1-1-2011

Video content analysis using the video time density function and statistical models

Junfeng Jiang
Ryerson University

Recommended Citation

Jiang, Junfeng, "Video content analysis using the video time density function and statistical models" (2011). *Theses and dissertations.* Paper 1101.

Video Content Analysis Using the Video Time Density Function and Statistical Models

by

Junfeng Jiang

BEng, Harbin Institute of Technology, Harbin, Sep. 1999

MEng, Harbin Institute of Technology, Harbin, Sep. 2001

A dissertation
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2011

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Video Content Analysis Using The Video Time Density Function And Statistical Models, Junfeng Jiang

PhD, Electrical and Computer Engineering, Ryerson University, 2011

As an interesting, meaningful, and challenging topic, video content analysis is to find meaningful structure and patterns from visual data for the purpose of efficient indexing and mining of videos. In this thesis, a new theoretical framework on video content analysis using the video time density function (VTDF) and statistical models is proposed. The proposed framework tries to tackle the problems in video content analysis based on its semantic information from three perspectives: video summarization, video similarity measure, and video event detection. In particular, the main research problems are formulated mathematically first. Two kinds of video data modeling tools are then presented to explore the spatiotemporal characteristics of video data, the independent component analysis (ICA)-based feature extraction and the VTDF. Video summarization is categorized into two types: static and dynamic. Two new methods are proposed to generate the static video summary. One is hierarchical key frame tree to summarize video content hierarchically. Another is vector quantization-based method using Gaussian mixture (GM) and ICA mixture (ICAM) to explore the characteristics of video data in the spatial domain to generate a compact video summary. The VTDF is then applied to develop several approaches for content-based video analysis. In particular, VTDF-based temporal quantization and statistical models are proposed to summarize video content dynamically. VTDF-based video similarity measure model is to measure the similarity between two video sequences. VTDF-based video event detection method is to classify a video into pre-defined events. Video players with content-based fast-forward playback support are designed, developed, and implemented to demonstrate the feasibility of the proposed methods. Given the richness of literature in effective and efficient information coding and representation using probability density function (PDF), the VTDF is expected to serve as a foundation of video content representation and more video content analysis methods will be developed based on the VTDF framework.

Acknowledgments

It is my great pleasure to thank the many people who made this thesis possible.

I want to thank the Department of Electrical and Computer Engineering of Ryerson University for giving me the opportunity to complete this thesis and to conduct the necessary research work that I enjoy.

I can not overstate my gratitude to my PhD supervisor, Prof. Xiao-Ping Zhang. His enthusiasm, knowledge, encouragement, and patience inspired me. He also provided good advice, good company, and lots of new ideas. I would not have finished this thesis without his help.

The greatest thing about this department is the helpful environment. I am deeply indebted to the professors for their kind assistance.

I am deeply appreciative of Epson Canada and Eastman Kodak for providing me with financial support and research opportunities. In particular, I would like to thank Mr. Ian Clarke and Mr. Yury Yakubovich of Epson Canada, Dr. Alexander C. Loui of Eastman Kodak for their valuable advice and help.

I am indebted to my student colleagues of this department and all over the campus for providing a stimulating and fun environment in which I can learn and grow. I am especially grateful to my labmates (Hui Zha, Triloke Rajbhandary, Iris Choi, Joseph Santarcangelo, Arsalan Bahojb, Timothy Little, Luan Vo, and many others) for having shared many experiences, thoughts, encouragement and laughs with me throughout these years.

I feel a deep sense of gratitude for my parents, my brother, and my wife, whose love, encouragement, and support are the backbones of my life. One of the greatest experiences of my PhD was the birth of my son Leon, who provided endless happiness and inspiration for the completion of my thesis and many wonderful things.

Acronyms

BIC	Bayesian Information Criterion
EM	Expectation Maximum
GM	Gaussian Mixture Model
GMVQ	Gaussian Mixture Vector Quantization
HMM	Hidden Markov Model
HSV	Hue, Saturation, and Value
ICA	Independent Component Analysis
ICAM	ICA Mixture Model
ICAMVQ	ICA Mixture Vector Quantization
LL	Parameter-based Lagrangian Distortion Function
MDI	Minimum Discrimination Information
MSE	Mean Square Error
MTMSE	Motion-based Temporal Mean Square Error
MVTDF	Motion-based Video Time Density Function
PCA	Principal Component Analysis
PDF	Probability Density Function
R-D	Rate-Distortion
RGB	Red, Green, and Blue

RWBS	Repeated Weighted Boosting Search
SVD	Singular Value Decomposition
SVS	Shot-based Video Summarization
TMSE	Temporal Mean Square Error
TQS	Non-uniform Sampling Using RWBS-based Temporal Quantization
TVQ	Non-uniform Sampling Using VTDF-based Temporal Quantization
US	Uniform Sampling
VQ	Vector Quantization
VTDF	Video Time Density Function
VTDFTQ	VTDF-based Temporal Quantization
VTDF-GM	VTDF-based Temporal Quantization and Gaussian Mixture Vector Quantization
VTDF-ICAM	VTDF-based Temporal Quantization and ICA Mixture Vector Quantization

List of Important Symbols

t	Video frame
$I(t)$	Video time density function
N	Total number of frames
M	Number of sample frames
Q	Class sets of partition
q	Quantization quanta
T	Temporal boundary
R	Rate
D	Distortion
U	Normalization factor
d	Dimension of features
x_t	Feature representation of video frame
K	Number of mixture components
μ	Mean
σ^2	Variance
$TMSE$	Temporal mean square error
BIC	Bayesian information criterion
A	ICA basis coefficients

W	Un-mixing matrix, sampling window
$d(.)$	Distance operator
$J(w)$	Cluster dissimilarity
X, Y	Video sequence
X'	Reconstructed video sequence
E	Class set of events
π	Probability of mixture component
C	Mixture component
θ	Parameter set
$p(.)$	Probability operator
ϵ	Small value to terminate iteration
s	Class set of video summary
S	Fast-forward speed factor
Σ	Covariance matrix
p	Number of parameters
b	Mean coefficients of mixtures
r, g	Normalized color feature vector
$H(t)$	Histogram of frame t
$g(.)$	Grey value operator
N_p	Number of pixels
$e(.)$	Error operator
δ	Coefficient of best or worst quanta
$q*$	New quanta
n	Time index

N_w	Non-uniform sampling interval
H	Non-uniform sampling factor
P	Slope
I_s	Sampling frame sequence
D_s	Sampling period/distance
$score(.)$	Similarity score operator
$cov(.)$	Covariance operator
$p(.), q(.)$	Alignment functions
L	Length of optimal path
$C(.)$	Event index operator
ρ	Correlation coefficient
$D(.)$	Event detection rate operator

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the rapid technology advances in digital TV, multimedia, and Internet, we have seen an amazing increase in the amount of digital image, audio, and video data in a very short time period. Among all the media types, video is the most challenging, as it combines all other media information into one single data stream. Due to the decreasing cost of storage devices, higher transmission rates, and improved compression techniques, digital video is becoming available at an ever-increasing rate. Thanks to the increasing availability of computing resources and the popularity of the Web 2.0 related technologies, we have witnessed a growing number of user-centric applications to allow ordinary people to record, edit, deliver, and publish their own home-made digital videos on social web or networks (e.g., YouTube). As a result, interaction with videos has become an important part of our life and many related applications have emerged. They include video on demand, digital video libraries, distance learning, and surveillance systems [55].

Currently, as a key element of multimedia computing, digital video has been widely employed in many industries as well as in various systems. However, because of the length of video and the unstructured format, efficient access to the video, especially video content-based access is not an easy task and is full of challenge. In other words, the increasing availability of digital video has not been accompanied by an increase in its accessibility [26]. The abundance of video data makes it increasingly difficult for people to manage and navigate their video collections. Therefore, efficient and effective automatic content-based video analysis techniques need to be developed.

1.1 Background Work on Video Content Analysis

Video content analysis is to find meaningful structure and patterns from visual data for the purpose of efficient indexing and mining of videos. The objective of video content analysis is to automatically analyze video semantic information for browsing, navigation, and retrieval. Therefore, the video content analysis techniques can be deployed in various domains. For example, we can find its application in gaming and entertainment. Microsoft launched Kinect for its game console, Xbox360 last year [4]. It uses video cameras for motion detection, skeletal tracking, and facial recognition. After capturing the input video and complex computing, it responds to simulate users' body actions effectively.

The early works on video processing mainly focus on low-level parsing (e.g., shot boundary detection), such as the methods in [50][14][15][42][30][16][9][73][61]. As the building blocks of video stream, a video shot is a continuous set of frames representing a continuous action in space and time. Detecting shot boundaries means segmenting a video temporally into its constituent shots and thus recovering the elementary units of a video [98]. Research on video shot boundary detection mainly focuses on two types of transitions: abrupt transition (cut) and gradual transition (wipe/fade-in-fade-out/dissolve). Compared with the abrupt transitions, the gradual transitions are more difficult to detect, this is due to the fact that it is not easy to define and capture the visual discontinuities. In general, automatic shot boundary detection techniques can be classified into five categories: pixel based, statistics based, transform based, feature based, and histogram based [92]. Complete reviews on shot boundary detection are presented in [26][41]. Currently, the field of video shot boundary detection has matured and many methods are capable of detecting not only simple cuts but also gradual transitions with considerable recall and precision [26]. The shot boundary detection usually serves as preprocessing steps for many video content analysis methods. However, the physical structure-based video analysis is not correlated to the semantic video content understanding. In other words, there is a huge semantic gap between low-level features and high-level content. The most efficient and straightforward way to narrow the semantic gap is to better understand the video content. It is the task of video content analysis.

This thesis refers all the content related tasks of video data to video content analysis, which includes, but is not limited to, content-based indexing, retrieval, summarization, classification, and semantic understanding. Compared with other multimedia data, video data is more random, less structured, high dimensional and is generally an integration of multiple modalities. Because of such characteristics, an efficient and effective video data modeling tool is very important. Considering probabilistic approaches are often used to model multimedia signals, this thesis tries to tackle the problems of video content analysis from statistical modeling point of view. The major concern of this thesis is not to build a complete content-based video analysis system. Instead, it is interested in developing methods, algorithms, and general frameworks to address three fundamental problems theoretically and then to validate the applications through preliminary implementations. In particular, three kinds of problems in video content analysis will be viewed and tackled in this thesis, video summarization (static and dynamic), video similarity measure, and video event detection. The research objectives for video content analysis in this thesis include:

- Finding effective and efficient feature representations of video data
- Modeling spatial and temporal characteristics of video data
- Summarizing video content in an effective and efficient way
- Finding effective and efficient similarity/dissimilarity measures for video sequences
- Detecting semantic video events for classification
- Achieving semantic video content understanding

1.1.1 Video Summarization

Key Frame Extraction

Key frame extraction is of great interest in many applications, such as video summary, video organization, video compression, and prints from video [66]. The definition of key frame depends on applications. From video summary point of view, key frame extraction can be considered as a procedure to select the most informative frames that capture the major elements in a video in terms of content. Several methods of key frame extraction have been proposed in literature. For example, a simple approach may just extract the first and the last frames of each shot as the key frames

[79]. More sophisticated key frame extraction techniques are based on visual content complexity indicators [45], shot activity indicators [40], shot motion indicators [89][34], spatiotemporal video modeling and segmentation [80], motion-activity-based extraction [29], and perceived motion energy model [63]. Currently, the status and trend in the field of key frame extraction can be addressed in two aspects. First is to summarize the whole video using extracted key frames. Second tends to towards extracting semantically meaningful key frames to match human visual system [66].

