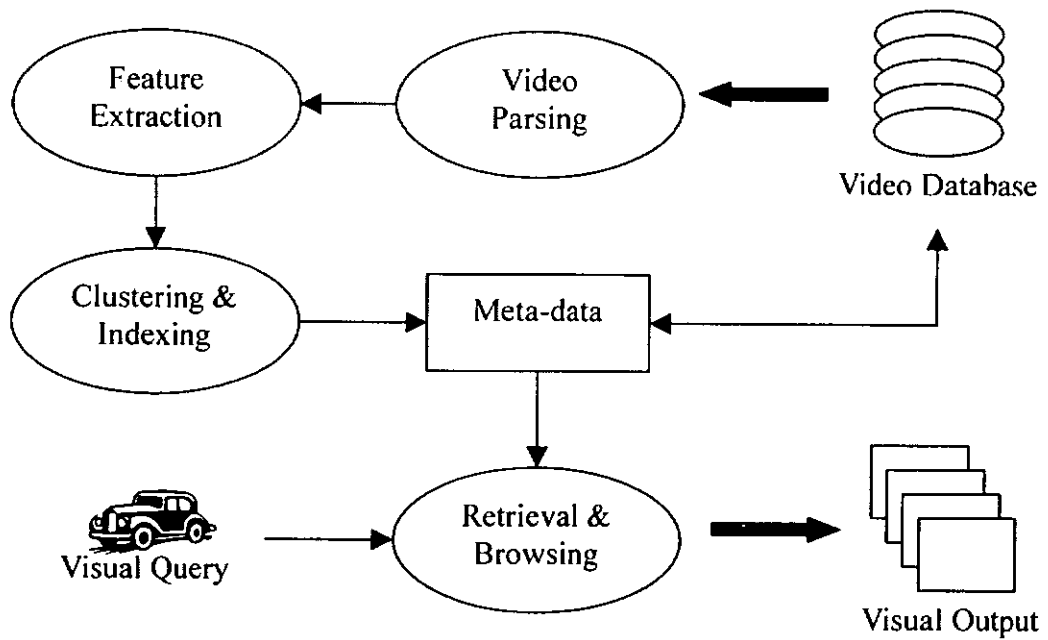


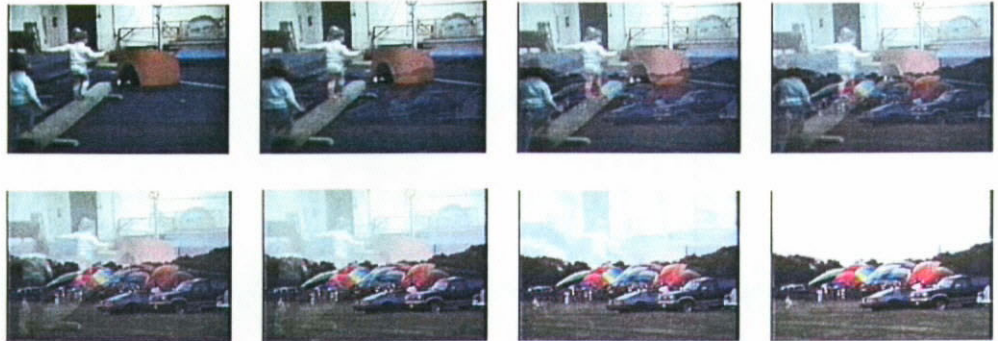
Figure 2.1 The overall structure of a CBVR system.

2.2 Problems of Shot Boundary Detection

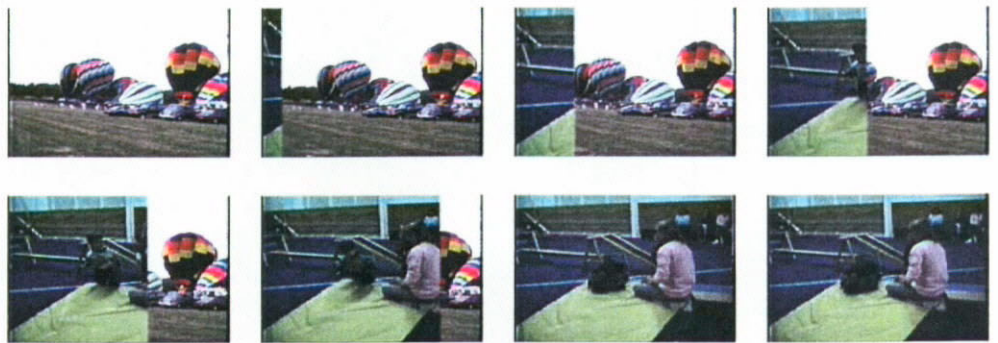
The task of identifying shot boundaries from a video sequence is a challenging process. Shot change detection refers to identify shot changes in a video sequence. It is not only a step to segment a video sequence into shots, but also to facilitate indexing the video sequence for fast browsing and retrieval of subsequences of interest to the users. The major difference between image signal and video signal is that a video signal consists of temporal information, which introduces the concept of object motion and includes camera motion. As a result, shot change detection may be corrupted by significant object and camera motion. Apart from these, shot change may occur in a variety of ways: abrupt transition, which refers to camera cut that is an instantaneous change from one shot to another; or gradual transitions such as cross-dissolve, fade-in, fade-out, and wipe, which are graphical editing effects used to accord varying semantic significance. A dissolve superimposes two shots where one shot gradually appears while the other fades out slowly. A wipe is a moving transition of a frame across the screen that enables one shot to gradually replace another. Figure 2.2 (a), (b) and (c) show examples of camera cut, dissolve and wipe, respectively.



(a) Camera Cut.



(b) Dissolve



(c) Wipe

Figure 2.2 Examples of cut dissolve and wipe.

2.3 Shot Boundary Detection

Shot Boundary Detection can be divided into two parts; feature extraction and shot change identification. Figure 2.3 shows the block diagram for video shot detection. Feature extraction refers to extracting useful data to represent the raw video data. The feature may be spatial information, histogram-based representation, etc., and can also be extracted from compressed domain directly. Shot change identification determines a shot change based on the extracted feature. Shot change can be occurred in different ways such as, abrupt change and gradual change. Several approaches have been developed to handle different kinds of gradual changes. These involve thresholding technique, statistical measure, graphical information, etc.

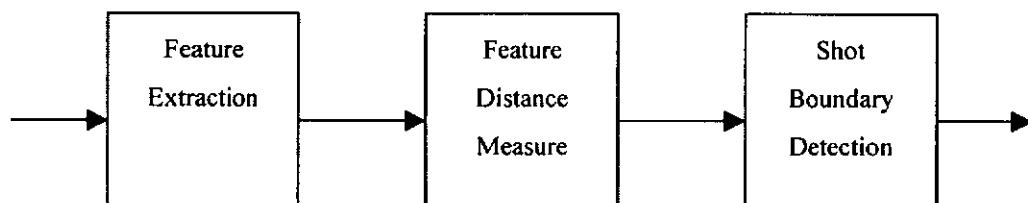


Figure 2.3 Block Diagram for Shot Boundary Detection

2.3.1 Spatial Matching Approach

Spatial matching approach [10] directly uses spatial information as feature to perform video shot detection. It temporally segments a video by identifying a large difference between two consecutive frames based on pixel-by-pixel difference or counting the number of different blocks [4, 43, 62]. The pixel-by-pixel difference and

the block-by-block difference measure the pixel difference and the total number of changed blocks, respectively. The total number of changed blocks is obtained by counting the number of corresponding blocks in two consecutive frames which have different. Then, abrupt change detection is performed by means of a threshold. If the number is larger than the predetermined threshold, these two frames will be considered being captured from different shots, and a shot change can be identified.

2.3.2 Twin-Comparison Approach

Due to the characteristic of gradual change, boundary detection with a large threshold cannot determine a gradual change. However, it is clear that certain changes are occurred during the gradual change. The twin-comparison approach [1, 4] was proposed to detect gradual change. This approach uses two thresholds to determine the beginning and the ending of a gradual change, as shown in Figure 2.4. This approach consists of two measures; the frame difference (T_g) and the accumulated difference (T_e). The frame difference is used to identify the starting frame of a gradual change and activate the accumulated difference measure where the frame difference is larger than a beginning threshold. The accumulated difference is used to identify the ending frame of a gradual change by an ending threshold. Therefore, a gradual change can be detected, as well as its beginning frame (F_b) and ending frame (F_e).

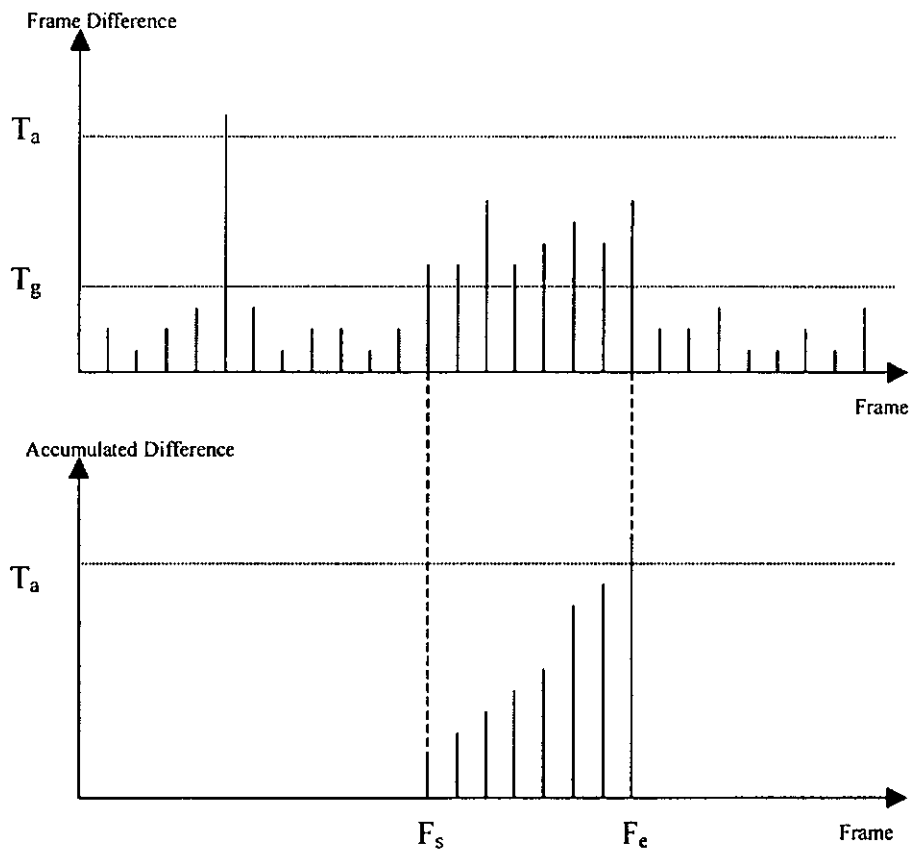


Figure 2.4 Operation of the Twin-Comparison Approach.

2.3.3 Statistical Approach

Statistical approach [4, 15, 44] is used to detect the dissolve special effect. A dissolve connects the boundaries of two shots smoothly. The connected shots share a blurred boundary region. A dissolve is denoted as the intensity function of a scan superimposed by two shots having intensity functions $S_1(x,y,t)$ with $t < t_2$ and $S_2(x,y,t)$ with $t > t_1$ respectively. It can be modeled as

$$Dissolve(x, y, t) = (1 - \alpha(t))S_1(x, y, t) + \alpha(t)S_2(x, y, t) \quad (2.1)$$

where $\alpha(t) = (t - t_1) / (t_2 - t_1)$ varies linearly with t in the range $[0, 1]$. Denote $\mu(t)$ as the mean intensity of a frame during the interval $t_1 < t < t_2$, then we have

$$\mu(t) = \mu^{S_1}(t) + (\mu^{S_2}(t) - \mu^{S_1}(t))\alpha(t) \quad (2.2)$$

where $\mu^{S_i}(t)$ is the mean intensity of a frame at time t that belongs to a shot. Taking the first derivative $\mu'(t) = (d\mu_i(t)/dt)$, we have

$$\mu'(t) = \frac{\mu^{S_2}(t) - \mu^{S_1}(t)}{t_2 - t_1}. \quad (2.3)$$

Assuming $\mu^{S_1}(t)$ and $\mu^{S_2}(t)$ remain unchanged during dissolve, $\mu'(t)$ is a constant value. Figure 2.5 illustrates the mean difference and variance of two dissolves in a video sequence.

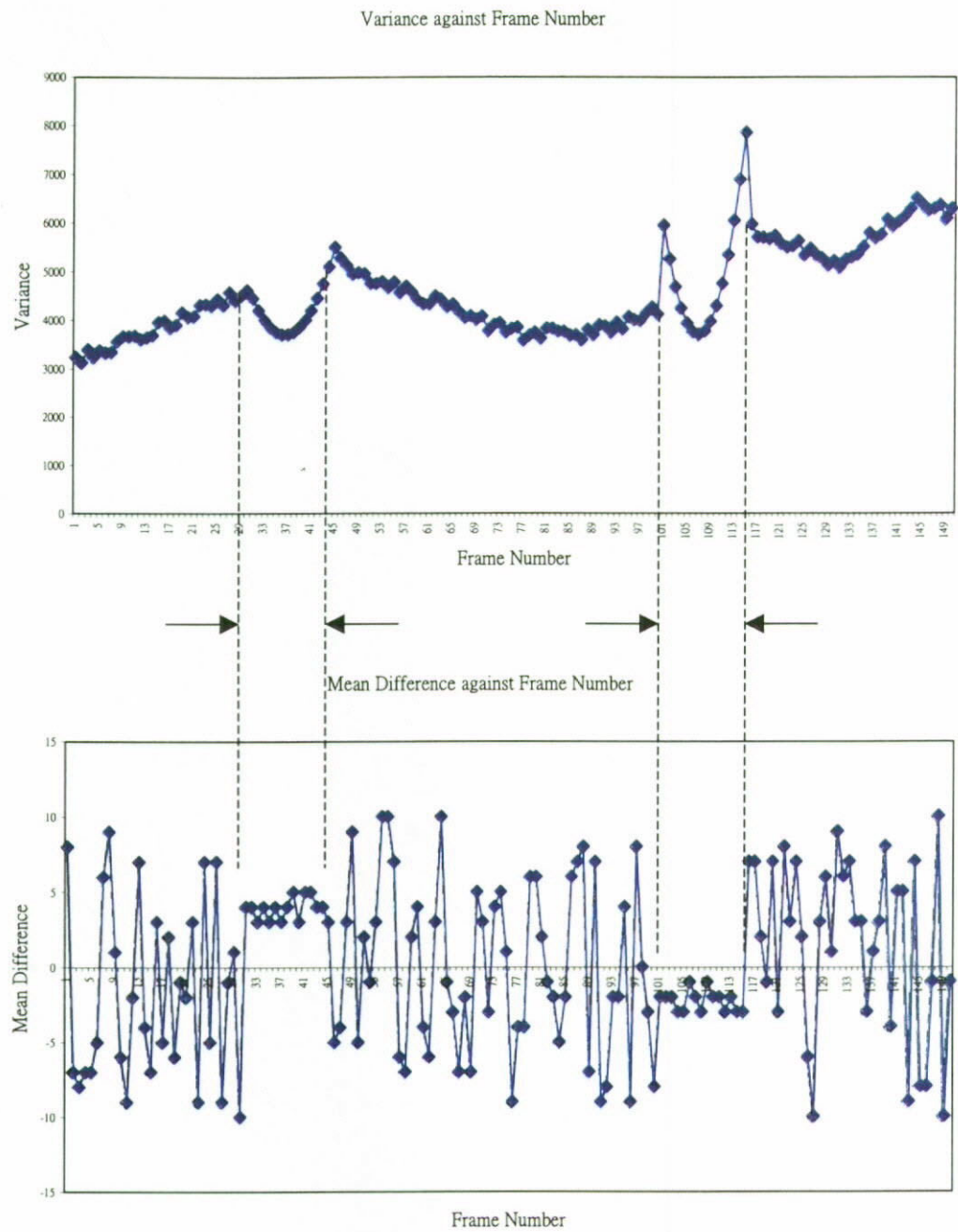


Figure 2.5 Detection of dissolve by searching parabolic curves for variance and approximating constant curve for mean derivative in a video sequence.