Video Summarization

Among existing video content analysis solutions, video summarization aims to organize video data into more compact forms and extract semantically meaningful information [28]. The current video summarization technologies can be roughly categorized into two main types: static and dynamic [83]. Static video summarization segments the whole video stream into several partitions (e.g., video shots). For each segment, one or more frames are extracted as the key frames. The summarization results lay out those key frames sequentially or hierarchically, such as the methods used in [20][92][101][29][63][35][49][12][11][43]. Although the static summarization can offer users a comprehensive view of video by generating a visual abstract of video content in a concise and informative way, it is susceptible to the smoothness problem, which means that the users may feel uncomfortable while browsing the results [92]. For example, given a video with long time duration, it is common to generate thousands of key frames using the above static methods. This is evidenced in [93]-there are 300 shots in a 15-minute video segment of the movie "Terminator 2: Judgment Day", and the movie lasts 139 minutes. The static layout of key frames is meaningless for the users' semantic video content understanding [92].

Dynamic video summarization is an alternative solution to generate some video skims from the original video stream. In literature, hidden Markov model (HMM) is used to generate the video skimming [13]. Attention models are applied to tackle this problem [68][67]. Different features (audio, visual and text) are combined together for video skimming generation [78]. A temporal semantic compression-based method is proposed for video browsing [8]. Highlight scene detection is applied to enable high-definition television systems [76]. Binary tree is also employed for online video summary [85]. In [77], VideoZoom is developed to browse video spatio-temporally. An interactive video navigation model is presented using optimal video decomposition in [31]. However, the high computational complexity of these methods makes them infeasible in practice [83]. For example, the HMM-based method used in [13] has to estimate the model parameters first before they can be applied to create video skims. In addition, most of existing summarization methods are video shot-based. However, the physical structure-based video analysis is not directly related to the semantic video content understanding [62]. According to the reviews on the advantages and disadvantages in existing methods made in [26][83], it becomes clear that new technique has to be developed to solve the video summarization problems in an effective and efficient way.

Performance Evaluation

Performance evaluation on the quality of video summary is a hot research topic. Unlike video shot boundary detection with ground truth dataset set by TRECVID [7] to evaluate the detection performance, video summary evaluation is mainly user-oriented and more subjective. There are two basic approaches identified in [56]: objective metrics (quantitative) and user studies (qualitative). For examples, in objective metrics, singular value decomposition (SVD) over feature frame matrix is used in [37]. Shot reconstruction degree is applied in [64]. Shot importance is evaluated in [84]. In user studies, key frame counting is used in [33] and user satisfaction is measured in [71]. From 2007, TRECVID BBC Rushes Summarization Workshop started tries to set up some ground truth summary based on BBC video data for researchers to compare and evaluate the summary quality [2]. Currently, no standard function can combine many factors such as camera motion, scene content, moving objects interaction, and image quality to evaluate the quality of video summary completely.

The current research trends in the performance evaluation can be addressed in two aspects: i) the technique enables automatic evaluation, and ii) it should integrate the users' subjective preferences and domain knowledge. Considering video summary is a form of video compression, it is reasonable to incorporate Rate (R)-Distortion (D) theory in quantitative performance evaluation, which is the reason to integrate R-D theory into the proposed content-based video processing framework in this thesis.

1.1.2 Video Similarity Measure

One crucial task in video content analysis is to measure the similarity/dissimilarity between two video sequences. Earlier solutions on video similarity measure are to extract the key frames of each video shot and compare those key frames to measure the shot similarity and video similarity, such as the methods used in [47][95]. However, these methods only consider one or several key frames and ignore the integrity of video. A fast random algorithms-based video signature is used to approximate the video similarity in [24]. The similarity measure is defined as the percentage of clusters of similar frames shared between two video sequences. A fast similarity search and clustering method is proposed to measure the similarity of video sequences over the Internet [23]. The authors in [52] try to solve the problem by calculating video hash values using block-based minimum variances. However, temporal information is ignored in the above methods.

Based on the nature that video similarity is a multiple-to-multiple matching process, some new solutions have been proposed in recent years. The probability distribution characteristics of video shots using nonlinear mapping are explored in [27]. A histogram pruning-based method to analyze audio and video signals is proposed in [51]. Ensemble similarity is proposed as a new metric for sequence similarity measure [100]. A video sequence matching similarity measure in shot-level is proposed in [97]. The main problem of those methods is that they are based on video shots in some forms. In other words, shot boundary detection has to be done first before the video similarity measure calculation in [97]. However, the video shot-based physical structure analysis is not correlated to the semantic video content understanding [62]. Therefore, new method should be developed to measure the similarity between video sequences effectively and efficiently.

1.1.3 Video Event Detection

A variety of video content analysis techniques have been developed to analyze video content in the past few years. Among these techniques, video event detection is considered as a crucial task to realize the semantic video understanding by using the recognition and detection of events from video.

In recent years, several video event detection solutions have been proposed in literature. HMM is applied to detect "play" and "break" events of soccer video [90]. A cross-correlation-based statistical method is applied to do video classification [75]. A real-time approach using high-level descriptors is proposed in [36]. Unsupervised classification based on color ratio and motion in soccer domain is discussed that uses Gaussian mixture (GM) as the observation model [90]. MPEG-7 audio features and entropic prior HMM are applied to detect predefined semantic events such as "cheering" and "applause" [91]. Baseball highlights are modeled as HMM with various kinds of features [38]. Non-Gaussian property of visual features is explored under independent component analysis (ICA) mixture observation model and HMM for golf video event detection [99]. ICA mixture (ICAM) hidden conditional random field model is proposed to detect sports events [86]. A hierarchical HMM structure is proposed to classify the video events [90].

However, the problems of existing methods can be addressed in three aspects. First, the model-based methods have to do training and model parameter estimations first before they can be applied in practice. Second, the detection results obtained in [90] and [91] depend on domain knowledge-based features. Therefore, they cannot be used for the generic semantic event detection and classification directly. Third, the computational complexity is high by using either complex statistical models or high dimensional domain features. For example, although the tree-like hierarchical HMM proposed in [90][70] is a powerful model, it is too complex and inefficient in practice. Therefore, a new technique has to be developed to detect and recognize semantic video events effectively.

1.2 Video Content Analysis Using the VTDF and Statistical Models

In this thesis, two types of video data modeling tools are presented first, the ICA-based feature extraction and the video time density function (VTDF). ICA is applied to explore the spatial characteristics of video content and model each video frame in a compact 2D feature space. The goal of VTDF is to explore the temporal dynamics of video content and model the whole video stream in an compact way. After the ICA-based feature extraction, one video frame is represented as one point in the 2D ICA subspace. After the VTDF modeling, one video sequence is represented as an one dimensional time series signal. Both of them are effective to explore spatiotemporal characteristics of video data and reduce the gap between low-level features and high level semantic content.

This thesis then focuses on video summarization, a crucial task in video content analysis. In particular, two models are proposed to solve the problems in static summary. One is hierarchical key frame tree to enable dynamic key frames layout. Another is statistical model vector quantization to generate a compact video summary. Inspired by the concept of vector quantization, two models are proposed to solve the problems in dynamic summary. One is VTDF-based temporal quantization for rapid video navigation and another is VTDF statistical model for video thumbnail extraction. Given a time/rate budget, the proposed method can generate an optimal video summary with the minimum distortion. Video players are built to demonstrate the feasibility of proposed methods in practice.

This thesis then extends the proposed theoretical framework using the VTDF and statistical models to solve other two applications in video content analysis: video similarity measure and video event detection.

1.3 Main Contributions

The main contributions of this thesis can be summarized as follows,

1. A new content-based video data modeling tool to represent each video sequence as an one dimensional time series signal, VTDF
2. A new key frame-based video representation tool to enable dynamic key frames layout, hierarchical key frame tree
3. A new static video summarization method using statistical model vector quantization to generate a compact video summary
4. A new theoretical framework using the VTDF and statistical models for video content analysis, dynamic video summary, video segmentation, video similarity measure, and video event detection
5. A new application implementation on content-based video analysis, video player

1.4 Thesis Outline

Figure 1.1 shows the whole structure of thesis outline.

1. In Chapter 2, the main research problems in this thesis for video content analysis are formulated in a mathematical way first. Several new approaches using the VTDF and statistical models are then presented. The last part is related to

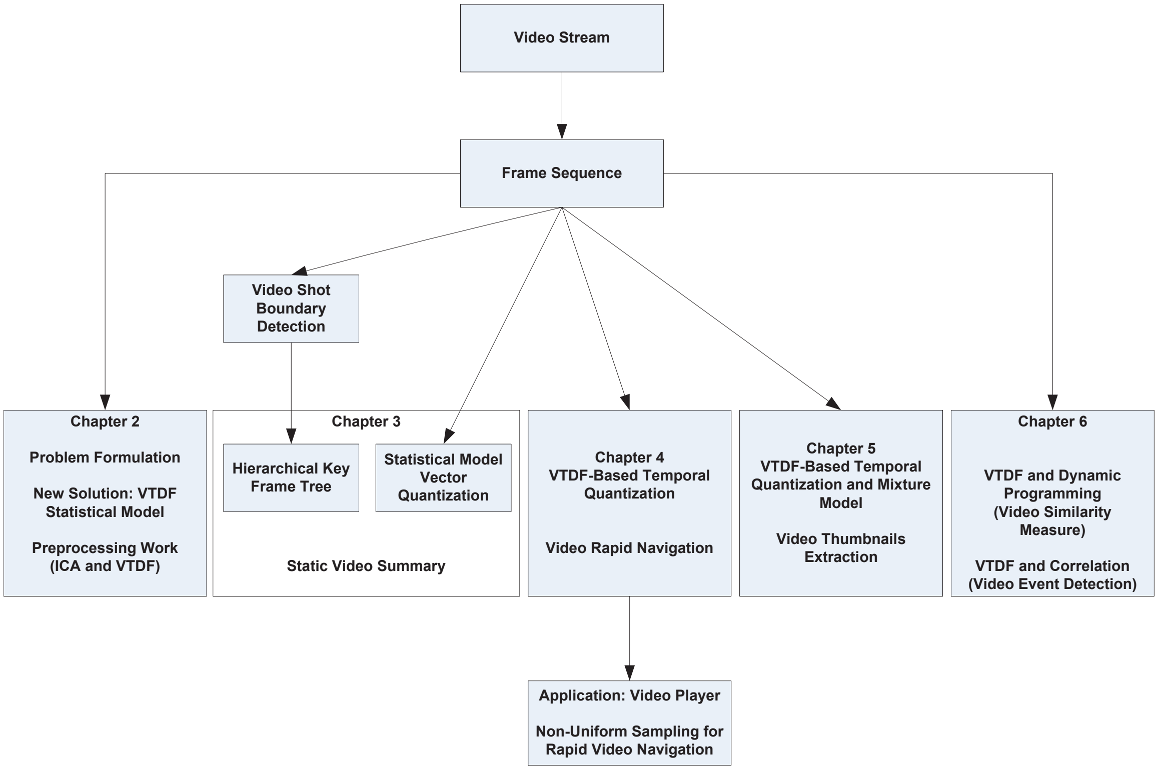

Video Stream
Frame Sequence
Video Shot Boundary Detection
Chapter 2
Problem Formulation
New Solution: VTDF Statistical Model
Preprocessing Work (ICA and VTDF)
Chapter 3
Hierarchical Key Frame Tree
Statistical Model Vector Quantization
Static Video Summary
Chapter 4
VTDF-Based Temporal Quantization
Video Rapid Navigation
Chapter 5
VTDF-Based Temporal Quantization and Mixture Model
Video Thumbnails Extraction
Chapter 6
VTDF and Dynamic Programming (Video Similarity Measure)
VTDF and Correlation (Video Event Detection)
Application: Video Player
Non-Uniform Sampling for Rapid Video Navigation

Figure 1.1: Thesis outline

the preprocessing work including ICA-based feature extraction and VTDF calculation. Four ways to calculate VTDF are proposed in this thesis, inter-frame histogram correlation, inter-frame histogram difference, inter-frame distance difference, and inter-frame mutual information. The VTDF is to explore the temporal dynamics of video content by measuring the dependency between two consecutive video frames. Motion-based video time density function (MVTDF) is then presented as an extension of VTDF to explore the motion activities of video data.

2. In Chapter 3, the solutions on static video summary are proposed in two aspects. One is hierarchical key frame tree to enable dynamic key frames layout.