2.3.4 Histogram-based Approach

Histogram-based approach [10, 13] for temporal video segmentation represents each frame of a video by a color histogram. Similar to the spatial matching approach, both abrupt change and gradual change can be detected by the same measurement techniques. Color is the most expressive of all the visual features and has been extensively studied in the research of image retrieval during the last decade. Color histogram is commonly used to analyze or represent visual content. Change detection is then performed based on the extracted color histograms. However, many different color spaces [29] can be used, such as monochrome, YCrCb, HSV and HMMD. RGB is defined as reference chromaticity primaries in the following because it is available from the capture process.

2.3.4.1 HSV Color Space

HSV color space is a popular choice for manipulating color. The HSV color space is developed to provide an intuitive representation of color and to approximate the way in which humans perceive and manipulate color. RGB to HSV is a nonlinear, but reversible, transformation. Figure 2.6 shows the space model of HSV color. The hue (H) represents the dominant spectral component – color in its pure form, as in green, red, and yellow. Adding white to the pure color changes the color: the less the white,

the more saturated the color is. This corresponds to the saturation (S). The value (V) corresponds to the brightness of a color. The coordinate system is cylindrical, and is often represented by a subspace defined by a six-sided inverted pyramid. The top of the pyramid corresponds to $V = 1$, with the “white” at the center. The hue is measured by the angle around the vertical axis, with red corresponding to 0. The saturation S ranges from 0 at the center to 1 on the surface of the pyramid. An inverted cone is also used to denote the subspace instead of the pyramid.

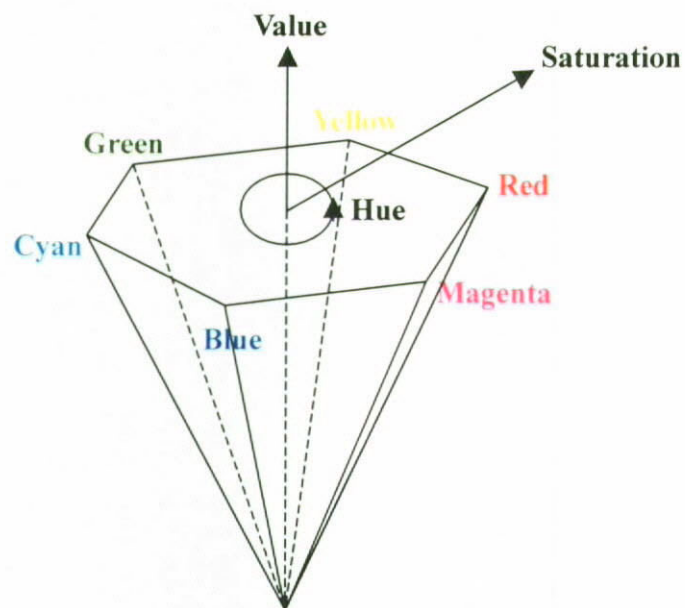


Figure 2.6 The space model of HSV color

2.3.4.2 HMMD Color Space

A new color space, the HMMD color space, is also supported in MPEG-7. This color space is formed by hue (H), saturation (S), and diff (D). The hue the saturation have the same meaning as in the HSV space, and max and min are the maximum and

minimum among the R, G and B values, respectively. The diff component is defined as the difference between max and min. Only three of the four components are sufficient to describe the HMMD space. This color space can be depicted using the double cone structure as shown in Figure 2.7.

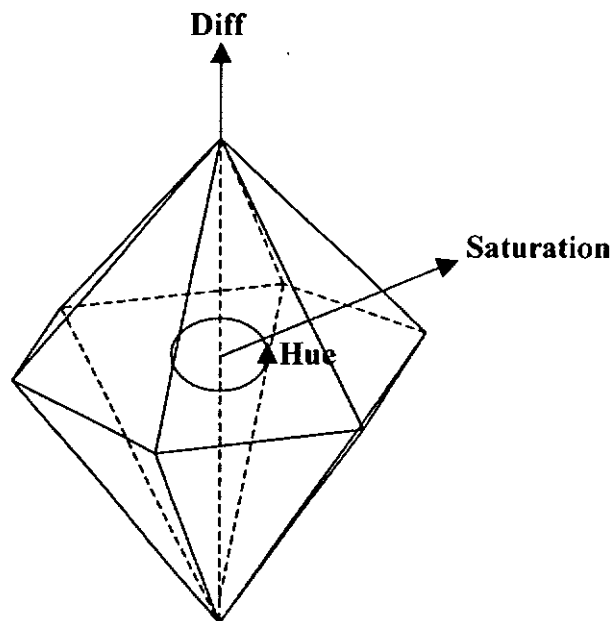


Figure 2.7 The space model of HMMD color.

2.3.5 Edge-based Approach

The idea of edge-based approach [30] is that new intensity edges appear far from the locations of old edges during a cut or dissolve. This approach detects the appearance of intensity edges that are distant from edges in the previous frame and appears to be more accurate at detecting and classifying scene change detection points that are difficult to detect with color histograms. However, a global motion computation is used to handle camera or object motion. This approach is

compute-intensive, and additional computation time is highly undesirable.

2.3.6 Compressed Domain Approach

Due to the large amount of data, video sequences are often compressed for efficient transmission or storage on-line. For efficient video storage and management, it is essential for us to have more intelligent video manipulating techniques. MPEG compressed streams are the most common approach for video data compressed and storage. However, shot change detection must be performed prior to all other processes. Most of the current shot change detection algorithms operate on uncompressed video sequences. For efficient shot change detection, it should be performed in the compressed domain. Three types of information can be directly achieved from a MPEG compressed stream: type of macro block [27], motion vectors of each macro block [23], and DCT coefficients [26, 36]. One possible approach is to directly extract visual feature from compressed domain [32-35, 41]. [25] proposed an algorithm which was made possible by a new mathematical formulation for deriving the edge information directly from the discrete cosine transform coefficients. The idea of this algorithm is to derive binary edge maps from the AC coefficients in blocks, which have been discrete-cosine transformed. Then, edge orientation, strength and offset using correlation between the AC coefficients in the derived binary edge maps can be measured by the new mathematical formulation. Finally, a scene change can be

detected by comparing the histograms of these features of two consecutive frames.

Another approach [31] determines shot boundaries using macroblock type information of MPEG compressed video bitstreams.

2.3.7 Spatio-temporal Slice Approach

Analysis of spatio-temporal slices for computational vision tasks has been proposed since 1985 [14]. A video can be arranged as a volume with image dimensions and temporal dimension. We can view the volume as formed by a set of 2D temporal slices each with dimension x and t or y and t . Each spatio-temporal slice is a collection of scans in the same selected position of every frame as a function of time. The slice is used to extract an indicator to capture the motion coherency of the video. Figure 2.8 shows three spatio-temporal video slices (horizontal slice, vertical slice and diagonal slice) taken from an image volume along the temporal dimension. Each slice contains several spatially uniform color-texture regions, and each region is considered having unique rhythm. Shot boundaries are located at places where the color and texture in a slice show dramatic changes. The shape and orientation of the dramatic changes are affected by the types of shot changes: a cut results in a vertical boundary line, a wipe results in a slanted boundary line, and a dissolve results in a slow transition which shows a burred boundary. Apart from these, camera motion and multiple motions can be determined by analyzing the texture of the slice [12].











Boundary Type	Horizontal Slice	Vertical Slice
Cut		
Wipe (Left To Right)		
Wipe (Barn Door)		
Wipe (Gradient)		
Dissolve		

Figure 2.8 Spatio-temporal slice patterns generated by various types of camera breaks.

2.4 Problems of Video Content Feature Extraction

Video content representation aims to effectively classify and index video shot for video browsing and retrieval. Retrieval and browsing require that the source material first be effectively indexed. A video document can be decomposed into three structural levels: stories, scenes and shots. Content-based indexing of video with visual features is still an open research problem. Visual features can be divided into two levels: low-level image features, and semantic features based on objects and events. The semantic level includes name, appearance, motion, and temporal variation of characteristics of constituent objects at different times and the contributions of all these attributes and relationships to the story being presented in a video sequence. Low-level feature indexing has been far more successful than that of semantic indexing in image database. However, the biggest problem with indexing video using the low-level image feature of every frame is its enormous volume, while uniform subsampling may reduce some data, but risky for losing important frames.

2.5 Video Content Feature Extraction

Key-frame representation for video indexing is a viable solution, but the problem becomes how to obtain the key-frames automatically from video sources. Some methods simply pick from every shot one or more frames in predetermined temporal locations [13], while other employ color and/or motion-based criteria for appropriate key frame selection [19-21]. In order to avoid the variations in the color description of a shot due to the inherent arbitrariness of key frame selection, a more favorable approach is to consider the color content of all the frames within a shot for color histogram computation.

2.5.1 Key-frame-based Representations

Key-frame-based approach use one or more key frames to represent a shot. The selected key frames are then used to represent the shot for video indexing, browsing and retrieval.

2.5.1.1 Predetermined Temporal Location

This approach simply selects the key frame(s) from a video shot based on predetermined temporal locations, such as the first frame, middle frame and/or last frame. This method is very simple and is suitable for static shot, but it may not select the most optimal frame for a shot that consists of strong temporal variations.

2.5.1.2 Optimal Key Frame Selection

Optimal key frame histogram [18] approach selects appropriate key frame based on color and/or motion-based criteria. Key frame is selected by optimizing the distortion function as shown below:

$$D(k_j) = \sum_{i=0}^{N-1} (f(i) - f(j))^2 \quad (2.4)$$

where $f(i)$ denotes the feature extracted from frame i .

This method considers the content of each shot to determine a suitable key frame for the shot. When a shot consists of strong motions, it can select a more suitable key frame for representing the shot. When the shot does not consist of strong motion, all frames from the shot are very similar. This method will therefore require more computation than the predetermined key frame selection.

2.5.1.3 Dynamic Key Frame Selection

Dynamic key frame selection aims to represent a video shots with different lengths and activities by using different number of key frames and the positions of key frames that could reflect the dynamics of a video shot. Similar to the optimal key frame selection, a set of appropriate key frame is selected based on different criteria such as color, motion, etc.

2.5.1.4 Compressed-Domain Approach

As mentioned Section 2.3.6, compressed-domain approach provides a more efficient way for shot boundary detection. Key frame extraction can also be performed based on the feature extracted from the compressed domain [42]. [22] measures visual content complexity of a shot by motion patterns which are extracted from the information about motion vector from a MPEG video stream. A motion pattern of a shot is usually composed of a motion acceleration process, followed by deceleration process. Such a motion pattern usually reflects an action in events. To extract key frames based on motion patterns, a motion model is built to reflect the motion activities in video shots for guiding the selection of key frames.

2.5.2 Shot-based Representations

Features extracted from key-frame-based representation cannot provide sufficient event-based classification and retrieval because the features themselves do not capture motion and temporal information in a shot. A more favorable approach is to use shot-based representation for classification and retrieval. This approach does not only provide a representation, which consists of temporal information of the shot, but also avoid the variations in the feature of a shot due to the inherent arbitrariness of key frame selection.

2.5.2.1 Histogram-based Representation

Histogram-based representation can be an extension of the key-frame-based representations. After a set of appropriate key frames is located, the video shot is then represented by histograms extracted from the key frames. However, in order to avoid the variations in the color description of a shot due to the inherent arbitrariness of key frame selection, a more favorable approach is to consider the color content of all the frames within a shot for color histogram computation. Average color histogram [18] may be considered as an appropriate choice. However, the average color histogram becomes vulnerable to outlier frames within a shot. Therefore, the alpha trimmed average histogram [18] was proposed to overcome the presence of outlier frames, which may skew the color representation unfavorably. The alpha trimmed average histogram considers all the corresponding histogram bin values from all the frames within the shot, and is generated using the trimmed mean operator. To obtain an alpha-trimmed average histogram, the corresponding bin values are sorted in either ascending or descending order, and then the average value for each bin is computed from the central members of the ordered array. Therefore, the value of bin j in the alpha-trimmed average histogram is computed as follows:

$$TrimHist_x(j, \alpha) = \frac{1}{M - 2 \cdot \lfloor \alpha M \rfloor} \sum_{m=\lfloor \alpha M \rfloor}^{M - \lfloor \alpha M \rfloor} h_j(m) \quad (2.5)$$

where, $\hat{h}_j(m)$ represents the ordered array of bin values, and the trimming parameter α , where $0 \leq \alpha \leq 0.5$, controls the number of bin values excluded from the average computation. When α is equal to zero, the resulting histogram is equivalent to the average histogram. On the contrary, when α is equal to 0.5, the resulting histogram is equivalent to the median histogram, as shown in Figure 2.9.