Another is statistical model vector quantization to generate a compact video summary. Two mixture models are used: GM and ICAM. Accordingly, two vector quantization methods are developed: GM vector quantization (GMVQ) and ICAM vector quantization (ICAMVQ).

3. In Chapter 4, a new VTDF-based temporal quantization method is proposed to solve one type of dynamic video summary problem: rapid video navigation. The proposed method is integrated into a video player to demonstrate the feasibility in practice. An extension of video player using a set of parameters to do non-uniform sampling is introduced in the last section.

4. In Chapter 5, a new VTDF statistical model is proposed to solve another type of dynamic video summary problem: video thumbnail extraction. Given a video, its temporal characteristics are explored by VTDF-based temporal quantization and its spatial characteristics are explored by mixture model vector quantization, GMVQ and ICAMVQ.

5. In Chapter 6, a new video similarity measure model is proposed by using VTDF and dynamic programming first. A new video event detection method is then proposed by using VTDF and correlation.

6. In Chapter 7, a conclusion is made by summarizing the main research contributions in this thesis first. The future research directions are then identified.

1.5 Chapter Summary

The main research objective in this thesis is to analyze video content using a new theoretical framework based on the VTDF and statistical models. This Chapter first gives a background introduction on video content analysis from three perspectives: video summarization, video similarity measure, and event detection. From the point of view of video summary, this Chapter reviews the main research work including video shot boundary detection, key frame extraction, video summary, and performance evaluation. The main problems on exiting methods to summarize video content statically and dynamically are then identified. A theoretical framework using the VTDF and statistical models is proposed to solve the problems accordingly. The proposed framework can also be extended to solve the problems in video similarity measure and event detection. The last part in this Chapter summarizes the main contributions and layouts the thesis outline.

Chapter 2

Problem Formulation on Video Content Analysis

In this Chapter, the main research problems on video content analysis are formulated. A new theoretical framework on video content analysis using the VTDF and statistical models is then proposed. The last section discusses the preprocessing work on feature extraction.

2.1 Problem Formulations

In this thesis, the main research problems include video summarization (static and dynamic), video similarity measure, and video event detection. In particular, different from the existing methods that emphasize on video shot or scene change in some forms, this thesis formulates the video summarization as an optimal sampling problem by exploring the spatiotemporal characteristics of video content. Inspired by the vector quantization concept, a new research framework using the VTDF and statistical models is proposed to find an optimal video summarization solution with the minimum distortion. The proposed framework is then extended to measure the similarity between video sequences and detect the semantic video events.

2.1.1 Formulation on Static Video Summary

From quantization point of view, static video summary means to summarize video content in a manner that is independent of time. In other words, there is no temporal relationship for the frames within each partition. The summarization result is to layout the best sample frames sequentially or hierarchically.

Given a video stream with N video frames, the problem of static video summary is to sample M frames ($M < N$) to represent the original video with the smallest distortion.

Let X be the original video sequence,

$$X = \{x_1, x_2, ..., x_N\}, \tag{2.1}$$

where x_t is the feature vector of video frame t ($1 \leq t \leq N$).

Let Q be a partition solution of X,

$$Q = \{Q_1, Q_2, ..., Q_M\}, \tag{2.2}$$

where Q_i is the set of video frames in the i-th partition,

$$\bigcup_{i=1}^{M} Q_i = X, \tag{2.3}$$

$$Q_i \cap Q_j = \phi, i \neq j. \tag{2.4}$$

Note that the frames within each set do not have the temporal relationship in the proposed static video summary solution. In other words, they are not necessarily consecutive in time.

Let q be the set of quanta,

$$q = \{q_1, q_2, ..., q_M\}, \tag{2.5}$$

$$q_i = cent(Q_i), \tag{2.6}$$

where $cent(.)$ is the operator to find the quanta for each partition. Quanta q_i is used to represent all the video frames in Q_i.

An optimal static video summary can be determined as,

$$Q_{opt} = \arg\min_{Q} \left(e(Q)\right), \tag{2.7}$$

where $e(.)$ is the operator to calculate the distortion error.

2.1.2 Formulation on Dynamic Video Summary

Different from the above static video summary, dynamic video summary means to summarize video content with time constraints. Inspired by traditional vector quantization, this thesis formulates the dynamic video summary from temporal quantization point of view. Given a time budget, the main research problem is to find the best sample frames with the smallest quantization distortion. Note that the frames within each partition have a temporal relationship in the proposed solution of dynamic video summary. Therefore, compared with the above static video summary, dynamic video summary is more useful for rapid video navigation in practice.

Let X be the original video stream. In dynamic video summary, it is represented as,

$$X = \{1, 2, ..., N\}, \tag{2.8}$$

where t $(1 \leq t \leq N)$ is used to index the t-th video frame.

M is the number of sample frames, which can be determined by two ways: time budget or speed factor.

Given a time budget T_τ (e.g., to watch the whole video in 5 seconds), M can be determined as:

$$M = T_\tau \times f_{ps}, \tag{2.9}$$

where f_{ps} is the frame rate of video sequence X. The units of T_τ are seconds.

Given a fast-forward factor S $(S > 1)$ (e.g., to watch the whole video in $S = 2$ times fast-forward speed), M can be determined as,

$$M = N/S, \tag{2.10}$$

Let Q be a partition solution of X,

$$Q = \left\{Q_1, Q_2, ..., Q_M\right\}, \tag{2.11}$$

where Q_i is the set of video frames in the i-th partition,

$$\bigcup_{i=1}^{M} Q_i = X, \tag{2.12}$$

$$Q_i \cap Q_j = \phi, i \neq j. \tag{2.13}$$

Let T be the set of boundaries to segment Q in the time domain,

$$T = \left\{t_1, t_2, ..., t_{M-1}\right\}, \tag{2.14}$$

where t_i is the i-th boundary.

Therefore, Q_i can be determined as,

$$Q_i = \left\{t | t_{i-1} \leq t < t_i\right\}. \tag{2.15}$$

where $t_0 = 1$ and $t_M = N{+}1$.

The above formula indicates that the frames within a set Q_i have the temporal relationship. In other words, they are consecutive in time.

Let q be the set of quanta,

$$q = \{q_1, q_2, ..., q_M\}, \tag{2.16}$$

$$q_i = cent(Q_i), \tag{2.17}$$

where $cent(.)$ is the operator to find the partition quanta. Quanta q_i is used to represent all video frames in Q_i.

An optimal dynamic video summary can be determined as,

$$Q_{opt} = \arg\min_{Q} \left(e(Q) \right), \tag{2.18}$$

where $e(.)$ is the operator to calculate the distortion error.

2.1.3 Formulation on Video Similarity Measure

The main research problem on video similarity measure is to define a similarity function to measure whether two video sequences are semantically similar or not. In this thesis, video similarity is modeled as a generic multiple-to-multiple matching process. In particular, the video sequences are modeled as several segments first. A similarity function is then defined to measure the similarity between two segments. Video similarity is calculated by combining the similarities of segments.

Let X be one video sequence with M frames, and Y be the other video sequence with N frames. Both of them can be divided into several segments:

$$X = \{x_1, x_2, ..., x_U\}, \tag{2.19}$$

$$Y = \{y_1, y_2, ..., y_V\}, \tag{2.20}$$

where x_i $(1 \leq i \leq U)$ is the i-th segment in video X and y_j $(1 \leq j \leq V)$ is the j-th segment of video Y. And U and V are the number of segments of X and Y, respectively.

Note that all the segments in both X and Y have to be modeled as feature vectors first before the similarity calculation.

As a result, the similarity between X and Y can be determined based on the combination of similarities of two segments,

$$score(X, Y) = f(d(x_i, y_j)), \tag{2.21}$$

where $d(.)$ is the function to model the similarity between two segments. And $f(.)$ is the function to measure the sequence similarity based on the segment similarity.

2.1.4 Formulation on Video Event Detection

Given a video sequence, the main research problem in video event detection is to find a solution to classify it into one predefined video event. From video similarity point of view, video event detection can be considered as a procedure to find the event index that has the highest similarity score with the video sequence for classification.

Considering X is one video sequence with N frames and E is the class set including all predefined M events:

$$X = \{x_1, x_2, ..., x_N\}, \tag{2.22}$$

$$E = \{E_1, E_2, ..., E_M\}, \tag{2.23}$$

where x_t is the feature vector of the t-th frame $(1 \leq t \leq N)$. E_i is the feature vector to model the i-th event $(1 \leq i \leq M)$.

Let $C(X)$ be the event index of X. The optimized solution to classify X can be obtained by:

$$C(X)_{opt} = \arg\max_{i} \left(f(X, E_i) \right), \tag{2.24}$$

where $f(.)$ is a predefined decision function to measure the semantic similarity between X and E_i.

2.2 Video Content Analysis Using the VTDF and Statistical Models

2.2.1 Static Summary Using Hierarchical Key Frame Tree

Key frame-based video representation is a procedure to summarize video content by mapping the entire video stream to several representative video frames. However, the existing methods are either computationally expensive to extract the key frames at higher levels rather than the shot level or ineffective to lay out the key frames sequentially.

To overcome the shortcomings, a new hierarchical key frame tree-based video representation technique is presented to model the video content hierarchically. Concretely, by projecting video frames from an illumination invariant raw feature space into a low dimensional ICA subspace, each video frame is represented by a two-dimensional compact feature vector. A new kD-tree-based method is then employed to extract key frames at the shot level. A hierarchical agglomerative clustering-based method is then applied to process the key frames hierarchically. Experimental results show that the proposed method enables dynamic key frames layout.

2.2.2 Static Summary Using Statistical Model Vector Quantization

Inspired by the concept of traditional vector quantization, a new statistical model-based vector quantization method is presented to explore the video characteristics in the spatial domain and generate a compact video summary. In particular, the same feature extraction method using ICA is applied to model each video frame in a two-dimensional compact feature space. GMVQ and ICAMVQ are then developed to segment the whole video stream in the 2D ICA subspace and find the representative video frames. The optimal number of sample frames is determined by Bayesian information criterion (BIC). Experimental results show that this statistical model-based method can summarize video content in a compact way.

2.2.3 Dynamic Summary Using the VTDF and Statistical Models

Although the above two methods can solve the problems in video summarization in some aspects, both of them have disadvantages in general. Hierarchical key frame tree depends on video shot boundary. In other words, shot boundary detection has to be done first before applying hierarchical clustering. Statistical model-based summarization method ignores the temporal characteristics of video content. Therefore, new approaches using the VTDF and statistical models to summarize video content dynamically are presented in this thesis. The proposed methods try to solve the problems of dynamic video summary from two perspectives: rapid video navigation and video thumbnail extraction. The main difference between them is the number of sample frames. For rapid video navigation, the number of sample frames is predefined. While for video thumbnail extraction, this number is unknown until the video thumbnails are created.

Rapid Video Navigation

In current video players, the uniform fast-forward mode is still the main way for users to perform rapid video navigation. The traditional fast-forward is a sampling procedure to play and skip video frames uniformly. However, it may not be effective to capture the semantic information of video data. From mathematical point of view, the main research problem in rapid video navigation can be formulated as a generic sampling. For example, given a time constraint (e.g., to watch one video in five seconds), we need to find an optimal solution to sample the best frames for fast-forward playback. In this thesis, a new VTDF-based temporal quantization method for rapid video navigation is presented. Note that the proposed method is integrated in a video player for content-based fast-forward playback.

Different from the existing methods that rely on shot boundary detection in some form, the key novelty of this method is the using of vector quantization with an imposed temporal ordering. Specifically, the rapid video navigation problem is formulated as a generic sampling problem. The VTDF calculated by the inter-frame dependency is to model the video activities in the time domain. There are four ways proposed in this thesis to calculate the VTDF, inter-frame histogram correlation, inter-frame histogram difference, inter-frame distance difference, and inter-frame mutual information. Inspired by the vector quantization concept, a new VTDF-based temporal quantization method is then presented to find the optimal solution to sample frames. The proposed method can be justified from two perspectives: theory and application. First, a new VTDF-based distortion function, temporal mean square error (TMSE) is presented to evaluate the quantization performance. Note that R-D theory is integrated in the video summary framework for the performance evaluation. Second, from application point of view, it is practical to integrate this technique in a video player that allows the users to do variable-rate fast-forward playback that preserves semantic content.