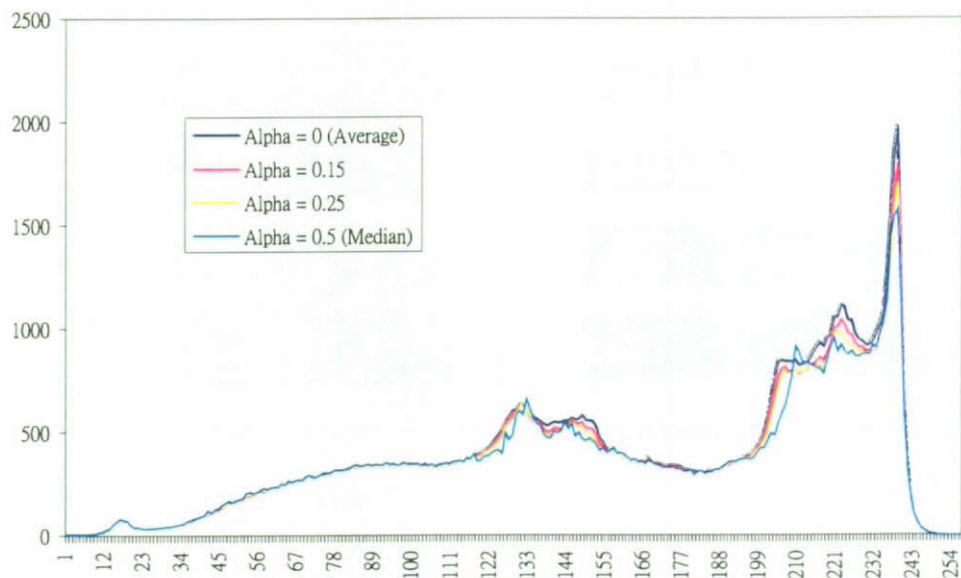


Figure 2.9 The alpha-trimmed average histograms of a video shot.

2.5.2.2 Temporal Slice Based Representation

Based on the analysis of spatio-temporal image volumes, an effective motion-based content representation was proposed in [17]. In the spatio-temporal slices of image volumes, motion is depicted as oriented patterns as shown in Figure

2.10. Using a tensor histogram computation algorithm, motion can be characterized efficiently. Not only the camera motion in a video can be annotated as static, pan, tilt, zoom, etc., but also the moving object can be segmented and tracked efficiently in the spatio-temporal images of video shots.











Boundary Type	Horizontal Slice	Vertical Slice
Static		
Strong Motion		
Tilt		
Zoom		
Tracking		

Figure 2.10 Motion patterns in horizontal and vertical spatio-temporal slices extracted from the center of an image volume.

2.5.2.3 Object-based Representation

As a Content-based query may involve an object in a video clip, the video frame-based representation may not provide sufficient resolution to support such query. Therefore, object-based representation can facilitate video content representation when dealing with queries about objects and their motion. Objects in clips can be represented by two types of information: `Descriptive_data` and `Motion_data`. `Descriptive_data` refers to object features like the identity of an object,

its color, shape, types, etc. Motion_data contains the center point locations of minimum bounding rectangles for the frames, and the widths and the heights of the object. Many works have been done in object-based representation schemes [23]. However, the problem of this approach is how to group a set of candidate regions to form an object automatically. This is still an area of ongoing research and is a challenging work. The motion-based approach uses the object motion information in order to characterize the events to allow subsequent retrieval. Based on the translation, spatial and temporal scale invariance properties, eight cases have been classified for Motion-based video indexing and retrieval.

2.6 Summary of Review

Shot boundary detection is an important step for video parsing, video indexing and video coding. However, the locations of shot boundaries in a video are very difficult to detect accurately. The performance to detect the boundary is affected by several variables, such as motions of objects, global motion (pan, zoom or tilt) and sophisticated transitions including dissolve, fade-in, fade-out and wipes, etc. In order to solve these problems, various approaches to shot boundary detection such as the spatial matching detection, twin-comparison approach, statistical approach, histogram-based approach, edge-based approach, and compressed domain approach have been introduced.

For video content feature extraction, one of the approaches is to extract features from the key frames that are selected from the shot based on several criteria. This method can facilitate shot representation for video indexing and retrieval, but it does not capture motion and temporal information in a shot for event-based classification and retrieval. Other approaches for video content feature extraction are shot-based representation, which extracts features by considering the temporal information of a shot. Both frame-based and object-based approaches to feature extraction have been presented.

An Evaluation of Low-Level Visual Feature for Automatic Shot Boundary Detection

3.1 Introduction

In Chapter 2, the problem of shot boundary detection and its applications have been addressed. Various approaches to shot boundary detection, such as spatial matching approaches, histogram-based approaches, edge-based approaches and compressed domain approaches, have been considered with a view to solve the problem. In this chapter, several histogram-based approaches will be evaluated, and a reliable and robust representation for shot boundary detection using the Colored Pattern Appearance Model (CPAM) will be introduced.

Content-based video indexing (CBVI) has been an extensively researched area in the computer vision community for the past decade, and many approaches have been developed and published. A general approach to CBVI is to temporally segment a video into shots based on the extracted low-level visual features, and to use the visual features for indexing and providing a high-level understanding of the video. Low-level representation of multimedia signals has been commonly used in segmentation,

indexing, retrieval, etc.

Temporal video segmentation plays a very important role in content-based video indexing. This process provides a fundamental understanding of indexing a video efficiently. Basically, it involves the detection of both abrupt and gradual transitions. In general, the detection of these kinds of transition involves two procedures: content representation and decision-making. In the content representation process, a video is represented by low-level features, such as DC image [37], edge image, monochrome histogram, color histogram in different color spaces, etc., which can be analyzed efficiently. In the decision-making process, thresholding technique is usually used to detect and identify the transitions. There are two ways to set the threshold: one is to pre-set it by experiments, the other is to set it adaptively.

In our approach, we use Colored Pattern Appearance Model (CPAM) as a frame representation for shot boundary detection, in which a scene is represented by means of global statistics of the local visual appearance, and was originally motivated by studies in human color vision. We will also evaluate the performances of different kinds of histogram-based low-level representation for automatic detection of abrupt transitions, so that the most effective and reliable one can be identified. There are many possible low-level visual features for representing video contents. With a particular representation, there may also be many ways to determine an abrupt

transition. The entropic thresholding technique is chosen in our analysis; it can provide an optimal and automatic solution in determining the threshold for detecting shot boundaries. The performance of our approach is then compared to several histogram-based approaches, such as RGB color histogram, monochrome color histogram, HSV color histogram, HMMD color histogram and opponent color histogram. In our experiments, the two video sequences in the MPEG-7 content set are used to evaluate the performances of the CPAM and the histogram-based methods. Experimental results show that our proposed model outperforms other histogram-based approaches in shot boundary detection.

3.2 Shot Boundary Detection Using Colored Pattern

Appearance Model (CPAM)

3.2.1 Colored Pattern Appearance Model (CPAM)

The main problem with shot boundary detection is the existence of strong noises and motion in the video data. Therefore, it is not easy to have a single representation that is efficient, reliable and robust for scene cut detection. The colored pattern appearance model (CPAM) which has two channels capturing the characteristics of the chromatic and achromatic spatial patterns of small image regions has been used to compile content descriptors for content-based still image retrieval [57]. The CPAM is shown in Figure 1.

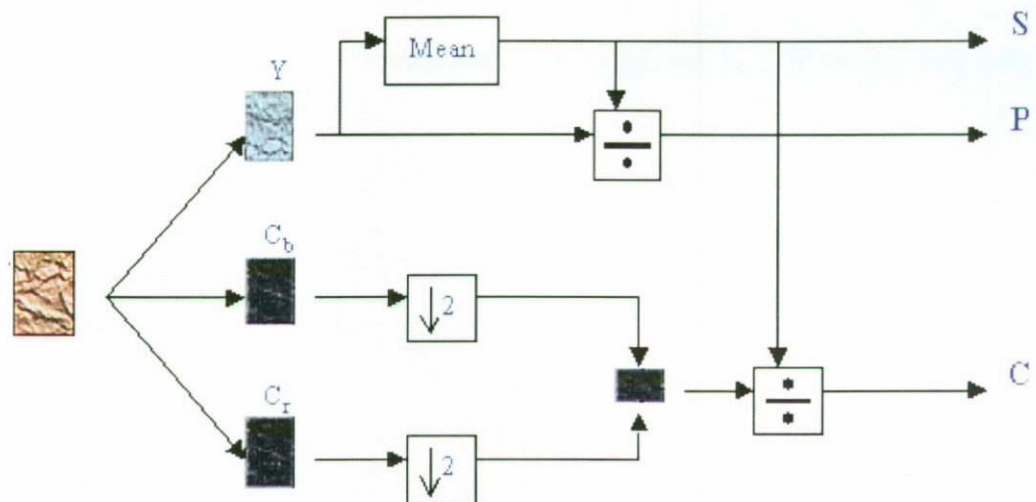


Figure 3.1 The CPAM algorithm.

In this model, the visual appearance of a small image block is modeled by three components: the stimulus strength, the spatial pattern and the color pattern. The

YCbCr space is used, and the stimulus strength S is approximated by the local mean of the Y component. The pixels in Y normalized by S form the achromatic spatial pattern (ASP) vector. Because C_b and C_r have lower bandwidth, they are sub-sampled. The sub-sampled pixels of C_b and C_r are normalized by S , and then concatenated together to form the chromatic spatial pattern (CSP) vector. There are two reasons to normalize the pattern and colour channels by the strength. First, from the coding point of view, removing the DC component makes the representation more efficient. Second, from the image indexing point of view, this can remove, to a certain extent, the effect of lighting conditions. This is because this process can make the visual appearance model somewhat “colour constant”, which can therefore improve the indexing and retrieval performance, especially in the case of retrieving similar surfaces under different light conditions. The formulations for extracting S , ASP vector and CSP vector are as follows:

Let $Y = \{y(i,j), i = 0, 1, 2, \dots, m, j = 0, 1, 2, \dots, n\}$ be the m by n Y image block, The stimulus strength of a block is calculated as

$$S = \frac{1}{m \times n} \sum_{i=0}^m \sum_{j=0}^n y(i, j) \quad (3.1)$$

Then the ASP pattern vector, $ASP = \{asp(i,j), i = 0, 1, 2, \dots, m, j = 0, 1, 2, \dots, n\}$ of the block, is formed as

$$asp(i, j) = \frac{y(i, j)}{S} \quad (3.2)$$

Then the CSP pattern vector, $CSP = \{csp(k), i = 0, 1, 2, \dots, 2M\}$ is formed by concatenating SC_b and SC_r .

The sub-sampled C_b signal, $SC_b = \{sc_b(k,l), k = 0, 1, 2, \dots, m/2, l = 0, 1, 2, \dots, n/2\}$

is obtained as

$$sc_b = \frac{1}{4S} \sum_{i=0}^{m/2} \sum_{j=0}^{n/2} c_b(2k+1, 2l+j) \quad (3.3)$$

Similarly, the sub-sampled C_r signal, $SC_r = \{sc_r(k,l), k = 0, 1, 2, \dots, m/2, l = 0, 1, 2, \dots, n/2\}$ is obtained as

$$sc_r = \frac{1}{4S} \sum_{i=0}^{m/2} \sum_{j=0}^{n/2} c_r(2k+1, 2l+j) \quad (3.4)$$

In order to use the representation scheme in content-based temporal video segmentation, vector quantization (VQ) is used to estimate statistically the most representative feature vectors in the feature space. A 256-codeword quantizer for the ASP vectors and a 256-codeword quantizer for the CSP vectors are generated by means of the frequency sensitive competitive learning (FSCL) algorithm [60]. Figure 3.2 shows how to generate the 256 codewords for the CSP vector and the 256 codewords for the ASP vector. Therefore, each frame can be represented by a 256-bin ASP histogram and/or a 256-bin CSP histogram. Figure 3.3 shows the block diagram of how to extract the ASP histogram and the CSP histogram of a frame. The training samples are based on the MIT Media Labs VisTex image database, which consists of images of different texture appearance. The codeword generated based on this

database can provide a general representation for different kinds of images/videos.

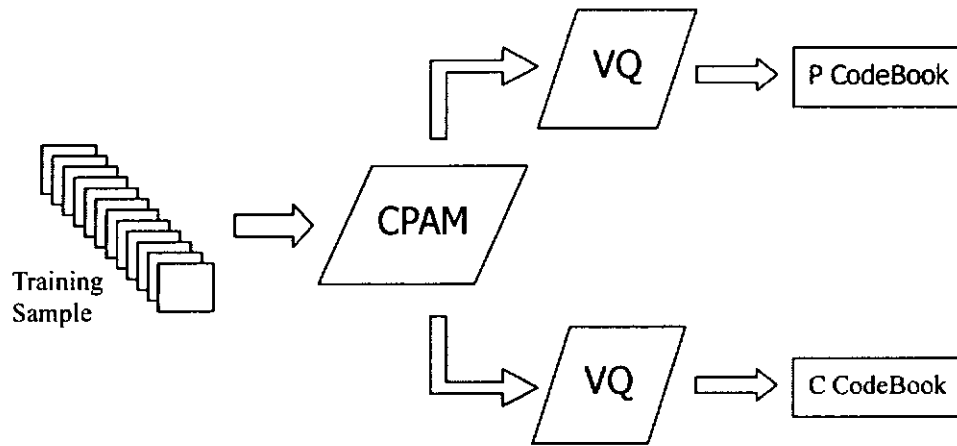


Figure 3.2 Block Diagram for Codebook Generation.

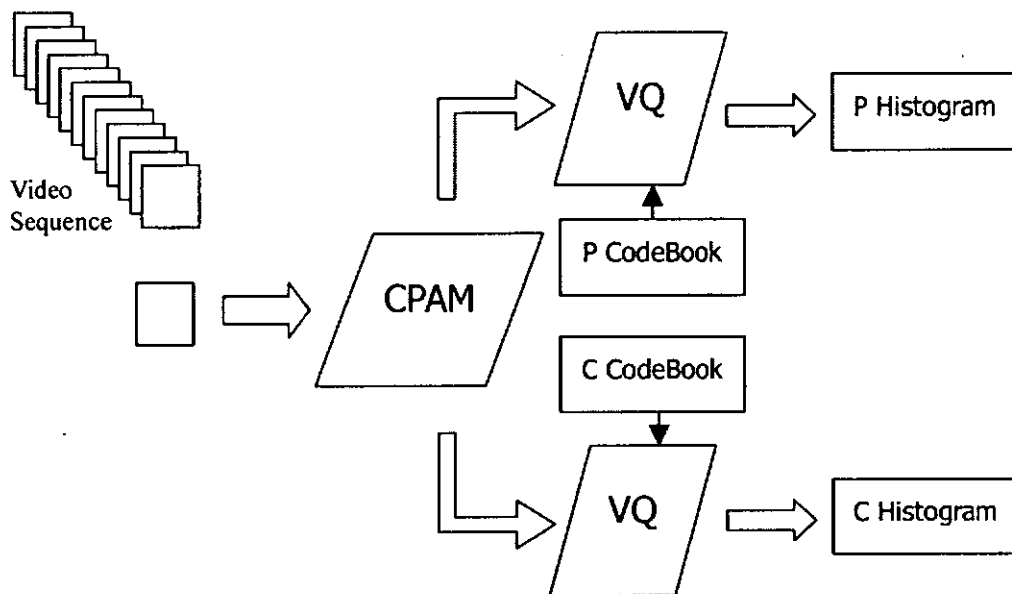


Figure 3.3 Block Diagram for Extracting P & C Histogram.