The advantages of the proposed technique over existing methods can be summarized in three aspects. First, unlike the existing model-based methods, it does not need training data and parameter estimation. It is independent of the domain features and knowledge in general. Second, as an efficient and effective video data modeling tool, the VTDF is employed to explore the temporal dynamics of video data. Third, different from the emphasized aspects (shot boundary, scene change) in literature, this proposed method borrows the concept of traditional vector quantization to solve the video summarization problem from another perspective: given any specific time constraint (number of sampling frames), find the best samples of the video. As a result, in the built video player, the users can set any specific time constraint to do the content-based fast-forward playback before committing time to the original video.

Video Thumbnail Extraction

The main research problem in video thumbnail extraction is to find an optimal solution to summarize video in a compact way. To solve this problem, the VTDF and statistical models are combined together to explore the temporal and spatial characteristics of video data. First, a VTDF-based temporal quantization is employed to explore the temporal characteristics of video data and segment the whole video stream in the time domain. The optimal number of temporal segments is determined by a TMSE-based criterion. Second, for each temporal segment, GMVQ and ICAMVQ are applied to find the representative frames. The optimal number of sample frames is determined by a BIC-based criterion. As a result, a video thumbnail can be generated to summarize the original video in a compact way. The temporal characteristics of video data are explored by VTDF-based temporal quantization and the spatial characteristics of video are explored by mixture model vector quantization.

2.2.4 Video Similarity Using VTDF and Dynamic Programming

In this thesis, a new video semantic similarity measure model using VTDF and dynamic programming is presented. In particular, the VTDF is calculated by using inter-frame dependency. The VTDF is used to explore the time density of video activities and model the video data in a compact and effective way. A temporal partition is then applied to divide each video stream into equal sized segments in the time domain. VTDF-based correlation coefficient is calculated to measure the similarity between two equal sized temporal segments. As a result, the video similarly measure is a combination of pair-to-pair similarity of temporal segments. Dynamic programming is developed to solve the above multiple-to-multiple mapping problem.

Compared with the existing methods in literature, the advantages of the proposed method can be addressed in three aspects. Firstly, the VTDF is employed to model each video sequence in a compact and effective way. Secondly, the proposed method considers the whole video stream rather than only several key frames. Thirdly, the distance measure combines both visual characteristics and temporal information together to measure the video similarity efficiently and effectively.

2.2.5 Video Event Detection Using VTDF and Correlation

This thesis presents a generic-oriented method to detect and recognize semantic video events. Specifically, the video event detection problem is formulated as a generic supervised learning problem first. A VTDF-based temporal quantization is then employed to find the optimal VTDF-based feature vectors for a given video sequence and makes all video sequences have the same computational dimensions. Correlation is calculated to measure the similarity between two video sequences. Given a video sequence for event detection and classification, the predefined event sequence that has the highest correlation coefficient value can be considered as the detection result in the proposed method.

An one-hour of golf video is used as the case study to test the detection performance. Compared with existing methods, the advantages of the proposed method can be addressed in three aspects. First, the VTDF is an effective video data modeling tool to explore the essential temporal characteristics of video content. Second, temporal quantization is employed to generate the same dimensional compact VTDF-based feature vectors for each video sequence. The VTDF-based features are generic-oriented and independent of domain knowledge. Third, it does not need model training, model learning, and parameter estimation in general.

2.3 Preprocessing Work on Feature Extraction

In this thesis, an ICA-based feature extraction method is applied to explore the characteristics of video data in the spatial domain and the VTDF is calculated to explore the temporal dynamics of video data.

2.3.1 Raw Feature Extraction Using Histogram

The color is chosen to model each video frame and build a raw feature space first. There are two kinds of histograms used to model each video frame in grey level: 256D RGB histogram and 128D HSV histogram.

256D RGB Histogram

Since illumination change is an important factor that affects the performance of video content analysis, a normalized chromaticity histogram [32][99] is chosen to reduce the lighting effects and quantize each video frame in grey level.

The feature vector $(r,\, g)$ is defined as,

$$r = R/(R + G + B), \tag{2.25}$$

$$g = G/(R + G + B), \tag{2.26}$$

where R, G, and B are the three components in RGB color space.

The color histogram is along r and g axes. For one video with N frames, a $16(r) \times 16(g)$, 256 bins color histogram is built to represent each video frame in grey level.

128D HSV Histogram

Since HSV color space is better than RGB color space to match human visual system [62], a 128D histogram in HSV color space is built to do video frame quantization in grey level.

The conversion from RGB to HSV can be determined as,

$$H = arccos\left\{\frac{(R-G)+(B-G)}{2\sqrt{(R-G)^2+(R-B)(G-B)}}\right\}, \tag{2.27}$$

$$S = max(R,G,B) - min(R,G,B), \tag{2.28}$$

$$V = max(R,G,B). \tag{2.29}$$

Under the HSV color space, a color histogram along H and S axes can be built. We disregard the V component because it is less robust to the lighting condition [92]. For one video with N frames, a $16(H) \times 8(S)$, 128 bins color histogram is built to represent each video frame in grey level.

Note that the 256D RGB histogram is used with no specify in this thesis.

2.3.2 ICA-based Feature Extraction

ICA learning method is applied in the feature domain to build a compact 2D feature space to represent each video frame in the ICA subspace.

Let X be a video sequence with N frames,

$$X = \{1, 2, ..., N\}, \tag{2.30}$$

where t $(1 \le t \le N)$ is used to index the t-th video frame.

A raw feature space in a matrix form can be built as,

$$A = [H(1)\ H(2)\ ...\ H(N)], \tag{2.31}$$

where each column vector $H(t)$ is the histogram of frame t.

ICA model is used to extract basis functions from natural images in [10]. Such basis functions can be used as features since two different classes of images tend to have different basis functions, which is the reason to use ICA in feature extraction in this thesis. ICA can be considered as an extension of classical principal component analysis (PCA). PCA is optimal in terms of reconstruction error in Euclidean space. The features produced by PCA are mutually uncorrelated. However, ICA not only de-correlates the data but also reduces higher-order statistical dependence of data [25].

Fast ICA learning method [46] is performed to generate the un-mixing matrix W and the independent sources. We reduce the dimension by only keeping the two most important projecting directions. The 2D output independent components are given by the product of matrices W and A.

Given frame t, its 2D ICA feature vector can be denoted by x_t,

$$x_t = [IC_1(t) \ \ IC_2(t)] \, . \tag{2.32}$$

Although the above ICA feature is calculated based on color histogram, it is effective to bridge the low-level features and the high-level semantic content. The main reason is that a video consists of a set of consecutive images and there is an inter-frame dependency relationship between two consecutive frames. So a video is different from an individual image. It is common that two images with different content have the same histogram (color distributions). However, In the above ICA-based feature extraction, the raw feature space matrix is built by using the histograms from all frames. The frames are similar in content become a compact cluster in the 2D ICA subspace. By exploring the specific trajectory characteristics of video data, the ICA feature is effective to model the semantic information of video in the spatial domain.

2.3.3 VTDF-Video Time Density Function

This thesis employs the VTDF to explore the temporal dynamics of video data. As a new video data modeling tool, VTDF is calculated by using inter-frame dependency to model video content in a compact way. After the VTDF calculation, one video stream will be represented as an one dimensional time series signal for following processing.

For one video, its time density carries a lot of information and makes it different from other videos. It is uncommon that two different videos have the same time density rhythms. Therefore, the time density rhythms can be considered as an ID to index the video and measure the similarity between two videos. From this point of view, the time density can be used to explore the semantic information of video data. Considering the purpose of the VTDF is to explore the video density in the time domain by assigning a weight to each video frame based on the inter-frame dependency, the VTDF is effective to explore the semantic information of video in the time domain.

Four ways to measure the inter-frame dependency are presented, histogram correlation, histogram difference, distance difference, and mutual information. An improved motion-based VTDF, MVTDF is then introduced to explore the motion activity dynamics of video data. Compared with VTDF, MVTDF can model video content in a more discriminative way. Based on the different application requirements, different VTDFs can be employed.

VTDF Calculation Using Histogram Correlation

Histogram correlation is used to model the inter-frame dependency. Two images with a high correlation coefficient mean that they are semantic correlated [62]. In other words, a small value of histogram correlation means a weak inter-frame dependency and a big difference in content.

$$I(t) = \frac{1}{U} \left| \frac{cov(H(t), H(t-1))}{\sigma(H(t))\sigma(H(t-1))} \right|, \tag{2.33}$$

$$I(1) = 0, \tag{2.34}$$

where U is the normalization factor. σ is the standard deviation and the expression $cov(.)$ means covariance operator.

VTDF Calculation Using Histogram Difference

The reason for using histogram difference to model the inter-frame dependency is that during sharp motion activities, the visual characteristics between two consecutive frames are expected to change dramatically [96].

Let $H(t, k)$ represent the k-th bin grey value of the histogram $H(t)$, where $0 \leq k \leq 255$. The VTDF calculated by using histogram difference can be determined as,

$$I(t) = \frac{1}{U} \left\{ \sum_{k=0}^{255} |H(t, k) - H(t-1, k)| \right\}^{-1}, \tag{2.35}$$

$$I(1) = 0, \tag{2.36}$$

where U is the normalization factor.

VTDF Calculation Using Distance Difference

Note that the ICA-based feature extraction is employed to model each video frame in the ICA subspace. A large Euclidian distance difference between two consecutive frames means a big difference in content and a weak dependency. In other words, the frames that are semantically similar tend to form a compact cluster, which means a small Euclidian distance in general. So the VTDF can be calculated using the inter-frame distance difference as well.

Let x_t and x_{t-1} represent frame t and its previous frame t-1 in the 2D ICA subspace. Its VTDF can be determined as,

$$I(t) = \frac{1}{U} \left\{ \|x_t, x_{t-1}\|_2 \right\}^{-1}. \tag{2.37}$$

$$I(1) = 0, \tag{2.38}$$

where U is the normalization factor.

VTDF Calculation Using Mutual Information

Let X be one discrete random variable. Its entropy to measure the information content can be defined as,

$$H(X) = -\sum_{x \in X} p(x) \log(p(x)), \tag{2.39}$$

where $p(x)$ is the PDF of X.

Let X and Y be two discrete random variables, their joint entropy is determined as,

$$H(X;Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log(p(x, y)), \tag{2.40}$$

where $p(x,y)$ is the joint PDF of X and Y.

Their mutual information can be defined as,

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p_1(x) p_2(y)} \right), \tag{2.41}$$

where $p_1(x)$ and $p_2(y)$ are the marginal PDF of X and Y, respectively.

The relationship between joint entropy and mutual information is,

$$I(X;Y) = H(X) + H(Y) - H(X;Y). \tag{2.42}$$

In probability theory and information theory, the mutual information of two random variables is a quantity that measures the mutual dependence of the two variables [69]. In other words, mutual information is to measure the amount of information that X conveys about Y. So it is useful to explore the dependency between two random variables [17]. Therefore, mutual information can be employed to calculate the inter-frame dependency, VTDF.

Before the inter-frame mutual information calculation, the joint probability between two consecutive frames needs to be modeled first.

Given one frame t with N_p pixels, the inter-frame dependency in pixel level can be explored by a class set C_{ij},

$$C_{ij} = \{k | g(t-1,k) = i, g(t,k) = j\}, \tag{2.43}$$

where $g(t,k)$ is the operator to get the grey value of frame t at pixel k $(1 \leq k \leq N_p)$, and $0 \leq i, j \leq 255$.

The C_{ij} is used to calculate the joint probability $D(i,j)$ as,

$$D(i,j) = \frac{|C_{ij}|}{N_p}. \tag{2.44}$$

Note that $D(i,j)$ is the joint probability to model that a pixel with grey value i at frame t-1 has a grey value j at frame t.

It is easy to see that the value scope of $D(i,j)$ is $[0,1]$. Therefore, based on the above joint probability matrix $D(256 \times 256)$, the mutual information based VTDF for frame t can be determined as,

$$I(t) = \frac{1}{U} \sum_{i=0}^{255} \sum_{j=0}^{255} D(i,j) \log \left(\frac{D(i,j)}{D(i,)D(,j)} \right), \tag{2.45}$$

$$I(1) = 0, \tag{2.46}$$

where $D(i,)$ and $D(,j)$ are the marginal probabilities, which can be determined as follows, and U is the normalization factor to make

$$\sum_{t=1}^{N} I(t) = 1, \tag{2.47}$$

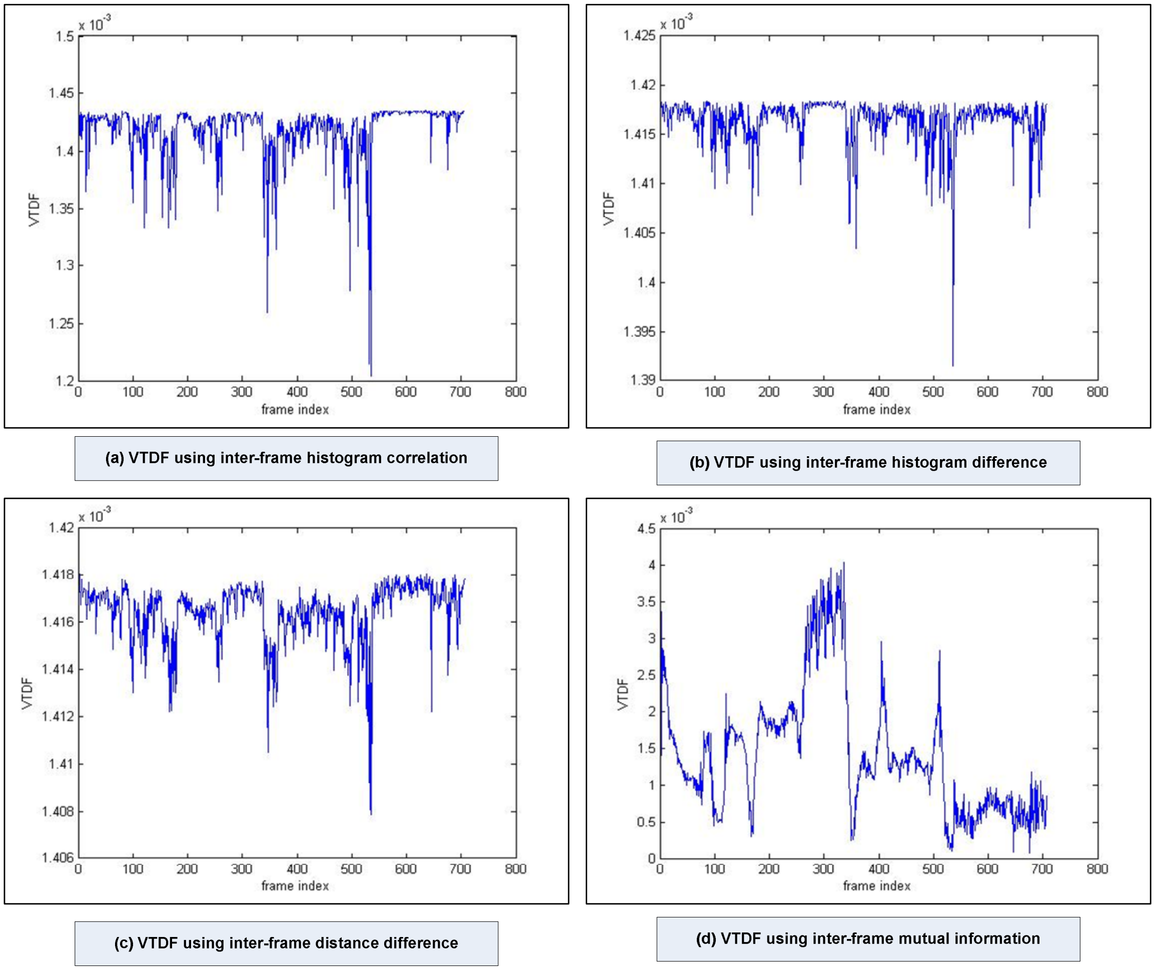
$$D(i,) = \sum_{j=0}^{255} D(i,j), \tag{2.48}$$

$$D(,j) = \sum_{i=0}^{255} D(i,j). \tag{2.49}$$

As a quantitative representation of inter-frame visual similarity measure, the above mutual information-based VTDF is an effective way to model the temporal density and explore the dependency between two successive video frames because a large difference in content between two frames shows a weak inter-frame dependency and leads to a small value of VTDF. Note that the above mutual information-based VTDF calculation is inspired by the work in literature [18]. Different from the work in [18] to calculate mutual information for video shot boundary detection, we calculate mutual information to measure the inter-frame dependency and give a weight to each video frame for its importance to solve the problem of dynamic video summary.

One video is randomly selected from our video collections to compare the four ways VTDF calculation. It has 705 frames and 352 × 240 frame size. First, we can compare them from visual point of view. In Figure 2.1, (a)-(d) are the four ways VTDF calculation, respectively. In each chart, the x-axis is “Frame Index”, t and the y-axis is “VTDF”, $I(t)$. In Figure 2.1, the VTDFs shown in (a)-(c) are similar from visual point of view because they are calculated based on histogram. In other words, the VTDF calculations from these three methods are in grey level in general. While Figure 2.1(d) is more discriminative to explore the temporal dynamics of video content. The reason is that the VTDF calculated using mutual information is in pixel level. Figure 2.1 effectively indicates that the VTDF is a discriminative way to explore the activity dynamics of video because the small value of VTDF means a weak inter-frame dependency and a large difference in video semantic content. In other words, the VTDF gives a weight to each video frame to measure its importance. One frame with a higher value of VTDF is more possible to be sampled in video summary. Therefore, the inter-frame VTDF between two adjacent frames makes it feasible to detect the segment boundary and sample the representative frames.

We can also compare the computational cost in the four ways VTDF calculation. Table 2.1 lists the processing time (units: seconds) for all four methods. From Table 2.1, it can conclude that the histogram difference and histogram correlation are the fastest methods. Distance difference is the third fastest method because it takes time to generate ICA features. Mutual information is the slowest method because it calculates the inter-frame dependency in pixel level. So there is a trade-off between precision and efficiency in these methods. Note that in this thesis, the inter-frame mutual information is used for VTDF calculation because it can explore the temporal dynamics in a more discriminative way.



(a) VTDF using inter-frame histogram correlation

(b) VTDF using inter-frame histogram difference

(c) VTDF using inter-frame distance difference

(d) VTDF using inter-frame mutual information

Figure 2.1: Example of four ways' VTDF calculation

2.3.4 Motion-based VTDF

Given a frame t, its previous frame t_p, and its next frame t_a, a new frame t' can be generated to represent the frame t using the following pixel-based frame difference method in RGB color space.

$$t'(R,i) = |t_p(R,i) - t(R,i)| + |t_a(R,i) - t(R,i)|\,, \tag{2.50}$$

$$t'(G,i) = |t_p(G,i) - t(G,i)| + |t_a(G,i) - t(G,i)|\,, \tag{2.51}$$

Table 2.1: Computational cost for four VTDF calculations

Methods	**Processing Time** (Units: Seconds)
Histogram Correlation	43.64
Histogram Difference	38.25
Distance Difference	97.47
Mutual Information	196.81

$$t'(B,i) = |t_p(B,i) - t(B,i)| + |t_a(B,i) - t(B,i)|\,, \tag{2.52}$$

where $t(R,i)$, $t(G,i)$, and $t(B,i)$ is the value of R, G, and B component at the pixel i for the frame t, respectively.

The above method is a procedure to create a new image by remaining the changed pixels and eliminating unchanged ones between one frame and its two adjacent (previous and after) frames. It is effective to explore the motion activity because during sharp motion activities, the visual characteristics among three consecutive video frames are expected to change dramatically. After this procedure, a new video sequence is generated to represent the original video sequence for following processing. Note that the new video sequence has the same number of video frames as the original sequence in nature.

Since motion that consists of a set of time consecutive frames is considered as an important semantic-related feature in content-based analysis [94], the inter-frame VTDF can be employed to explore the dependency between two consecutive video frames and model the motion characteristics of video data.

Figure 2.2 shows an example to generate a new image using the above pixel-level frame difference-based method. Figure 2.2 shows three time continuous frames and the new image generated to represent the middle frame. In Figure 2.2, the new image clearly describes the motion regions including both local motion (moving players) and global motion (camera).

We can also demonstrate the effectiveness of MVTDF for a bowling video and compare it with the VTDF. The inter-frame VTDF on its new video sequence is calculated and shown the MVTDF in Figure 2.3(a). For comparisons, the inter-frame VTDF, calculated on the original video sequence is shown in Figure 2.3(b). For one non-zero small value of MVTDF in the bowling video (circled in Figure 2.3(a)), we lay out the video frames and their previous ones for two video sequences, which are shown on the bottom of Figure 2.3. Figure 2.3 indicates that MVTDF is more discriminative to explore the motion activity dynamics of video because the small MVTDF value means a weak inter-frame dependency and a large difference in motion content. Therefore, MVTDF makes the boundary of temporal segments detection feasible and enables the representative frames sampling using a temporal quantization.

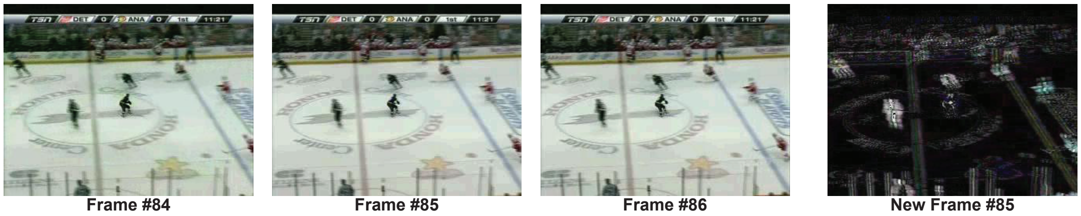

Frame #84 Frame #85 Frame #86 New Frame #85

Figure 2.2: Motion-based new image example

2.4 Chapter Summary

In this Chapter, the main research problems in this thesis are formulated in a mathematical way first. The new solutions on video content analysis are then introduced.

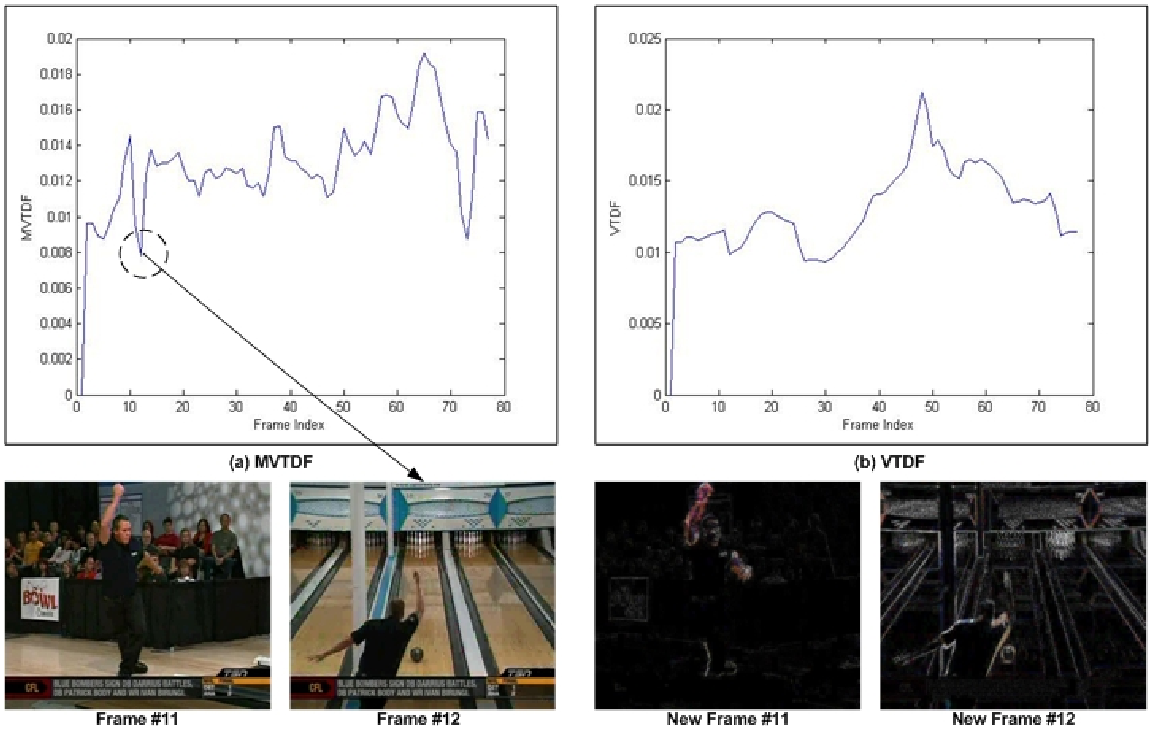


Figure 2.3: MVTDF and VTDF of bowling video

In particular, two new solutions on static video summary are introduced: hierarchical key frame tree and statistical model vector quantization. A theoretical framework using the VTDF and statistical models is then introduced to solve the problems in dynamic video summary, video similarity measure, and video event detection. The last part is related to the preprocessing work on the spatiotemporal video data modeling, ICA-based feature extraction and the VTDF calculation. The ICA-based feature extraction models each video frame using a compact 2D ICA feature vector. The VTDF models each video sequence as an one dimensional time series signal. MVTDF is an extension of the VTDF to explore the motion activities of video data.