3.2.2 Frequency Sensitive Competitive Learning (FSCL)

Vector Quantization is a mature method of lossy signal compression/coding in which statistical techniques are used to optimize distortion/rate trade-offs. In this study, we used a specific neural network-training algorithm, the frequency sensitive competitive learning (FSCL) to design our codebook. According to [57], FSCL is insensitive to the initial choice of codeword, and the codeword designed by FSCL are more efficiently utilized than those designed by methods such as the LGB algorithm.

The FSCL method can be briefly described as follows:

1. Initialize the codeword, $C_i(0)$, $i = 1, 2, \dots, I$, to random numbers and set the counters associated with each codeword to 1, i.e., $n_i(0) = 1$.
2. Present the training sample, $X(t)$, where t is the sequence index, and calculate the distance between $X(t)$ and the codeword, $D_i(t) = D(X(t), C_i(t))$, and modify the distance according to $D'_i(t) = n_i(t)D_i(t)$.

3. Find j , such that $D'_j(t) \leq D'_i(t)$ for all i , update the codeword and counter

$$C_j(t+1) = C_j(t) + a[X(t) - C_j(t)] \quad (3.5)$$

$$n_j(t+1) = n_j + 1, \quad (3.6)$$

where $0 < a < 1$ is the training rate.

4. Repeat by going to 2.

3.3 Low-Level Feature Video Content Representation

An abrupt transition is usually induced by a camera break in a video. This change can be detected by computing the differences between the visual features of the consecutive frames. Many kinds of features for representing an image/frame have been proposed. Monochrome histograms and color histograms with different color spaces are the most commonly used methods for image representation. In MPEG-7, the color descriptors use different color spaces [29], such as monochrome, RGB, HSV, YCrCb, and HMMD. The opponent color space can also represent an image well for image indexing, and its transformation from the RGB color space is simple. In our evaluation, performances of these histogram-based representations will be compared with our approach and extraction procedures of these histogram-based representations are illustrated.

3.3.1 RGB Color Histogram

The RGB color space is the most common representation of color information. To generate a color histogram in RGB color space, the R, G and B components of each pixel in a frame are quantized into 256 color indices by vector quantization. This 256-bin color histogram $H_{RGB}(k)$ is then formed as follows:

$$H_{RGB}(k) = \sum_{i,j} \delta(Q_{R,G,B}(R_{i,j}, G_{i,j}, B_{i,j}) - k) \quad (3.7)$$

for $0 \leq k \leq 255$

$$\delta(i-j) = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

where i, j are the co-ordinates of a pixel, and $Q_{RGB}()$ represents the color quantization function which quantizes a color to one of the 256 color indices.

3.3.2 HSV Color Histogram

The HSV color space shown in Figure 2.4 can represent color information in the form most similar to human perception. RGB to HSV transformation is a nonlinear but reversible process. In order to quantize the HSV space into 256-bins, fixed quantization is used. The H, S and V values are coded into 4 bits, 2 bits and 2 bits, respectively. The 256-bin HSV color histogram $H_{H,S,V}(k)$ is then formed as follows:

$$H_{HSV}(k) = \sum_{i,j} \delta(L_{H,S,V}(H_{i,j}, S_{i,j}, V_{i,j}) - k), \quad (3.8)$$

for $0 \leq k \leq 255$

where $L_{H,S,V}()$ represents the color index of a pixel in the HSV color space.

3.3.3 HMMD Color Histogram

The HMMD color space shown in Figure 2.5 is a new color space supported in MPEG-7. Three components, Diff, Sum and Hue, are used to describe a color in the HMMD space. The Hue (H) has the same meaning as in the HSV space. The Diff (D) and Sum (S) components are defined as the difference between max and min and the

average of max and min, respectively, where max and min are the maximum and minimum among the R, G and B values. The H, D and S values are quantized into 4 bits, 2 bits and 2 bits, respectively, by a fixed quantization scheme according to MPEG-7 standard.

3.3.4 Opponent Color Histogram

The opponent color space [5] is a brightness-independent chromaticities space. This color space has the advantage of reducing the histogram dimensionality from 3-D to 2-D. The transformation of RGB to RgBy is simple. The opponent chromaticities are defined in terms of the r, g and b chromaticities:

$$(Rg, By) = \left(r - g, \frac{r + g}{2} - b \right) \quad (3.9)$$

where $r = \frac{R}{R + G + B}$

$$g = \frac{G}{R + G + B}$$

$$b = \frac{B}{R + G + B}$$

A 2-D color histogram is then formed with 32 bins per color axis.

3.4 Entropic Thresholding for Shot Boundary Detection

Thresholding technique is commonly used in segmentation and classification. With a selected threshold, an abrupt transition is declared if the histogram difference is larger than this threshold; otherwise, no abrupt transition occurs. The main problem here is how to determine the optimal threshold for different situations. Basically, there are two forms of the methods that can set the threshold. One is to preset the threshold by experimental results; the other is to set the threshold automatically based on the input data (video) itself. In shot boundary detection, it is difficult to pre-set a fixed threshold because different directors may have different styles and the videos may have different natures. Adaptive thresholding plays an important role in determination of the threshold under different situations. One of the optimal approaches is called entropic thresholding, which finds the optimal threshold by applying information theory. The entropic thresholding method has been extended to find the optimal threshold for spatial and temporal video segmentation. Two entropies are obtained from two separate probability distributions: one is for the in-class; the other is for the non in-class. The threshold used for segmentation is selected in such a way that the total entropy is maximized. In our experiments, entropic thresholding was used to determine the optimal thresholds for the different low-level video representations. After extracting each of the low-level representations, the differences between

successive frames of a video will be computed. The histograms f of successive frame

differences can be formed as follows:

$$dH_i = \sum_{j=0}^{G-1} |H_{i-1}(j) - H_i(j)| \quad \text{for } i = 1, 2, \dots, L-1 \quad (3.10)$$

$$W = \max_{i=1, 2, \dots, L-1} \{dH_i\} \quad (3.11)$$

$$f_k = \sum_{i=0}^{L-1} \delta(dH_i - k) \quad \text{for } 0 \leq k \leq W \quad (3.12)$$

where G is the total number of color levels in an image, L is the total number of frames in the video sequence, and dH represents the histogram difference between successive frames.

For shot boundary detection, the optimal threshold is calculated as follows:

$$P_{ns}(i) = \frac{f_i}{\sum_{k=0}^T f_k}, \quad 0 \leq i \leq T, \quad (3.13)$$

$$P_s(i) = \frac{f_i}{\sum_{k=T+1}^W f_k}, \quad T+1 \leq i \leq W. \quad (3.14)$$

$P_{ns}(i)$ and $P_s(i)$ represent the probability for the frames with the non-scene cut relationship with their successive frames and the probability for the frames with the scene cut relationship, respectively. The corresponding entropies for these two classes are:

$$E_{ns}(T) = -\sum_{i=0}^T P_{ns}(i) \log P_{ns}(i), \quad (3.15)$$

$$E_s(T) = -\sum_{i=T+1}^W P_s(i) \log P_s(i), \quad (3.16)$$

$E_{ns}(T)$ and $E_s(T)$ represent the entropies for these two classes regions separated by a

threshold T . The optimal threshold T_{opt} is chosen to satisfy the following criterion:

$$E(T_{opt}) = \max_{T=0,1,2,\dots,W} \{E_{ns}(T) + E_s(T)\} \quad (3.17)$$

3.5 Experimental Results

Many shot boundary detection schemes have been proposed and evaluated using different video sequences. This makes it difficult to compare the performances of the different detection schemes. In our experiments, the two home video sequences, namely Lgerca_1.mpg and Lgerca_2.mpg, from the MPEG-7 content set were used. These two video sequences have strong noises and motion. Each of the sequences consists of 42 scene cuts, which have been marked manually by the Requirements Group of the MPEG-7 standard committee.

The objective of our experiment is to seek the best representative low-level feature for automatic shot boundary detection. The histograms of monochrome, RGB, HSV, YCrCb, HMMD, RgBy, and CPAM were extracted from the videos, and their successive frame differences were then computed. For CPAM, the histograms based on the ASP, CSP, and joint ASP & CSP were considered. The entropic thresholding technique is then applied to each of the approaches such that their optimal thresholds for a video based on the frame difference values are selected. Recalls and Precisions of each of the representations were then measured. The formulations of the recall and precision are shown as follow:

$$\text{Recall} = \frac{CD}{CD + FP} \quad (3.18)$$

$$\text{Precision} = \frac{CD}{CD + FN} \quad (3.19)$$

where CD , FP and FN denote the correct detection, false positive and false negative, respectively. FP represents the number of false detection of scene break while FN represents the number of scene breaks missed in the detection.

The experimental results for the two video sequences are shown in Tables 3.1 and 3.2. Recall and Precision will be the basis of our analysis. The ranges of recall and precision are both between 0 and 1. When recall is equal to 1, no missing occurs. A higher value of precision represents a lower false alarm rate. For shot boundary detection, it is difficult to have an algorithm that can provide perfect segmentation in terms of human perception. Nevertheless, the algorithm can help pre-segment the video, and so reduce the workload of the human operator. Therefore, an algorithm that can provide no missed detection and the minimum false alarms is highly desirable. These results show that the precisions of all the approaches are low due to the existence of strong noise and motion in the video sequences, and approaches based on the HSV color space and CPAM achieve the highest recall values. More importantly, these two methods can achieve zero missing in video sequence 1. From the results with video sequence 1, HSV and all representations using CPAM can obtain the highest recall rate, and the joint ASP & CSP method also achieves the highest level of precision. Therefore, it is clear that the joint ASP & CSP method outperforms other

methods. From the results with video sequence 2, monochrome, HSV and joint ASP & CSP achieve the highest recall rate, while the joint ASP & CSP method also obtains the highest precision level. In other words, the joint ASP & CSP method achieves the best performance level in automatic scene cut detection.

Lgerca_1.mpg					
Low-level Representations	CD	FN	FP	Recall	Precision
Monochrome	41	1	73	0.9762	0.3596
Color Space					
256 RGB Color	23	19	88	0.5476	0.2072
64 RGB Color	21	21	125	0.5	0.1438
32x32 RgBy	39	3	51	0.9286	0.4333
HSV	42	0	66	1	0.3889
HMMD	41	1	44	0.9762	0.4824
CPAM					
ASP	42	0	115	1	0.2375
CSP	42	0	83	1	0.3360
Joint ASP & CSP	42	0	39	1	0.5063

Table 3.1 Performances of different low-level representations for scene cut detection based on video sequence 1.

Lgerca_2.mpg					
Low-level Representations	CD	FN	FP	Recall	Precision
Monochrome	38	4	162	0.9048	0.1900
Color Space					
256 RGB Color	28	14	82	0.6667	0.2545
64 RGB Color	24	18	134	0.5714	0.1519
32x32 RgBy	35	7	320	0.8333	0.0986
HSV	38	4	177	0.9048	0.1767
HMMD	36	6	133	0.8571	0.2130
CPAM					
ASP	32	10	94	0.7619	0.2540
CSP	37	5	222	0.8810	0.1429
Joint ASP & CSP	38	4	150	0.9048	0.2021

Table 3.2 Performances of different low-level representations for scene cut detection based on video sequence 2.

3.6 Conclusion

In this chapter, we have introduced the idea of using CPAM for scene representation and shot boundary detection. In our experiments, the entropic thresholding technique was used to determine the optimal threshold, and the two MPEG-7 video sequences were used to evaluate the performances of the CPAM and several histogram-based low-level representations. The experimental results show that the CPAM method outperforms other methods in terms of shot boundary detection. The CPAM does not only detect shot boundary accurately, but also improve efficiency of shot content representation of a shot due to the reduction of noise of suffered from neighborhood shot. In next chapter, we will present a new video content representation scheme based on global statistics for video shot retrieval. This is the next step of video signal analysis for CBVR.

A New Content Representation for Video Segment Retrieval

4.1 Introduction

In Chapter 3, we presented an efficient and reliable approach for shot boundary detection. Based on the detected boundary, the video can be temporally segment into shot. For content-based video retrieval system, the content of a shot is then represented by a feature vector or clustering [28, 51] is then applied for indexing, searching and retrieval. Color histogram representation is commonly used in image retrieval, which can also provide a motion invariant representation for video retrieval as mentioned in Chapter 2. Also, key frame approach and content-based approach for video segment representation have been reviewed. In this Chapter, we will present an optimal key frame representation scheme based on global statistics for video shot retrieval. Each pixel in this optimal key frame is constructed by considering the probability of occurrence of those pixels at the corresponding pixel position among the frames in a video shot. Therefore, this constructed key frame is called “Temporally Maximum Occurrence Frame” (TMOF), which is an optimal representation of all the

frames in a video shot. The retrieval performance of this representation scheme is further improved by considering the k pixel values with the largest probabilities of occurrence and the highest peaks at each pixel position for a video shot. The corresponding schemes are called k -TMOF and k -pTMOF, respectively. These key frame representation schemes are compared to other histogram-based techniques for video shot representation and retrieval. In the experiments, the three video sequences in the MPEG-7 content set were used to evaluate the performances of the different key frame representation schemes. Experimental results show that our proposed representations outperform the alpha-trimmed average histogram for video retrieval, which have been mentioned in Chapter 2.

4.2 Temporally Maximum Occurrence Frame (TMOF)

Traditionally, the key frames can be simply selected from predetermined temporal locations such as the first, middle or last frame. However, these selected frames may not be optimal in representing the corresponding video shots. Other methods extract an appropriate key frame based on the color/motion-based criteria. However, all these methods may still not provide an optimal representation of the video shots concerned. The alpha-trimmed average histogram can provide a robust description of a GoF, and is therefore considered an optimal histogram representation. However, it possesses the same drawback as the histogram-based representation – it cannot provide spatial information about the frame represented by the histogram. The spatial information or structure of a GoF is particularly important for video retrieval because the query usually has a similar structure to the frames in the GoF. Therefore, it is desirable to construct a key frame or extract a key frame representation which contains all the visually important information within the video shot. The algorithm proposed in this paper is to construct a key frame representation which contains most of the significant visual contents in a shot. This idea is illustrated in Figure 4.1, where a video shot has six frames.

Within this shot, a vehicle stops at the bottom left corner and a helicopter is landing during the first three frames, then the vehicle drives away and the helicopter remains on the roof-top during the last three frames. Obviously, the representative frame for this video shot should contain both the vehicle stopping near the house and the helicopter stopping on the house, as shown in Figure 4.1(g). However, existing approaches cannot capture all this important information in the frames.