Chapter 3

Video Summary Using Clustering and Quantization

In this Chapter, a new video representation tool, hierarchical key frame tree is presented to summarize video content hierarchically first. A new statistical model vector quantization method is then proposed to generate video summary in a compact way. There are two mixture models used in the proposed vector quantization method: GM and ICAM. Accordingly, two vector quantization methods are developed: GMVQ and ICAMVQ.

3.1 Hierarchical Key Frame Tree

Most existing methods in literature on static video summary depend on video shot in some form. For one shot, one or more frames are extracted as key frames. The problem is that it is very common to generate thousands of key frames for the videos with long time duration. It is not always beneficial to layout all key frames sequentially for the users' video content understanding. In this thesis, a new hierarchical key frame tree is presented to decrease the number of key frames and enable a hierarchical key frames layout.

3.1.1 Model Description

Let q_o be a set of original key frames sequence,

$$q_o = \{q_1, q_2, ..., q_M\}, \tag{3.1}$$

where q_i (i=1,...,M) is the i-th key frame and M is the size of original key frames sequence.

A partition solution can be applied to the original video stream to generate q_o,

$$Q = \{Q_1, Q_2, ..., Q_M\}, \tag{3.2}$$

$$q_i = cent(Q_i), \tag{3.3}$$

where $cent(.)$ is the operator to find the partition quanta for each set.

The following criterion can be applied to decrease the size of q_o at each iteration,

$$\arg\min_{i,j} \left(d(Q_i, Q_j)\right), \tag{3.4}$$

where $d(.)$ is the distance operator to calculate the similarity of two class sets, Q_i and Q_j, $1 \leq i < j \leq M$.

Two sets with the smallest distance will be merged as a new set. As a result, a new set of key frames with one key frame less will be generated as,

$$q_n = \{q_1, q_2, ..., q_{i-1}, cent(Q_i \cup Q_j), q_{i+1}, ..., q_{j-1}, q_{j+1}, ..., q_M\}. \tag{3.5}$$

The algorithm will be applied iteratively to generate the hierarchical key frames. The proposed method includes the following steps: (i) kD-tree-based key frame extraction; (ii) hierarchical agglomerative clustering-based key frame processing. Concretely, ICA-based feature extraction is employed to build a compact 2D feature space to represent each video frame. An existing method on video shot boundary is applied to segment the whole video stream into video shots [98]. A new kD-tree-based method is then employed to extract the key frames at the shot level. One frame is extracted from one shot to generate the original key frame sequence. In the last step, a hierarchical clustering-based method is developed to process the key frame sequence hierarchically. Based on the above steps, a hierarchical key frame tree can be generated to represent the video content.

Compared with existing methods, the advantages of this proposed method can be addressed in three aspects. First, the hierarchical key frame tree is built at the shot level. Second, the feature space is compact. Third, the kD-tree-based key frame extraction method has fast processing speed because it does not need to access all sample points for one given sample set.

3.1.2 KD-Tree-based Key Frame Extraction

After video shot boundary detection, each set Q_i is one video shot. To extract the key frame, a nearest neighbor-like similarity strategy by calculating the distance between each video frame and the target instance can be applied, through which the frame with the smallest distance is selected as the key frame.

The mean point of all frames within Q_i is used as the partition quanta, q_i,

$$q_i = cent(Q_i) = \frac{1}{N(Q_i)} \sum_{t \in Q_i} x_t, \tag{3.6}$$

where x_t is the feature vector of frame t. $N(.)$ is the operator to get the size of sample set Q_i.

Euclidean distance is calculated to measure the dissimilarity between frame x_t and the quanta q_i in the ICA subspace. The distortion error is determined as,

$$e(Q) = \frac{1}{M} \sum_{i=1}^{M} \frac{\sum_{x_t \in Q_i} \|q_i - x_t\|_2}{N(Q_i)}. \tag{3.7}$$

The frame with the smallest distance will be extracted as the key frame, denoted by k_i.

$$\arg\min_{t} \|q_i - x_t\|_2. \tag{3.8}$$

Therefore, the video X can be represented by its key frame sequence,

$$X = \{k_1, k_2, ..., k_M\}. \tag{3.9}$$

To enable the above key frame extraction procedure to be more efficient by searching only a part of the sample set, a new kD-tree-based method is employed, in which the sample set is represented as a tree. The built kD-tree divides the ICA subspace with a hyper plane and then splits the horizontal and vertical partition, recursively. Therefore, the search can be made by searching the built kD-tree accordingly. During the kD-tree building, all splits are parallel to either a horizontal axis (h) or a vertical axis (v).

Figure 3.1(a) shows an example of kD-tree ($k = 2$). In Figure 3.1(b), each member of the sample set is only one point in the coordinate, in which the star point is the target instance. And the tree root is the search result for this example.

In the built kD-tree, when the split is vertical, all left children have smaller values of IC_1 than the root point and all right children have larger values of IC_1 than the root point. Similarly, when the split is horizontal, all left children and right children

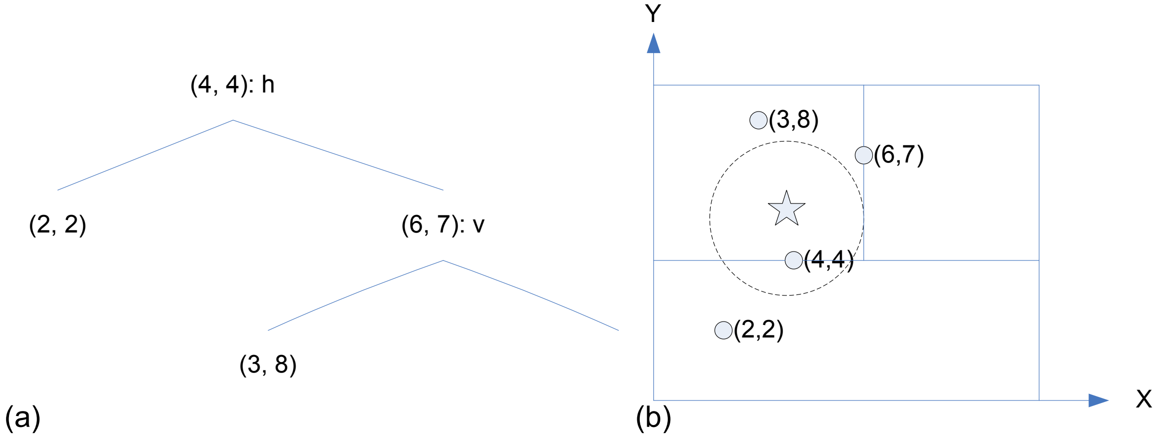


Figure 3.1: A kD-tree example in 2D coordinate

have smaller or larger values of IC_2 than the root point, respectively. As a result, only one branch needs to be searched. Note that IC_1 and IC_2 are the feature vectors in the 2D ICA subspace.

3.1.3 Hierarchical Key Frame Tree

In general, hierarchical clustering methods are based on the pair-wise dissimilarities among the observations in two groups. A basic paradigm of hierarchical clustering, agglomerative, can be developed to implement the hierarchical key frame processing. This strategy starts at the bottom and at each level recursively merges a selected pair of clusters into a single cluster [44]. The pair chosen for merging consists of the two groups with the smallest inter-group dissimilarity. There are M-1 levels in the hierarchy in total.

The clustering algorithm is summarized as follows:

- Initialization: $X = \{k_1, k_2, ..., k_M\}$

  Start by assigning each key frame as one item, the group it belongs as one cluster (denoted by σ), so there are M items and M clusters (one key frame per cluster).

Fisher's linear discriminator is calculated to measure the dissimilarity between two clusters, denoted by $J(w)$,

$$d(Q_i, Q_j) = J(w) = \frac{|q_i - q_j|^2}{\sigma_i^2 + \sigma_j^2}, \tag{3.10}$$

where σ_i^2 and σ_j^2 are the variance of two clusters, Q_i and Q_j, respectively.

- Iterative clustering:

  1. Find the closest (the most similar with the smallest distance) pair of clusters and merge the group they belong into a single new group and extract a new key frame based on the kD-tree-based method for this group. After that, we have one cluster and one item less.

  2. Compute distances (similarities) between the new cluster and each of existing old clusters.

  3. Repeat steps 1 and 2 until all items are clustered into a single cluster.

To track the hierarchical clustering iteration effectively, $L(i)$ and $M(i)$ $(0 \leq i \leq M-1)$ are to represent the remaining clusters and the distance matrix after the i-th clustering. In the initial condition, $L(0) = \{k_1, k_2, ..., k_M\}$ and the size of $M(0)$ is $M \times M$.

3.1.4 Experimental Results

Note that all experimental results in this thesis are obtained under the following programming environment: Windows XP Professional Service Pack3 (OS), 2G (RAM), Intel Xeon 3.06GHz (CPU), Java (JDK1.6, JMF and Eclipse3.4.2), and Matlab R2009a.

In the ICA-based feature extraction step, a frame sequence will be generated first for one given video. The performance of ICA-based feature extraction method can be demonstrated in Figure 3.2.