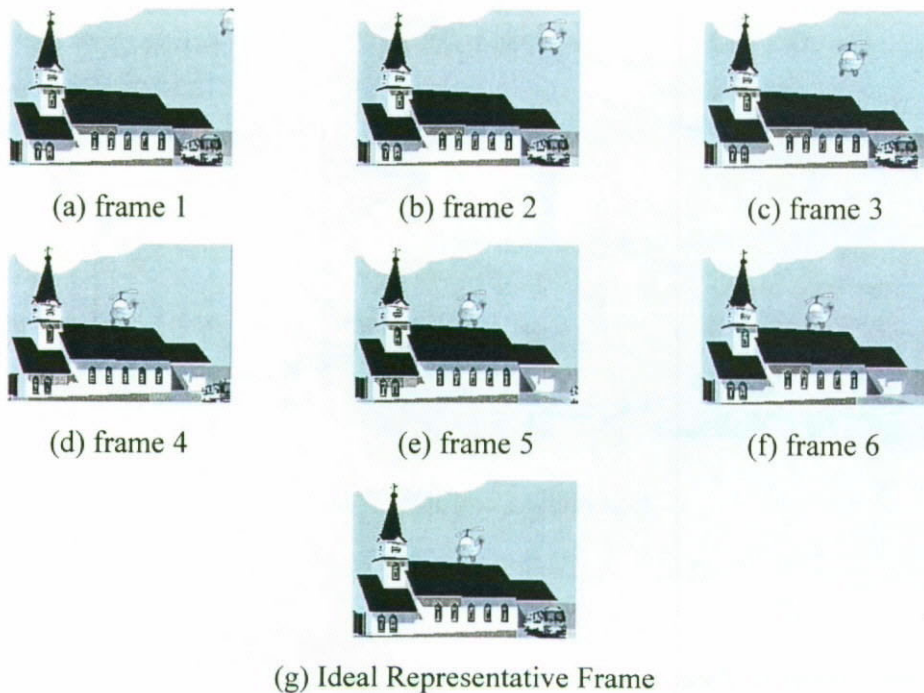


Figure 4.1 An ideal representative frame for a shot with six frames.

In view of this, our proposed representative frame, called the “Temporally Maximum Occurrence Frame” (TMOF), is constructed based on the probability of occurrence of pixel values at each pixel position for all the frames within a video shot.

Figure 4.2 illustrates the simplified algorithm of how to obtain our proposed representative frame.

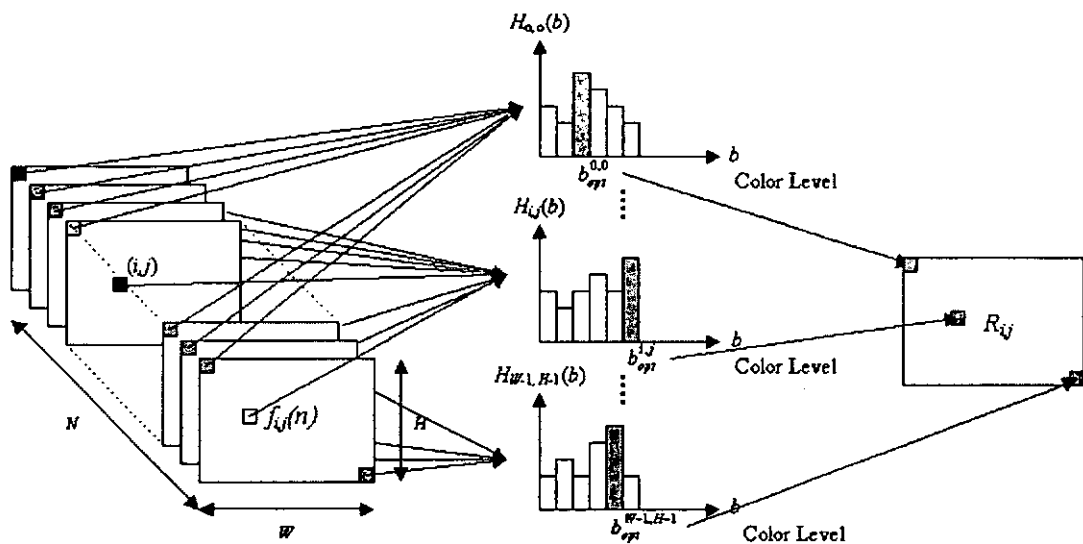


Figure 4.2 The construction of the TMOF for a video shot.

A histogram is formed based on the pixel values at each corresponding pixel position in the GoF, and is smoothed using a Gaussian function. Then, the value at a pixel position in a TMOF is the bin value whose frequency of occurrence, or count, is a maximum in the smoothed histogram. Therefore, the pixels of the TMOF are computed as follows:

$$TMOF(i, j) = b_{opt}, \quad 0 \leq i \leq W - 1 \quad \text{and} \quad 0 \leq j \leq H - 1, \quad (4.1)$$

where $W \times H$ is the size of a frame, and b_{opt} is chosen as follows:

$$b_{opt} = \arg \max_b \{H'_{i,j}(b)\}, \quad \text{for } 0 \leq b \leq B, \quad (4.2)$$

The smoothed histogram is obtained by using a Gaussian filter as follows:

$$H'_{i,j}(b) = H_{i,j}(b) * G(\sigma, b),$$

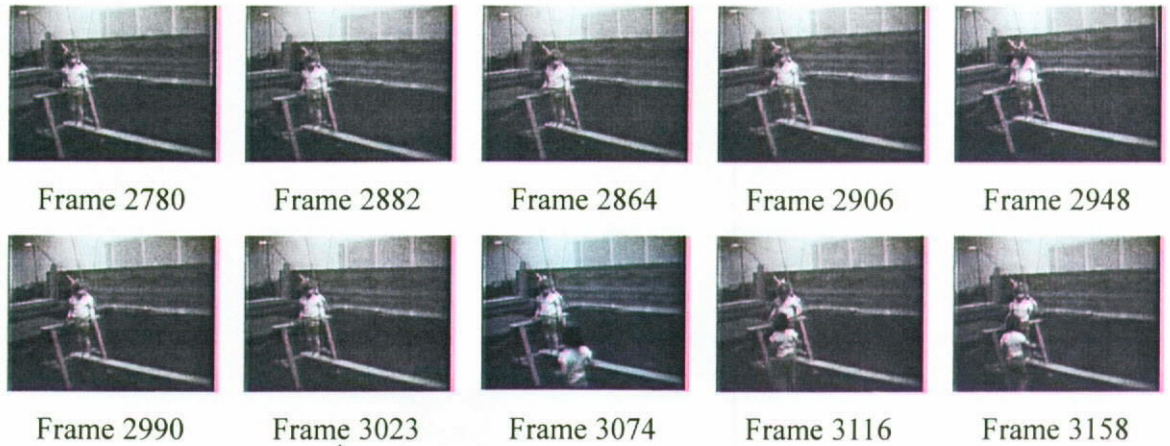
where $G(s, b)$ is a Gaussian function with variance s ,

$$H_{i,j}(b) = \sum_{n=0}^{N-1} \delta(f_n(i, j) - b), \quad \text{for } 0 \leq b \leq B, \quad (4.3)$$

$$\text{and } \delta(m - n) = \begin{cases} 1 & \text{for } m = n, \\ 0 & \text{for } m \neq n. \end{cases}$$

$H_{i,j}$ is the histogram formed by the corresponding pixels at pixel position (i, j) , $f_n(i, j)$ represents the pixel level at coordinates (i, j) in frame n , N is the total number of frames in the GoF, and B is the number of bins in the histogram. Normally, the number of bins in a histogram is equal to the number of intensity levels for a pixel. However, in order to maintain a good estimation of the value in the TMOF when the video shot is short in length, the number of bins for the histogram of a pixel position may be decreased, depending on the variations of the bin values in the histogram. In our experiments with 3 video sequences, the shortest video segment contains 48 frames, while the average shot length is 282. The frames within a GoF are similar to each other, so are the corresponding pixels at the same pixel position. Therefore, the value at a pixel position in the TMOF can be determined accurately even if the shot length is

short. Figures 4.3 and 4.4 illustrate two video shots extracted from a home video and a news program, and their corresponding TMOFs. The video shot shown in Figure 4.3 is rather static. The background and the girl shown in the video shot are clearly extracted to construct the TMOF. The video shot demonstrated in Figure 4.4 comprises both camera motions and object motions. The camera is following a group of people who are walking. Our proposed schemes can also effectively extract the major objects from the video shot to form the TMOF.



(a) The video shot from a home video.



(b) Representative frame (TMOF) constructed with the pixel value with the maximum frequency of occurrence or the highest peak at each pixel position.

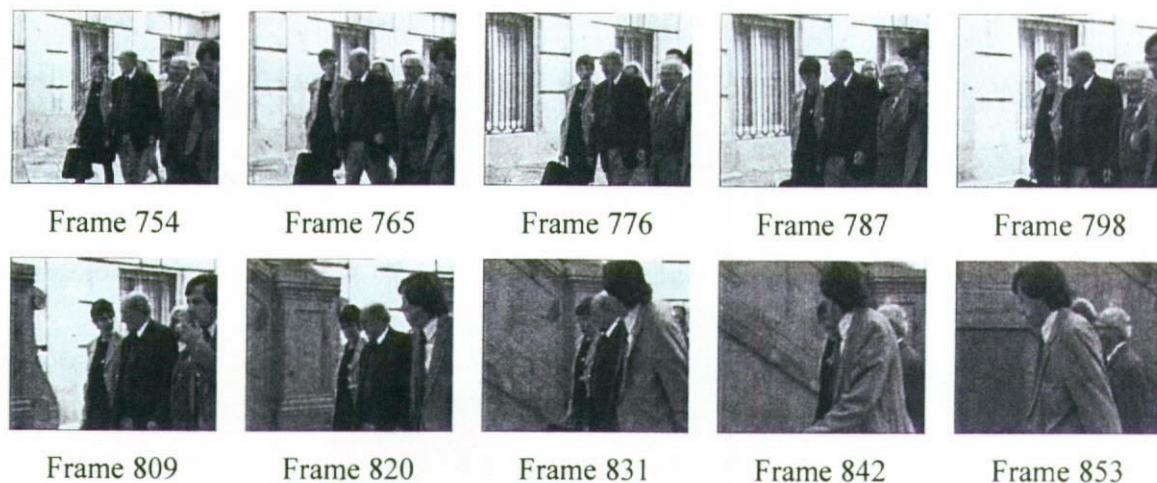


(c) Representative frame of a GoF constructed with the pixel value with the second maximum frequency of occurrence at each pixel position for 2-TMOF.



(d) Representative frame of a GoF constructed with the pixel value of the second highest peak at each pixel position for 2-pTMOF.

Figure 4.3 The video shot, and its TMOF/2-TMOF/2-pTMOF of a home video.



(a) The video shot from a news program



(b) Representative frame (TMOF) constructed with the pixel value with the maximum frequency of occurrence or the highest peak at each pixel position.



(c) Representative frame of a GoF constructed with the pixel value with the second maximum frequency of occurrence at each pixel position for 2-TMOF.



(d) Representative frame of a GoF constructed with the pixel value of the second highest peak at each pixel position for 2-pTMOF.

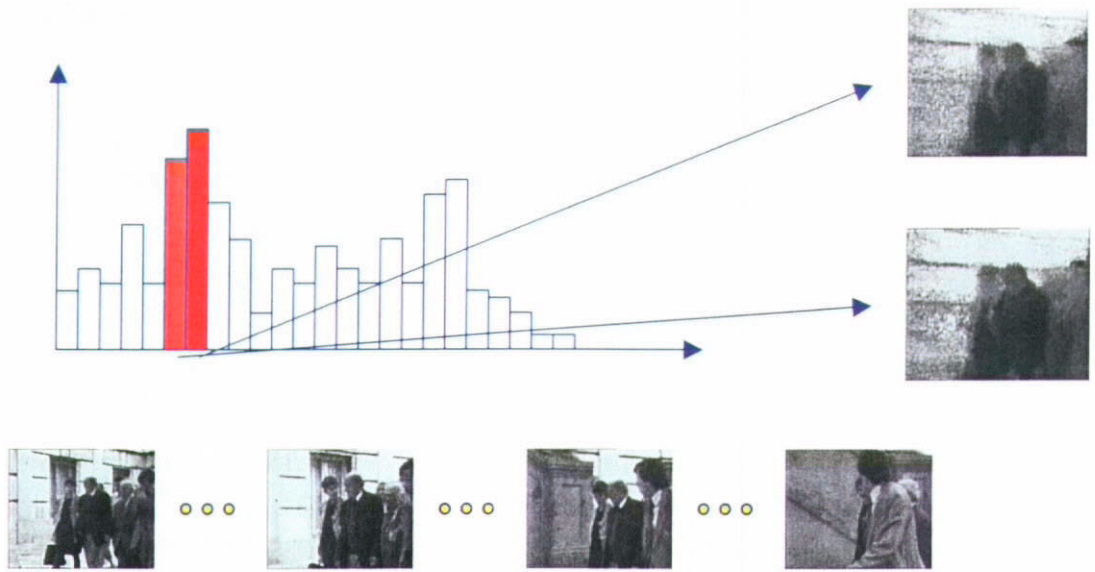
Figure 4.4 The video shot, and its TMOF/2-TMOF/2-pTMOF of a news program.