The test video is randomly selected from our video collection, which is encoded in MPEG2 with a frame rate of 29.9, frame size of 352 × 240, frame number of 720, and seven video shots. In Figure 3.2, the "red" and "blue" color points represent the key frames and other frames, respectively. Note that all clusters have been labeled. Figure 3.2 demonstrates that in the ICA subspace, the frames within one shot tend to form a compact cluster. Therefore, the ICA feature is effective to explore the frame level based global characteristics by projecting video frames from histogram-based

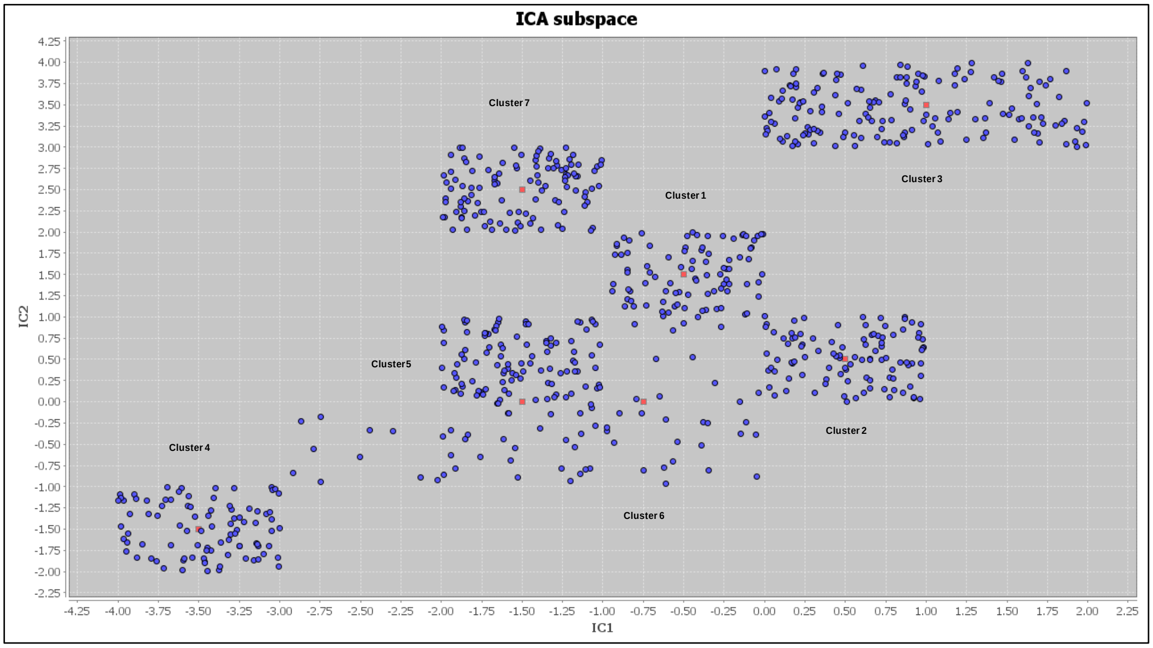


Figure 3.2: Video frame representation in 2D ICA subspace

raw feature space into the low dimensional ICA subspace. The processing time to generate the ICA features for the above video is 6.23 seconds.

In the kD-tree-based key frame extraction step, one key frame is extracted from each video shot. To show the effectiveness and efficiency of the proposed method, we can compare with the classic widely used k-means method and use three videos to demonstrate. They are all MPEG2 compressed with a frame rate of 30. Table 3.1 lists the experimental results on processing speed (units: milliseconds). Table 3.1 demonstrates that the proposed method enables real-time or near real-time video key frame extraction at the shot level. From the perspective of computational complexity, the built kD-tree is a binary search tree in nature. For one built kD-tree with n points, its computational complexity is only $O(\log n)$ [88].

Table 3.1: Time cost and results for key frame extraction

Test videos	**Total Frames**	**KeyFrame Number**	**KD-tree** (ms)	**K-means** (ms)
Video01	806	11	176	204
Video02	1717	20	283	372
Video03	7951	94	1526	2049

After key frame extraction, a hierarchical key frame tree can be generated by using the proposed hierarchical clustering method. For the video demonstrated in Figure 3.2, Table 3.2 shows its initial distance matrix.

Figure 3.3 shows the output of hierarchical key frame tree. In Figure 3.3, the bottom level is the original key frame sequence. The hierarchical clustering can be tracked by the labeled key frame indices accordingly.

The advantages of hierarchical key frame tree in video content structure modeling can be addressed in two aspects. First, it considers the relationship between two

Table 3.2: Initial distance matrix

	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7
σ_1	0	0.61	1.04	1.26	0.82	0.77	0.96
σ_2	0.61	0	1.03	1.27	1.09	0.69	1.15
σ_3	1.04	1.03	0	1.61	1.37	1.33	1.23
σ_4	1.26	1.27	1.61	0	0.81	1.09	1.39
σ_5	0.82	1.09	1.37	0.81	0	0.76	1.08
σ_6	0.77	0.69	1.33	1.09	0.76	0	1.32
σ_7	0.96	1.15	1.23	1.39	1.08	1.32	0

segmented video shots and extends the key frame-based video representation beyond the shot level by not modeling the higher levels of video. Second, compared with the single sequential key frame representation, it enables adaptive outputs of key frame sequence. The dynamic key frames layout makes it effective to represent the videos with long time duration using the hierarchical key frames.

3.2 Statistical Model Vector Quantization

The problem of the above hierarchical key frame tree is that shot boundary has to be detected first. In this section, a new vector quantization method using statistical models is presented to summarize video content regardless of video shots. In particular, the same ICA-based feature extraction is applied to model each video frame first. Mixture model vector quantization is then applied to find an optimal quantization codebook. Two types of mixture models are used: GM and ICAM. The quantization error is measured by mean square error (MSE). The optimal number of mixture components is determined by BIC.

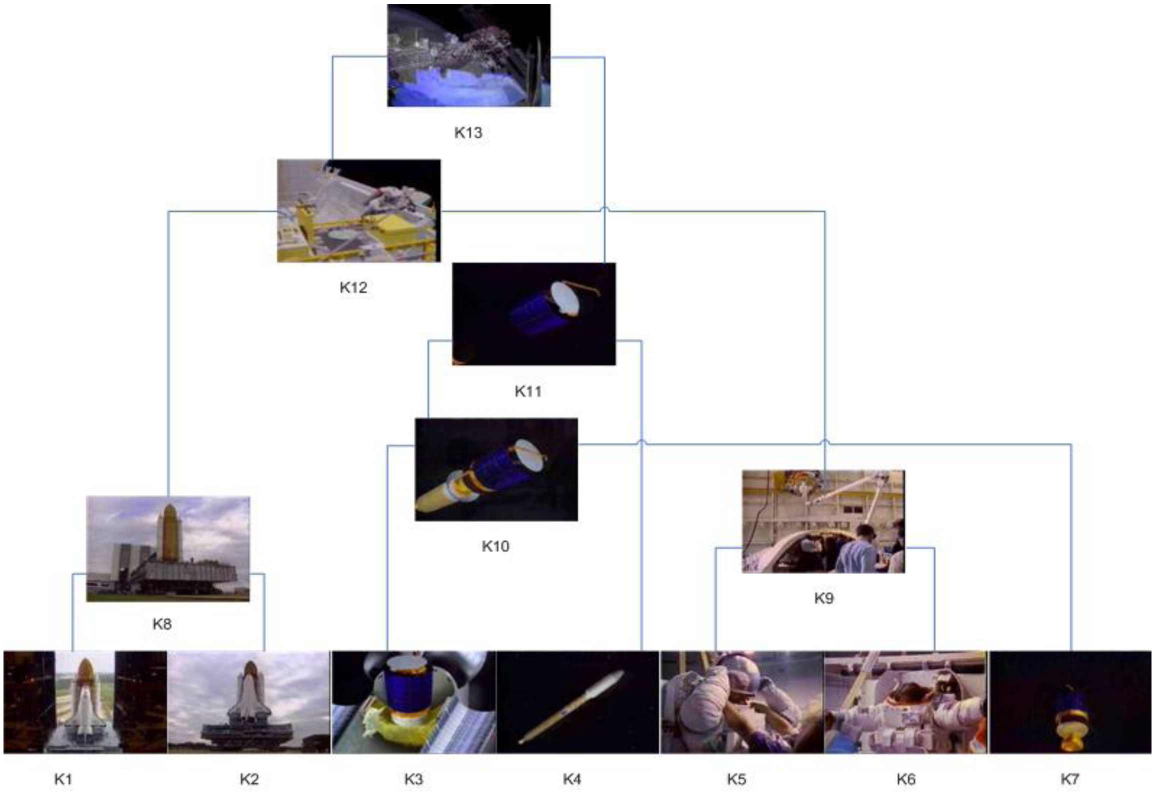


Figure 3.3: Hierarchical key frame tree

3.2.1 Model Description

Using mixture model under the 2D ICA feature space, each element x_t within Q has a PDF form as,

$$f(x_t|\theta) = \sum_{i=1}^{K} \pi_i p_i(x_t|C_i, \theta_i), \tag{3.11}$$

where π_i represents the probability of the i-th mixture component, $p_i(x_t|C_i, \theta_i)$ is the probability to produce x_t from C_i for a given parameter θ_i. θ is the class set of θ_i.

As a result, all x_t produced by C_i (maximum probability) can become a class set Q_i within Q.

$$Q_i = \left\{ x_t | i = \arg\min_i \left(p_i(x_t|C_i, \theta_i) \right) \right\}. \tag{3.12}$$

Note that different mixture models have different set of parameters to be estimated in nature.

No matter which mixture model is used, the following maximum log likelihood function is applied for parameter estimation.

$$\hat{\theta}_i = \arg\max_{\theta_i} \left(\log(f(x_t|\theta_i))\right). \tag{3.13}$$

As a result, expectation maximum (EM) algorithm can be used to estimate the parameters iteratively.

BIC is applied to determine the optimal number of mixture components, value of K as,

$$BIC = -2 \times \log(f(x_t|\theta)) + p \times \log N, \tag{3.14}$$

where p is the number of parameters for estimation.

The above BIC criterion is based on a trade-off between the performance and number of parameters used for describing the mixture distribution [94] [81]. The K caused the smallest BIC will be chosen as the optimal codebook size.

After above vector quantization, all video frame feature vectors can be iteratively classified into different class sets. For each Q_i, the objective of the proposed method is to find its optimized quanta representing all elements of it.

For each class set Q_i, its quanta can be calculated as,

$$q_i = cent(Q_i) = \frac{1}{N(Q_i)} \sum_{x_t \in Q_i} x_t. \tag{3.15}$$

MSE is used to measure the above spatial quantization error. The MSE and the termination condition of iteration can be determined as,

$$e(Q) = \frac{1}{K} \sum_{i=1}^{K} \frac{\sum_{x_t \in Q_i} \|q_i - x_t\|_2}{N(Q_i)}, \tag{3.16}$$

$$\frac{e(Q)^{(j-1)} - e(Q)^{(j)}}{e(Q)^{(j-1)}} < \epsilon_1, \tag{3.17}$$

where $e(Q)^{(j)}$ is the quantization error in the j-th iteration, $j > 1$. ϵ_1 is a predefined small number.

The representative video frame of class set Q_i, denoted by s_i, can be determined as,

$$s_i = \underset{x_t \in Q_i}{\arg\min} \left\| x_t - q_i \right\|_2 . \tag{3.18}$$

The total summarization result for Q can be represented by s,

$$s = \{s_1, s_2, ..., s_K\} . \tag{3.19}$$

3.2.2 Gaussian Mixture Model Vector Quantization

GMVQ is originally proposed to compress data in [39]. In literature, the traditional GMVQ methods mainly work on image processing related applications such as image retrieval [48][74][87].

The differences of the proposed GMVQ method from those in literature can be summarized in three aspects. First, our GMVQ is employed to a different application area, video summarization. Second, BIC is integrated into GMVQ to find the optimal codebook size. Third, a different quantization distortion function is presented. Unlike the minimum discrimination information (MDI) distortion function used in [39] and the parameter-based Lagrangian (LL) distortion function proposed in [48], MSE is used as the criterion to evaluate the quantization error. The reason to apply MSE to GMVQ is that it is effective to measure the quantization distortion in the spatial domain [65].

There are two main steps in the proposed GMVQ-based video summarization method. GM is used to estimate the PDF of feature vector x_t and segment the whole feature space first. A quantization procedure is then applied to find the best quanta with the minimum MSE. The video frames that are the nearest-neighbours to the quanta in the GMVQ codebook are extracted to summarize the whole video.

Given a Gaussian component C_i, its parameter set θ_i can be represented as,

$$\theta_i = (\pi_i, \mu_i, \Sigma_i), \tag{3.20}$$

where μ_i and Σ_i are the mean and covariance matrix, respectively.

Therefore, $p_i(x_t|\theta_i)$ can be calculated as,

$$p_i(x_t|\theta_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} e^{-\frac{1}{2}(x_t-\mu_i)^T \Sigma_i^{-1}(x_t-\mu_i)}, \tag{3.21}$$

where $d(d = 2)$ is the dimension of frame vector x_t.