To further improve the representation power of TMOFs, we propose that each pixel position of the TMOF contains k possible values. Two different types of k values are proposed. The first set is the corresponding k pixel values with the maximum frequencies of occurrence at the pixel position. This representation scheme is called “ k -TMOF”. The second type is where the k values are equal to the k highest peaks in a histogram. A peak occurs when a bin value is higher than its two adjacent bins. Suppose that $H_{i,j}(b)$ represents the value of bin k at pixel position (i, j) . Then $H_{i,j}(b)$ is a peak if $H_{i,j}(b) > H_{i,j}(b-1)$ and $H_{i,j}(b) > H_{i,j}(b+1)$. If m consecutive histogram bins have the same value while this value is larger than that of the two adjacent bins of these m bins, i.e. $H_{i,j}(b) = H_{i,j}(b+1) = \dots = H_{i,j}(b+m-1)$, and $H_{i,j}(b) > H_{i,j}(b-1)$ and $H_{i,j}(b) > H_{i,j}(b+m)$, then the middle value of these m consecutive bins is defined as the peak,. This scheme is called “ k -pTMOF”. These representations have much more power than those with a single value for each pixel position. When $k = 1$, both k -TMOF and k -pTMOF will be equivalent to the TMOF. Figures 4.3(c) and 4.4(c) illustrate the representations constructed based on the 2nd pixel values with the maximum frequencies of occurrence (i.e. 2-TMOF), while Figures 4.3(d) and 4.4(d) show the corresponding representations based on the 2nd highest peaks (i.e. the 2-pTMOF). The two representative frames shown for $k = 2$ for each of the two schemes are only two of the possible frames that can be represented. Figure 4.5 illustrates the selections of

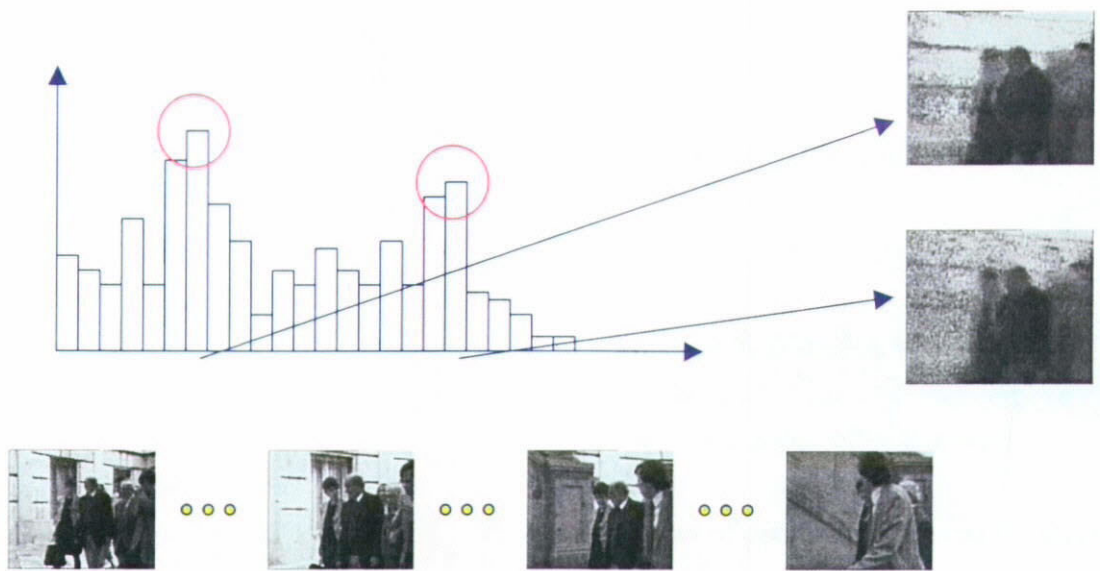
pixel values for 2-TMOF and 2-pTMOF. Suppose that the size of the frame is $H \times L$, the number of possible frames that can be generated from a k -TMOF or k -pTMOF is k^{HK} . To measure the distance between a query input and a k -TMOF/ k -pTMOF, we have to consider the distance between each pixel in the query and the corresponding k values in the k -TMOF/ k -pTMOF. The proposed representation schemes are trying to extract the most important color information at each pixel position within a shot for content-based video retrieval. The k values at each pixel position represent the most representative pixel values among the frames in a GoF. Among the k values, the one nearest to the pixel value of the query input should be selected in measuring the distance between the query and the key-frame representation. Therefore, to measure the distance between the query input and representative frame, the minimum distance measure D_{min} is used and is defined as follows:

$$D_{min}(l) = \sum_{i=0}^H \sum_{j=0}^W \min(|q(i, j) - R_l(i, j, u)|), \quad \text{for } u = 1, \dots, k, \quad (4.4)$$

where $q(i, j)$ and $\{R_l(i, j, u), u = 1, \dots, k\}$ represent the pixel intensities of the query at (i, j) and the k values of the k -TMOP or k -pTMOP of the l^{th} video shot in the database, respectively.



(a) Selection of pixel values using 2-TMOF.



(b) Selection of pixel values using 2-pTMOF.

Figure 4.5 Selections of pixel values using 2-TMOF and 2-pTMOF.

4.3 Computational Complexity Analysis

In this section, the computational complexities of the TMOF and the alpha-trimmed average histogram will be analyzed. For both of these approaches, the computations required are divided into two parts: one is for the construction of the key frame representation, which is an offline process, while the other one is the computations required for retrieval. For the alpha-trimmed average histogram, construction of the representative histogram can be divided into 3 parts: histogram constructions, sorting for each bin, and averaging. The number of operations required to construct the histograms in a GoF having N frames with frame size of $W \times H$ is in the order of $O(N \cdot W \cdot H)$. The corresponding values for the same bin of the N histograms are sorted, and the computation required is in order of $O(N_h \cdot N \log_2 N)$, where N_h is the number of bins in a histogram. Finally, depending on the value of α , the average of $(1 - 2\alpha)N$ numbers is computed for a bin and the order of operations required for this part is $O((1 - 2\alpha)NN_h)$. Therefore, the total computations required to construct the alpha-trimmed average histogram of a GoF is $O(N \cdot W \cdot H) + O(N_h \cdot N \log_2 N) + O((1 - 2\alpha)NN_h)$.

For the TMOF, the process of generating a key frame representation for a GoF can also be divided into 3 parts: down-scaling a frame to a size of N_h , histogram construction for each pixel position, and determining the pixel value with the

maximum count. The down-scaling process divides a frame into N_h regions and computes the average values for each of these regions. Therefore, the computation required is in the order of $O(W \cdot H \cdot N_h)$. The total number of operations for constructing the histograms at each pixel position is $O(N_h \cdot N)$. To construct the TMOF for a GoF, the number of operations required to determine the pixel values with maximum count is in the order of $O(N_h \cdot N)$. Therefore, the total computation required for the TMOF is $O(W \cdot H \cdot N_h) + O(N_h \cdot N) + O(N_h \cdot N)$. In general, the computations required by the two approaches are very similar. Experimental results show that the average runtimes are 1.094 seconds and 1.012 seconds per GoF for constructing the representation for an alpha-trimmed average histogram and a TMOF, respectively.

For retrieval, the computations required are much less than those required to construct the representative frames. For a histogram-based distance measure, the computation for comparing a query to a representation of a GoF is in the order of $O(N_h)$. With the k -TMOF and k -pTMOF schemes, the dimension of the feature representation is also N_h , but there are k values for each of the N_h elements. Therefore, the computation required is $O(kN_h)$ for a GoF.

4.4 Experimental Results

Our experiments were conducted based on 427 shots extracted from the three video sequences of the MPEG-7 content set. The selected video data consist of a complete news program and two edited home videos. The shot boundaries have been identified manually, and queries are selected as the first, middle and last frames of each shot. Each segmented video shot is represented by different representation schemes, which include the alpha trimmed average histogram, TMOF, k -TMOF, k -pTMOF, histogram representations of k -TMOF and k -pTMOF, etc. Video retrieval is then based on the measured distances between the feature vectors of the query and each video shot represented by the different representation schemes as follows:

$$D_q(i) = d(V_q, V_i) \quad \text{for } i = 1, 2, 3, \dots, N, \quad (4.5)$$

where V_q is the feature vector of the query, V_i the feature vector representing the i -th video shot, and N the total number of video shots considered.

To evaluate the respective performances of the key frame representation schemes, the Average Recall (AR) and the Average Normalized Modified Retrieval Rank ($ANMRR$) [2] are used. These were developed by the MPEG Video Group for the evaluation of MPEG-7 core experiments. The range of both AR and $ANMRR$ is between 0 and 1. These measures determine the number of correct GoFs retrieved and where they rank among the retrievals. To measure these two terms, the following

parameters are defined:

$ng(q)$ denotes the number of ground truth GoFs for a query q , and

$nr(q)$ denotes the number of correctly retrieved items in the top K retrievals,

where $GTM = \max\{ng(q)\}$, is the maximum number of ground truth GoFs over all

defined queries and $K = \min\{4 \times ng(q), 2 \times GTM\}$.

The recall for query q and the average recall are then computed as follows:

$$R(q) = \frac{nr(q)}{ng(q)} \text{ and } AR = \frac{1}{Q} \sum_{q=1}^Q R(q), \quad (4.6)$$

respectively. A higher value of AR implies a better retrieval performance. To obtain the

$ANMRR$, the Average Retrieval Rank (ARR) should be computed first.

$$ARR(q) = \sum_{i=1}^{ng(q)} \frac{r(i)}{ng(q)}. \quad (6.7)$$

Then the modified retrieval rank is computed as follows:

$$MRR(q) = ARR(q) - \frac{ng(q)}{2} - 0.5, \quad (6.8)$$

where $r(i) = \begin{cases} \text{Rank} & \text{Each of the } ng(q) \text{ in the top } K \text{ retrievals,} \\ K + 1 & \text{Otherwise.} \end{cases}$

The $MRR(q)$ is then normalized to the range $[0,1]$ to obtain the $NMRR(q)$, as shown

below:

$$NMRR(q) = \frac{MRR(q)}{K - \frac{ng(q)}{2} + 0.5}. \quad (6.9)$$

The average of $NMRR(q)$ is calculated as follows:

$$ANMRR = \frac{1}{Q} \sum_{q=1}^Q NMRR(q). \quad (6.10)$$

A lower value of $ANMRR$ implies a higher retrieval rate, with the relevant items

ranked at the top positions.

In the experiments, the proposed representation schemes were compared with the optimal key frame histogram [1] and the alpha-trimmed average histogram with six different values of the alpha parameter. For histogram-based representations, the L_1 distance measure is used because it can provide the best result in video retrieval [1]. Our experiments are divided into three parts. First, gray-level alpha-trimmed histograms and the optimal key frame histograms with different numbers of bins are evaluated to obtain the optimal histogram for video retrieval. Our proposed schemes are then compared with the optimal histogram-based representation scheme. Second, the minimum distance measure is used to evaluate the representation performances of the k -TMOP and k -pTMOF with different values of k for video retrieval. Finally, the respective performances of our proposed schemes using histogram representation are then evaluated and compared with that of the optimal histogram-based representation scheme based on the first set of experiments.

4.4.1 Optimum alpha-trimmed average histogram and TMOF

The objective of this set of experiments is to compare the performances of our proposed representation scheme (TMOF) and the optimal histogram-based representation for video retrieval. To obtain the optimal histogram-based representation, we used two different numbers of bins for the optimal key frame histogram and the alpha-trimmed average histogram with six different values of $\alpha = \{0, 0.1, 0.15, 0.20, 0.25, 0.50\}$. To compare the performances of TMOF and histogram-based schemes, each frame for TMOF is divided into a number of blocks of equal size and each block is represented by its corresponding mean. This number is set as equal to the number of bins used in the histogram representation, so we can compare the performances of the different key frame representations with the same feature vector size. Table 4.1 shows the *AR* and the *ANMRR* of the respective histogram representation schemes with two different numbers of bins. The number of queries used is 1281. The alpha-trimmed average histogram achieves the best performance when $\alpha = 0.15$ with 256 bins. The corresponding *AR* and *ANMRR* values are 0.5950 and 0.4451, respectively. In addition, TMOF outperforms the histogram-based representations, whose *AR* and *ANMRR* values are 0.7088 and 0.3315, respectively. Figures 4.6 and 4.7 illustrate the *ANMRR* and *AR* of our proposed representation scheme and the optimal alpha-trimmed average histogram, while the

number of queries was varied from 1 to 1281. Experimental results show that our proposed representations result in a slightly lower *ANMMR* and a higher *AR* compared to the alpha-trimmed average histogram, i.e. the retrieval performance of TMOF is better than that of the optimal alpha-trimmed average histogram.

	ANMRR	AR
L_1 Distance with 256 bins		
Average Histogram	0.4520	0.5885
0.10 Alpha Histogram	0.4464	0.5942
0.15 Alpha Histogram	0.4451	0.5950
0.20 Alpha Histogram	0.4462	0.5944
0.25 Alpha Histogram	0.4468	0.5957
Median Histogram	0.4453	0.5968
Optimal Key Frame Histogram	0.4593	0.5735
L_1 Distance with 128 bins		
Average Histogram	0.4518	0.5872
0.10 Alpha Histogram	0.4461	0.5957
0.15 Alpha Histogram	0.4460	0.5952
0.20 Alpha Histogram	0.4453	0.5948
0.25 Alpha Histogram	0.4453	0.5955
Median Histogram	0.4468	0.5965
Optimal Key Frame Histogram	0.4562	0.5770
TMOF	0.3315	0.7088
3-TMOF	0.2912	0.7545
3-pTMOF	0.2609	0.7891

Table 4.1 The respective performances of the optimal key frame histogram, the alpha-trimmed average histograms with $\alpha = \{0, 0.10, 0.15, 0.20, 0.25, 0.5\}$, and the TMOF, the 3-TMOF, and the 3-pTMOF.

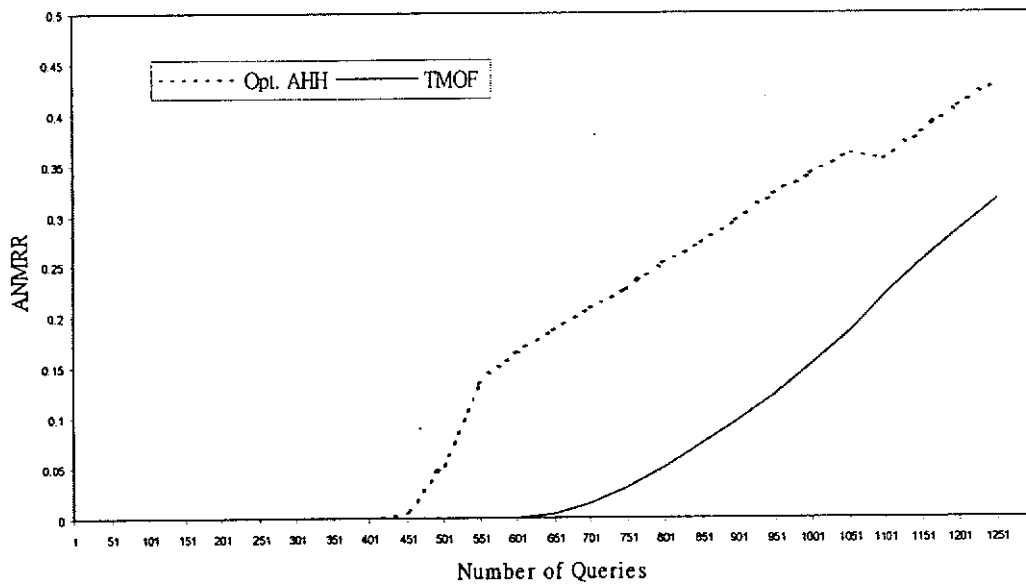


Figure 4.6 The overall average normalized modified retrieval rank of TMOF and the optimum alpha-trimmed average histogram when the number of queries varied from 1 to 1281.

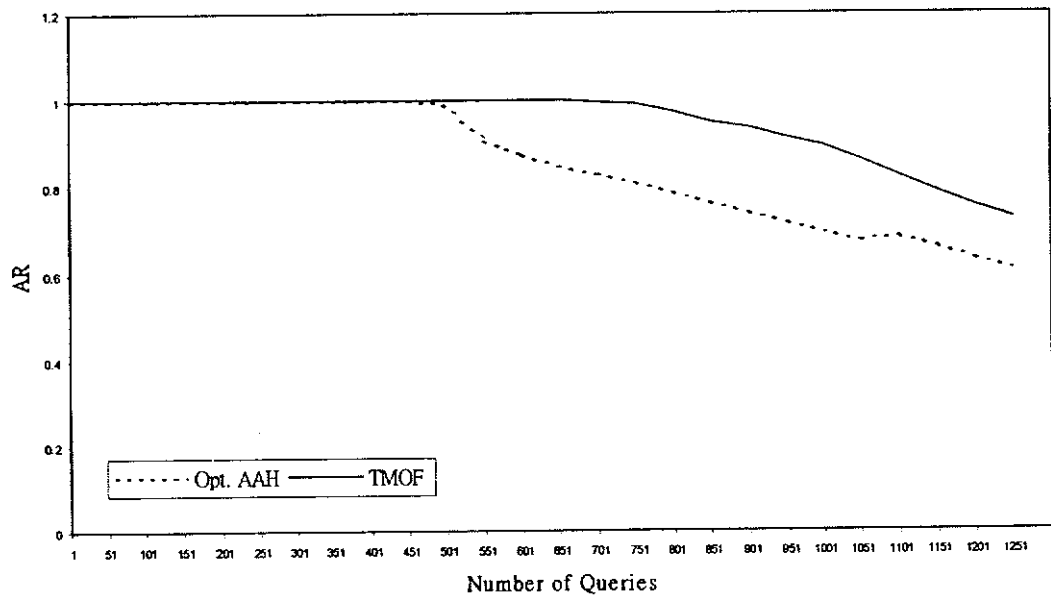
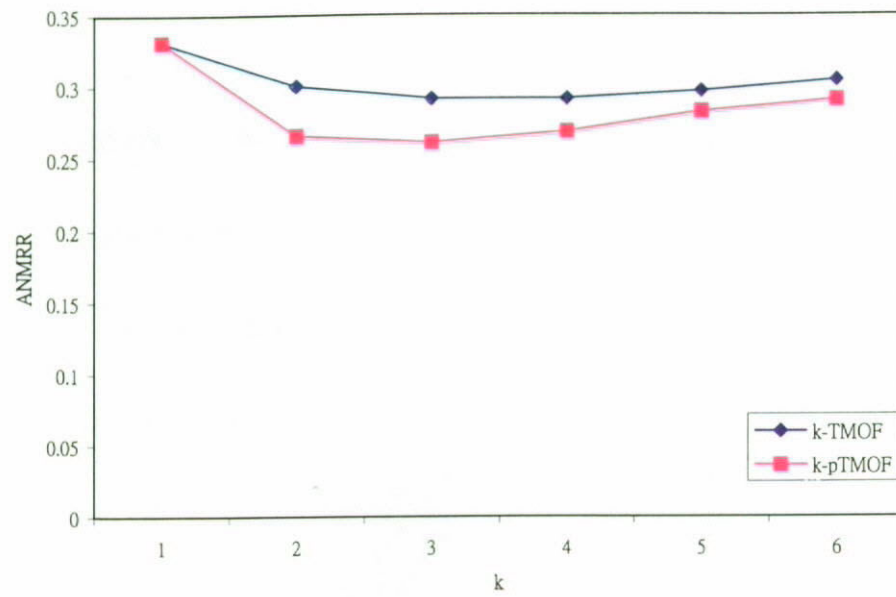


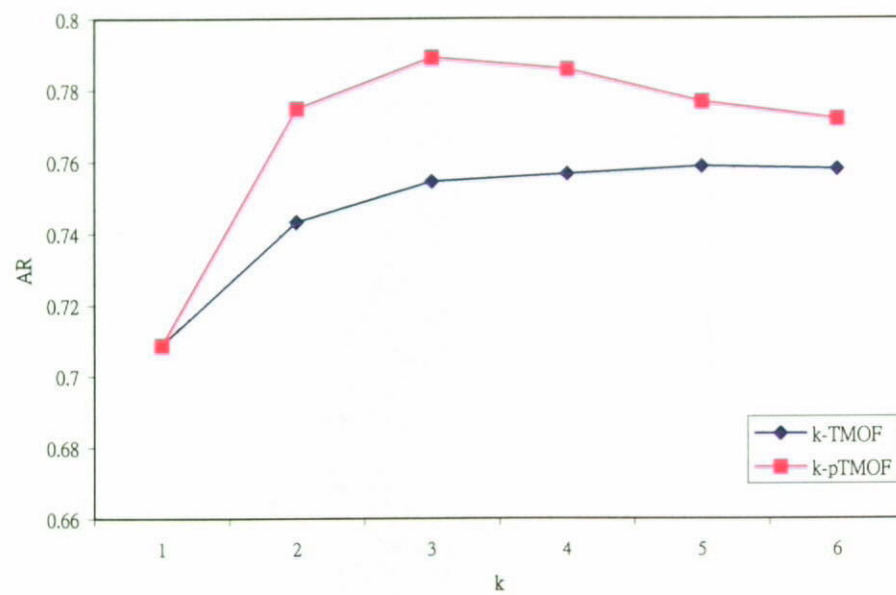
Figure 4.7 The overall average recall of TMOF and the optimum alpha-trimmed average histogram when the number of queries varied from 1 to 1281.

4.4.2 Performances of k -TMOF and k -pTMOF

In this set of experiments, we evaluate the performances of k -TMOF and k -pTMOF using the minimum distance measure with different values of k . Both the k -TMOF and k -pTMOF are divided into 16×16 blocks to form a 256-D feature vector. Experimental results with different values of k are illustrated in Figure 4.8. When the value k increases, the representational powers of both the k -TMOF and k -pTMOF schemes become higher. However, when k is larger than a certain value, it is found that the performance will be degraded. The reason is that when k is large, the number of false alarms will also be higher, so the performance degrades. Experimental results show that these two representation schemes achieve the best performance when k is 3.



(a) Average Normalized Modified Retrieval Rank



(b) Average Recall

Figure 4.8 The performances of k -TMOF and k -pTMOF using the minimum distance measure with different values of k .

4.4.3 Histogram Representation of k -TMOF and k -pTMOF

The k -TMOF and k -pTMOF can also be represented by histograms for video retrieval. According to the number k , each of the corresponding histogram bins will contain k different values in their representations. However, the design of k -TMOF and k -pTMOF is not optimal for histogram representation, and the spatial information is also lost. Therefore, the histogram representations for k -TMOF and k -pTMOF will result in a degradation of retrieval performance. Nevertheless, the k -TMOF and k -pTMOF histograms can still achieve comparable performances to the optimal alpha-trimmed average histogram.

In this experiment, our proposed schemes are compared to the 256-bin alpha-trimmed average histogram with $\alpha = 0.15$. As illustrated in the first experiment, the best performance can be achieved with this setting. Table 4.2 shows the performances of the k -TMOF and k -pTMOF with 256 bins, and that of the optimal alpha-trimmed average histogram. From Table 4.2, the k -pTMOF with k higher than 1 outperforms the optimal alpha-trimmed average histogram in terms of the retrieval performances, while the optimal alpha-trimmed average histogram achieves a better performance level than that of the k -TMOF for all k .

k	ANMRR	AR
k-TMOF Histograms		
1	0.5363	0.5110
2	0.4689	0.5942
3	0.4532	0.6056
4	0.4513	0.6114
5	0.4464	0.6200
6	0.4475	0.6167
k-pTMOF Histograms		
1	0.5248	0.5228
2	0.4400	0.6201
3	0.4209	0.6317
4	0.4081	0.6478
5	0.4027	0.6525
6	0.4058	0.6489
0.15 Alpha Histogram with 256 bins		
	0.4451	0.5950

Table 4.2 The performances of the k -TMOF and k -pTMOF represented as 256-bin histograms, and that of the optimum alpha-trimmed average histogram.

4.5 Conclusion

In this chapter, we have presented a new representative frame, namely “Temporally Maximum Occurrence Frame” (TMOF), for video retrieval. This TMOF can capture the most significant visual contents within a video shot. The representational power of the TMOF is further enhanced by considering the k most frequently-occurring values and the k highest peaks of the probability distribution at each of its pixel positions. Our proposed schemes are compared with the family of alpha-trimmed average histograms, which is a video segment descriptor in MPEG-7. Our proposed schemes have a similar computational complexity as compared to the alpha-trimmed average histogram method. In our experiments, 427 shots extracted from the four MPEG-7 content set video sequences were used to evaluate the respective performances of the schemes. Experimental results show that both the k -TMOF and k -pTMOF can achieve the best performance when $k = 3$, and this scheme outperforms the alpha-trimmed average histogram representation for video retrieval.

5.1 Conclusion

In this thesis, we have provided an overview of CBVR and a variety of existing techniques for video structure parsing and video content representation using visual features for CBVR. For shot boundary detection, the spatial matching detection, twin-comparison approach, statistical approach, histogram-based approach, edge-based approach, spatio-temporal slice approach and compressed domain approach have all been reviewed. For abrupt transition, the spatial matching approach, histogram-based approach, and edge-based approach can achieve a good detection result while the video does not consist of strong motion and noise. For gradual transition, the twin-comparison approach is a simple and effective method, but it cannot classify the type of gradual changes. The statistical approach is particularly designed to detect dissolve, fade-in and fade-out based on the equation for generating this kind of special effect. The spatio-temporal slice approach captures a collection of scans in the same selected position of every frame as a function of time. Abrupt and gradual transitions can be detected by analyzing the rhythm of slices.

For Video content feature extraction, key-frame-based representation and shot-based representation were presented in Chapter 2. Key-frame-based representation selects one or more frames from a video shot to represent that shot by various criteria, such as color change, motion energy, etc. Shot-based representation extracts a feature vector from a video shot by considering its statistical information or captured spatio-temporal slices to represent the video shot for video retrieval.

In our research, we have proposed efficient methods for shot boundary detection and video content representation for video retrieval. Our efficient approach for detecting a shot boundary consists of two stages. Firstly, each frame of a video is represented by a feature vector, which can be the color histogram in different color space or our proposed CPAM histogram. The differences between two consecutive frames are then calculated by mean of their feature vectors. Entropic thresholding is then applied to determine the optimal threshold for shot boundary detection. A shot boundary can be identified by thresholding based on the determined threshold. Experimental results show that our method can provide a better performance level than other histogram-based methods. The major advantage of this method is that it is less sensitive to lighting conditions and motion.

A new video content representation scheme has also been proposed. This scheme constructs a new frame or a set of new frames to represent a video shot by considering

the probability of occurrence of those pixels at the corresponding pixel position among the frames in a video shot. This scheme is also compatible with existing histogram representation methods. Experimental results show that this scheme can provide a better performance level than that of alpha-trimmed histogram representation.

In conclusion, we have developed different techniques for representing video frames, detecting shot boundaries, and representing video shots. To facilitate the use of these techniques, we have built a software library based on the work for this thesis. The library is called “ Video Retrieval Library” and an overview of this library is presented in the Appendix. The major purpose of this library is for users to develop their own video retrieval techniques and systems easily and efficiently based on this library.

5.2 Future works

The employment of visual information to strengthen video management systems is an important application. In this thesis, we have reviewed various methods and have proposed two methods for the video management system: one for shot boundary detection and the other for video content representation. However, these methods involve visual information only. The performance still suffers from strong motion and noise. As the audio signal is supplementary to the video signal, the retrieval performance of the overall system can be further improved by considering the audio and visual information jointly for video analysis. Audio information can also be used in video parsing because the characteristics and style of audio from different programs are usually not similar. Based on speech recognition, we can also identify an actor or actress from a video. Therefore, by identifying the correlation of the audio and visual features, a more reliable and efficient CBVRS can be developed.

Appendix

Overview of the Video Retrieval Library (VideoRetL)

A software library called Video Retrieval Library (VideoRetL) was developed based on this study. The objective of this library is to help application developers to build a video management system efficiently. The VideoRetL provides function calls for video frame feature extraction, video scene break detection, and video shot representation for video retrieval.

The VideoRetL provides algorithms for video management and retrieval. In this library, function calls for representing the frames in a video, calculating frame difference between two frames, segmenting a video into a series of video shots, representing video shots for retrieval, etc. are provided. The following is a list of classes and their function calls provided in this library:

CFrameDescriptor:

RGBHist(), HSVHist(), HMMDHist(), OppHist(), CPAMHist().

CHistDistanceMeasure:

HistTotalCount(), NormalHist(), NL1HistDistance(), NL2HistDistance(), NHistIntersect(), NHistChi2().

CThresholdDetermination:

EntropicThreshold().

CSceneBreakDetection:

FrameHistDiff(), FramePixelDiff(), FrameBlockDiff(), SceneBreakThresholding().

CShotRepresentation:

OptKeyFrameHist(), AlphaTrimAvgHist(), TMOF(), TempVariKeyFrame().

CFrameDescriptor

The Class CFrameDescriptor provides the function calls for frame description.

The color histograms or descriptors may be constructed based on different color spaces, such as RGB, HSV, YCrCb, and HMMD. The opponent color space can also represent an image well for image indexing, and its transformation from the RGB color space is simple. The details of these color spaces can be found in Chapter 3. In this Class, the function calls provided for extracting these frame descriptors based on the RGB histogram, HSV histogram, HMMD histogram, opponent color histogram and CPAM histogram, are *RGBHist()*, *HSVHist()*, *HMMDHist()*, *OppHist()*, *CPAMHist()*, respectively.

CHistDistanceMeasure:

Two color images with corresponding histograms I and Q can be compared using different metrics metrics. In this Library, the distance measures provided include the L_1 , L_2 , normalized histogram intersection, and χ^2 . The respective distance measures are defined as follows:

$$L_1\text{-distance: } d_{L_1}(\mathbf{I}, \mathbf{Q}) = \sum_{i=1}^n |I_i - Q_i| \quad (\text{A-1})$$

$$L_2\text{-distance: } d_{L_2}(\mathbf{I}, \mathbf{Q}) = \sum_{i=1}^n (I_i - Q_i)^2 \quad (\text{A-2})$$

$$\text{Intersection:} \quad d_{\text{int}}(\mathbf{I}, \mathbf{Q}) = \sum_{i=1}^n \min(I_i, Q_i) \quad (\text{A-3})$$

$$\chi^2: \quad \chi^2(\mathbf{I}, \mathbf{Q}) = \sum_{i=1}^n \frac{[I_i - Q_i]^2}{I_i + Q_i} \quad (\text{A-4})$$

In this Class, the function calls provided for computing these histogram distance measures: the L_1 , L_2 , normalized histogram intersection, and χ^2 , are *NL1HistDistance()*, *NL2HistDistance()*, *NHistIntersect()*, and *NHistChi2()*, respectively. To compare images of different sizes, the count for each bin is divided by the total number of pixels in the image. Hence, this normalized color histogram can be considered as the probability density function of the color values. Two function calls are provided to normalize a histogram. The function call, *HistTotalCount()*, calculates the sum of counts of all the bins in a histogram, and *NormalHist()* normalizes a histogram based on its total count.

CSceneBreakDetection

In this Class, three simple methods for detecting scene break are provided. *FramePixelDiff()*, *FrameBlockDiff()* and *FrameHistDiff()* are used to detect scene break by comparing the difference in pixel intensities of corresponding pixels, the statistical characteristics of corresponding regions, and histogram difference between two successive frames, respectively. The function call *SceneBreakThresholding()* is provided to determine scene breaks by thresholding.

CThresholdDetermination

The entropic thresholding has been described in Chapter 3. In this Class, the function call *EntropicThreshold()* is provided to determine this optimal threshold value for scene break detection.

CShotRepresentation

In this Class, the function calls for four shot representation schemes are provided. The function calls, *OptKeyFrameHist()*, *AlphaTrimAvgHist()* and *TMOF()*, are provided to extract the optimal key frame histogram, the alpha trimmed average histogram and the temporal maximum occurrence frame of a shot, which have been described in Chapter 4. The function call, *TempVariKeyFrame()*, is used to determine the key frames in a video shot based on a specific threshold. This function call takes the first frame of each shot as a key frame. The differences of the image features between this key frame and the consequent frames are computed. The farther away the current frame is from the key frame, the larger the difference should be. Whenever this difference is larger than a certain threshold, the corresponding current frame will be considered to be a key frame.

References

- [1] D. Feng, W. C. Siu and H. J. Zhang, "Multimedia Information Retrieval and Management: Technological Fundamentals and Applications" Springer, 2003.
- [2] ISO/IEC Standard 15938-Part 3: Information Technology – Multimedia Content Description Interface: Visual, 2002.
- [3] S. Antani, R. Kasturi and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video", Pattern Recognition, Vol. 35, No. 4, pp. 945-965, Apr. 2002.
- [4] U. Gargi, R. kasturi and S. H. Strayer, "Performance Characterization of Video-Shot-Change Detection", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, No. 1, pp. 1-13, Feb. 2000.
- [5] M. Swain and D. Ballard, "Color Indexing", Computer Vision, Vol. 7, No. 1, pp. 11-32, 1991.
- [6] S. F. Chang, W. Chen, H. J.Meng, H. Sundaram and D. Zhong; "A fully automated content-based video search engine supporting spatiotemporal queries", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, No. 5, pp. 605-615, Sept, 1998.
- [7] Y. Wang; Z. Liu and J. C. Huang, "Multimedia Content Analysis-Using Both Audio And Visual Clues", IEEE Signal Processing Magazine, Vol. 17 Iss. 6, Nov. 2000.
- [8] D. Li, I. K. Sethi, N. Dimitrova and T. McGee , "Classification Of General Audio Data For Content-Based Retrieval", Pattern Recognition Letters, Vol. 22, Iss. 5, pp. 533-544, Apr. 2001.

- [9] J. M. Thong, P. J. Moreno, B. Logan, B. Fidler, K. Maffey and M. Moores, "Speechbot: an experimental speech-based search engine for multimedia content on the web", *IEEE Transactions on Multimedia*, Vol. 4, pp. 88-96, Mar. 2002.
- [10] C. L. Huang and B. Y. Liao, "A Robust Scene-Change Detection Method for Video Segmentation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 12, pp. 1281-1288, Dec. 2001.
- [11] J. Yu and M. D. Srinath, "An Efficient Method for Scene Cut Detection", *Pattern Recognition Letters*, Vol. 22, Iss. 13, pp. 1379-1391, Nov. 2001.
- [12] M. S. Lee, Y. M. Yang and S. W. Lee, "Automatic video parsing using shot boundary detection and camera operation analysis," *Pattern Recognition*, Vol. 34, No. 3, pp. 711-719, Mar 2001.
- [13] H. S. Chang, S. Sull and S. U. Lee, "Efficient Video Indexing Scheme for Content-Based Retrieval", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 8, pp. 1269-1279, Dec. 1999.
- [14] E. H. Adelson and J. Bergen, "Spatiotemporal energy models for the perception of motion", *J. Optical Soc. Amer.*, vol. 2, No. 2, pp. 284-299, Feb. 1985.
- [15] C. W., T. C. Pong and R. T. Chin, "Video Partitioning by Temporal Slice Coherency", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 8, pp. 941-953, Aug. 2001.
- [16] Chong-Wah Ngo, Ting-Chuen Pong, Hong-Jiang Zhang, "On Clustering and Retrieval of Video Shots Through Temporal Slices Analysis", *IEEE Transactions on Multimedia*, Vol. 4, No. 4, pp. 446-458, Dec. 2002.
- [17] Chong-Wah Ngo, Ting-Chuen Pong, and Hong-Jiang Zhang, "Motion Analysis and Segmentation Through Spatio-Temporal Slices Processing", *IEEE*

Transactions on Image Processing, Vol. 12, No. 3, pp. 341-355, Mar. 2003.

- [18] A. M. Ferman, A. M. Tekalp and R. Mehrotra, "Robust Color Histogram Descriptors for Video Segment Retrieval and Identification", IEEE Transactions on Image Processing, Vol. 11, No. 5, pp. 497-508, May 2002.
- [19] L. Zhao, W. Qi, S. Z. Li, S. Q. Yang and H. J. Zhang, "Key-frame Extraction and Shot Retrieval Using Nearest Feature Line (NFL)", Proceedings of the ACM Multimedia 2000 Workshops, Oct. 2000.
- [20] H. C. Lee and S. D. Kim, "Rate-driven key frame selection using temporal variation of visual content", Electronic Letters, Vol. 38, No. 5, pp. 217-218, Feb. 2002.
- [21] X. D. Zhang, T. Y. Liu, K. T. Lo and J. Feng, "Dynamic Selection and effective compression of key frames for video abstraction", Pattern Recognition Letters, Vol. 24, pp. 1523-1532, 2003.
- [22] T. Liu, H. J. Zhang, and F. Qi, "A Novel Video Key-Frame-Extraction Algorithm Based on Perceived Motion Energy Model", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 10, pp. 1006-1013, Oct. 2003.
- [23] S. Dagtas, W. Al-Knatib, A. Ghafoor and R. L. Kashyap, "Models for Motion-based Video Indexing and Retrieval", IEEE Transactions on Image Processing, Vol. 9, No. 1, pp. 88-101, Aug. 2001.
- [24] Symbolic Description and Visual Querying of Image Sequences Using Spatio-Temporal Logic", IEEE Transactions on Knowledge and Data Engineering, Vol. 7, No. 4, pp. 609-622, Aug. 1995.
- [25] 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.
- [26] D. Lelescu and D. Schonfeld, "Statistical Sequential Analysis for Real-Time Video Scene Change Detection on Compressed Multimedia Bitstream", IEEE Transactions on Multimedia, Vol. 5, No. 1, pp. 106-117, Mar. 2003.
- [27] S. C. Pei and Y. Z. Chou, "Efficient MPEG Compressed Video Analysis Using Macroblock Type Information", IEEE Transactions on Multimedia, Vol. 1, No. 4, pp. 321-333, Dec. 1999.
- [28] C. C. Lo and S. J. Wang, "A Histogram-Based, Moment-Preserving Clustering Algorithm for Video Segmentation", Pattern Recognition Letters, Vol. 24, pp. 2209-2218, 2003.
- [29] 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, Oct. 2001.
- [30] R. Zabih, J. Miller and K. Mai, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks", ACM Multimedia – Electronic Proceedings, pp. 189-200, 1995.
- [31] J. Meng, Y. Juan and S. F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence", Digital Video Compression: Algorithms and Technologies Proceeding, Vol. 2419, pp. 14-25, 1995.
- [32] C. W. Ngo, T. C. Pong and R. T. Chin, "Camera Break Detection by Partitioning of 2D Spatio-temporal Images in MPEG Domain", Proceedings, IEEE International Conference Multimedia Computing and Systems 1999, Vol. 1, pp. 750-755, 1999.
- [33] B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Video", IEEE

- Transactions on Circuits and Systems for Video Technology, Vol. 5, No. 6, pp. 533-544, Dec. 1995.
- [34] N. V. Patel and I. K. Sethi, "Compressed Video Processing for Cut Detection", IEE Proceeding Visual Image Signal Processing, Vol. 143, pp. 315-323, Oct. 1996.
- [35] A. M. Alattar, "Wipe Scene Change Detection for use with Video Compression Algorithms and MPEG-7", Proceedings, IEEE International Conference on Image Processing 1999, Vol. 3 , pp. 294 -298, Oct. 1999.
- [36] S. F. Chang and D.G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video", IEEE Journal on Selected Areas in Communications, Vol. 13, pp. 1-11, Jan. 1995.
- [37] B. L. Yeo, B. Liu, "A unified approach to temporal segmentation of motion JPEG and MPEG compressed video", Proceedings, IEEE International Conference on Multimedia Computing and Systems 1995, pp. 81-88, May 1995.
- [38] G. Paschos, M. Petrou, "Histogram ratio features for color texture classification", Pattern Recognition Letters, Vol. 24, Iss. 1-3, pp. 309-314, Jan. 2003.
- [39] Y. Rui, T. S. Huang and S. Mehrotra, "Exploring Video Structure Beyond the Shots", Proceedings, IEEE International Conference on Multimedia Computing and Systems 1998, pp. 237-240, Jul. 1998. Page(s): 237 -240
- [40] Y. Wu, E. Chang and B. Li, "Shot Transition Detection Using a Perceptual Distance Function", Proceedings, IEEE International Conference on Multimedia and Expo 2002, Vol. 1, pp. 293-296, Aug. 2002.
- [41] B. T. Truong, S. venkatesh and C. Dorai, "Scene Extraction in Motion Pictures", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 1,

pp. 5 -15, Jan. 2003.

- [42] E. K. Kang, S. J. Kim and J. S. Choi, "Video retrieval based on key frame extraction in compressed domain", Proceedings, IEEE International Conference on Image Processing 1999, Vol. 3, pp. 260 –264, Oct. 1999.
- [43] M. Yazdi and A. Zaccarin, "Scene Break Detection and Classification Using a Block-wise Difference Method", Proceedings, IEEE International Conference on Image Processing 2001, Vol. 3, 394-297, Oct. 2001.
- [44] H. B. Lu, Y. J. Zhang and Y. R. Yao, "Robust Gradual Scene Change Detection", Proceedings, IEEE International Conference on Image Processing 1999, Vol. 3, pp. 304 – 308, Oct. 1999.
- [45] E. Stringa and C. S. Regazzoni, "Real-Time Video-Shot Detection for Scene Surveillance Applications", IEEE Transactions on Image Processing, Vol. 9, No. 1, pp. 69-78, Jan. 2000.
- [46] N. Dimitrova and M. M. Abdel-Mottaied, "Content-based video retrieval by example video clip", The International Society for Optical Engineering, Vol. 3022, 1998
- [47] N. Dimitrova, J. Martino, L. Agnihotri and H. Elenbaas, "Color superhistograms for video representation", Proceedings, 1999 IEEE International Conference on Image Processing, Vol. 3, pp. 24-28, Oct. 1999.
- [48] Z. Li, X. Zhong and M. S. Drew, "Spatial-temporal joint probability images for video segmentation", Pattern Recognition, Vol. 35, Iss. 9, pp. 1847-1867, Sept. 2002.
- [49] J. Lee and B. W. Dickinson, "Hierarchical video indexing and retrieval for subband-coded video", IEEE Transactions on Circuits and Systems for Video

Technology, Vol. 10, pp. 824 –829, Aug. 2000.

- [50] J. M. Gauch, S. Gauch, S. Bouix and X. Zhu, “Real time video scene detection and classification”, *Information Processing & Management*, Vol. 35, Iss. 3, pp. 381-400, May 1999.
- [51] S. H. Kim and R. H. Park, “An Efficient Algorithm For Video Sequence Matching Using The Modified Hausdorff Distance And The Directed Divergence”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, pp. 592 –596, Jul. 2002.
- [52] M. Vishwanath, P. Chou, “An Efficient Algorithm For Hierarchical Compression Of Video”, *Proceedings, IEEE International Conference Image Processing 1994*, Vol. 3, pp. 275 -279, Nov. 1994.
- [53] P. Muneesawang, L. Guan, “Automatic Relevance Feedback For Video Retrieval”, *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing 2003*, Vol. 3, pp. III_1 -III_4, Apr. 2003.
- [54] P. Muneesawang, L. Guan, “Video Retrieval Using An Adaptive Video Indexing Technique And Automatic Relevance Feedback”, *IEEE Workshop on Multimedia Signal Processing 2002*, pp. 220 - 223, 2002.
- [55] J. R. Smith, C. Y. Lin, M. Naphade, “Video Texture Indexing Using Spatio-Temporal Wavelets”, *Proceedings, IEEE International Conference on Image Processing 2002*, Vol. 2, pp. II-437 -II-440, Sept. 2002.
- [56] Y. P. Tan, S. R.Kulkarni, P. J. Ramadge, “A framework for measuring video similarity and its application to video query by example”, *Proceedings, IEEE International Conference on Image Processing 1999*, Vol. 2, pp. 106-110 Oct. 1999.

- [57] T. Lin and H. J. Zhang, "Automatic Video Scene Extraction By Shot Grouping", Proceedings, IEEE International Conference on Pattern Recognition 2000, Vol. 4, pp. 3-7, Sept. 2000.
- [58] Y. Gong; L. T. Sin; C. H. Chuan; H. J. Zhang; M. Sakauchi; "Automatic parsing of TV soccer programs", Proceedings, IEEE International Conference on Multimedia Computing and Systems 1995, pp. 167-174, May 1995.
- [59] M. Yeung, B. L. Yeo and B. Liu "Extracting story units from long programs for video browsing and navigation", Proceedings, IEEE International Conference on Multimedia Computing and Systems 1996, pp. 296-305, Jun. 1996.
- [60] G. Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval", Pattern Recognition, Vol. 35, Iss. 8, pp. 1675-1686, Aug. 2002.
- [61] W. A. C. Fernando, C. N. Canagarajah, D. R. Bull, "A unified approach to scene change detection in uncompressed and compressed video", IEEE Transactions on Consumer Electronics, Vol. 46, No. 3, pp. 769-779, Aug. 2000.