The maximum log likelihood function is used for parameter estimation.

$$\hat{\theta}_i = \arg\max_{\theta_i} \left(\log(p_i(x_t|\theta))\right). \tag{3.22}$$

EM algorithm is used to estimate the parameters iteratively.

$$\pi_i = \frac{N(Q_i)}{N(Q)}, \tag{3.23}$$

$$\mu_i = \frac{1}{N(Q_i)} \sum_{x_t \in Q_i} x_t, \tag{3.24}$$

$$\Sigma_i = \frac{1}{N(Q_i)} \sum_{x_t \in Q_i} (x_t - \mu_i)(x_t - \mu_i)^T, \tag{3.25}$$

where $N(.)$ is the operator to get the size of class set.

GMVQ employs BIC as an optimal criterion for the codebook size, the value of K.

$$BIC = -2 \times \log f(x_t|\theta) + p \times \log N, \tag{3.26}$$

where $p(p = 3K)$ is the number of parameters for estimation.

Therefore, the proposed GMVQ is to find the optimal quantization quanta iteratively and the quantization quanta will be used to summarize the video content in a compact way.

3.2.3 ICA Mixture Model Vector Quantization

ICAM is originally presented to categorize the observations into mutually exclusive classes and model a broader range of PDF [53][54]. Similar to the above GMVQ, a new ICAM vector quantization method, ICAMVQ is presented to do video summarization. The reason to employ ICAM is that video data often shows non-Gaussian characteristics [99].

There are two steps in ICAMVQ. First, ICAM is used to estimate the PDF of feature vector x_t and segment the whole video in the spatial domain. Second, ICAMVQ is applied to find the best quanta with the minimum MSE.

$\forall x_t \in Q_i$, x_t can be modeled by a standard ICA model,

$$x_t = A_i S_i + b_i, \tag{3.27}$$

where A_i is the ICA basis coefficients and b_i is the mean coefficients for the mixtures. And S_i is the hidden source.

Therefore, given a ICAM component C_i, its parameter set θ_i can be represented as,

$$\theta_i = (\pi_i, A_i, b_i). \tag{3.28}$$

Accordingly, $p_i(x_t|\theta_i)$ can be calculated as,

$$\log(p_i(x_t|\theta_i)) = \log(p(S_i)) - \log(det|A_i|), \tag{3.29}$$

Similar to GMVQ, EM algorithm is applied to calculate the parameters iteratively.

$$\pi_i = \frac{N(Q_i)}{N(Q)}, \tag{3.30}$$

$$b_i = \frac{\sum_{t=1}^{N(Q)} \pi_i}{\sum_{t=1}^{N(Q)} \pi_i x_t}, \tag{3.31}$$

$$\triangle A_i \propto \frac{p(x_t|C_i,\theta_i)\pi_i}{\sum_{i=1}^{K}(p(x_t|C_i,\theta_i)\pi_i)} \frac{\partial}{\partial A_i} \log(p(x_t|C_i,\theta_i)), \tag{3.32}$$

where $N(Q_i)$ is the size of class set Q_i.

Similar to GMVQ, ICAMVQ employs the same BIC criterion as an optimal criterion for the codebook size, the value of K.

$$BIC = -2 \times \log f(x_t|\theta) + p \times \log N, \tag{3.33}$$

where $p(p = 3K)$ is the number of parameters for estimation.

Therefore, the above ICAMVQ is to find the optimal quantization quanta iteratively. The quantization quanta will be used to summarize the video content in a compact way.

3.2.4 Experimental Results

Experimental Results on GMVQ

To compare with the methods in static video summarization in literature, we refer to a video shot-based method (simple: SVS) for discussions. As a traditional method introduced in [92], SVS extracts one frame per shot as the key frame.

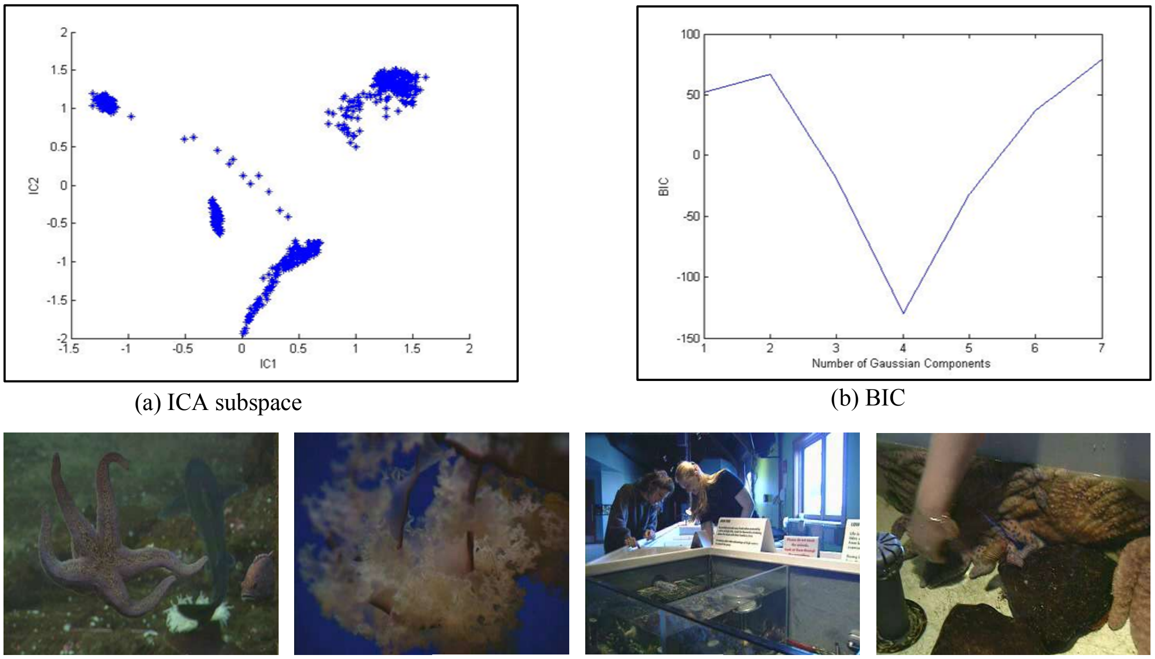
To show the effectiveness and the efficiency of GMVQ, we use three short videos and one long video to demonstrate. They are all encoded in MPEG2 with 29.9 frame rate and 352 × 240 frame size.

Video 1 is a classic small video to show the summarization performance of the proposed GMVQ method. Video 2 is a golf sports video to demonstrate that a comparable performance can be achieved by GMVQ for the same size of sample frames. Video 3 is a home-recorded video that has five shots including four fade-in-fade-out transitions with a still background. Video 3 is used to show that GMVQ can remove the redundancy of video summary obtained by SVS method. Video 4 is a movie video with long time duration to prove the capability of GMVQ to summarize the long duration video and generate a compact video summary.

Figure 3.4 shows GMVQ-based video summarization for video 1. An ICA-based 2D feature space shown in Figure 3.4(a) explores the spatial distribution characteristics of video content in the ICA subspace. The BIC chart is shown in Figure 3.4(b). The BIC criterion is a trade-off between the performance and the number of parameters used for GM distribution modeling. The K caused the smallest BIC value can be considered as the optimal codebook size. From Figure 3.4(b), it can acquire 4 as the optimized codebook size (number of GM components). The decision visually matches the spatial characteristics shown in Figure 3.4(a). Figure 3.4(c) shows the video summarization result. There are four sample video frames, one frame per GM component.

Table 3.3 lists the values of the optimized codebook size for all four test videos using BIC criterion. To compare with SVS, an ICA based shot-boundary detection method [98] is employed to obtain the number of total video shots. Other shot-boundary detection methods may be used as well.

In Table 3.3, the "Total Shots" column describes the number of video summary frames using SVS. The "Codebook Size" column shows the number of sample frames using GMVQ.



(a) ICA subspace

(b) BIC

(c) Video summarization

Figure 3.4: Video summary using GMVQ for video 1

Video 2 is a golf video to model a "full swing" scene in golf sports, which begins with a zoom-in to capture the player's preparation for his hit, and then followed by a quick camera motion to track the ball. The last scene locates the slowly moving ball. Figure 3.5 shows the sample frames of two methods. The top of Figure 3.5 is the summarization result of GMVQ and the bottom figure shows the summarization result of SVS. Figure 3.5 shows that a comparable performance can be achieved by GMVQ in golf "full swing" scene modeling.

Table 3.3 demonstrates that the optimized codebook size obtained by GMVQ is different from the number of shots in two test videos (video 3 and video 4). In other words, as a physical structure-based method, SVS may not be good enough to represent the semantic content in an optimal compact way.

Figure 3.6 shows the summarization results for video 3. Video 3 has five shots

Table 3.3: Summary of test videos for GMVQ

Video Index	**Total Frames**	**Total Shots**	**Codebook Size**
Video 1	803	4	4
Video 2	532	4	4
Video 3	616	5	2
Video 4	250200	3417	469

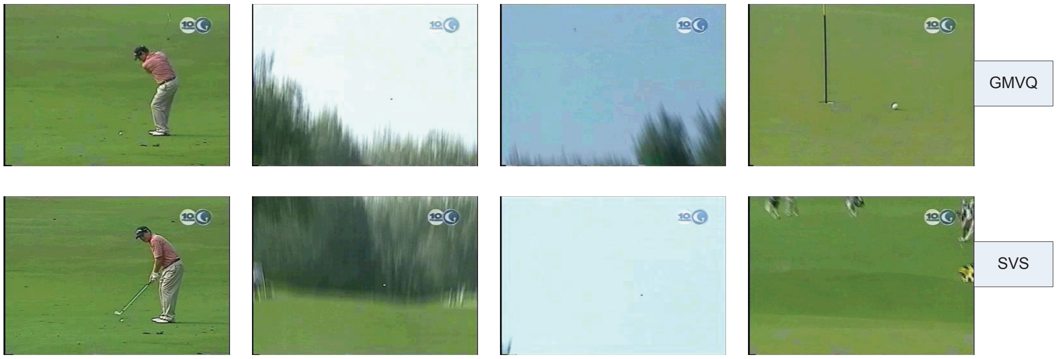


Figure 3.5: Video summary using GMVQ for video 2

with four semantic-similar fade-in and fade-out transitions. In each transition, the same person holding a piece of paper with a different word in his hands will fade-in and fade-out the still background. Although the number of sample frames of GMVQ is 2.5 times less than SVS, it can still grasp the important semantic information effectively.

Figure 3.6 shows that SVS may have the redundancy drawback. It can be demonstrated in test video 4. Video 4 is a movie video, “Terminator 2: Judgment Day”. The movie lasts over 120 minutes with 3417 shots. It is meaningless to lay out all 3417 key frames sequentially. In other words, it is useless to process such long time

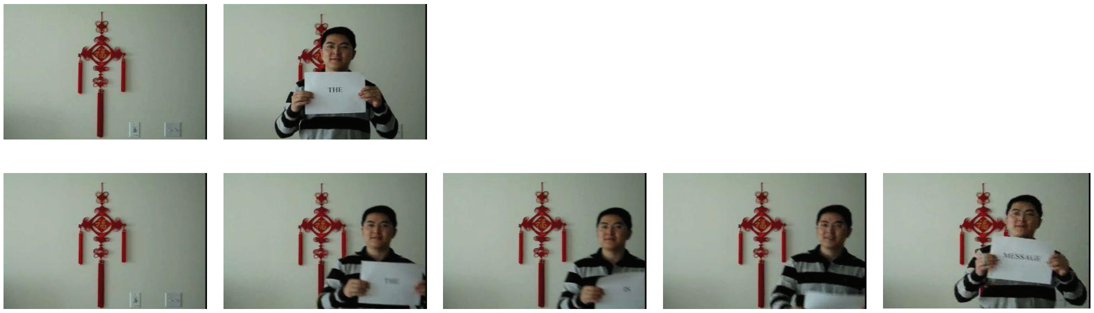


Figure 3.6: Video summary using GMVQ for video 3

duration video using SVS. However, GMVQ can still be applied in this case. Figure 3.7 shows the summarization results of video 4 using GMVQ.

In literature, a hierarchical clustering-based method is used to process the same movie video [93]. However, its computational complexity is $O(n^2)$ in order to get such compact summary. The proposed GMVQ method can obtain a compact video abstract without the hierarchical clustering procedure. And moreover, GMVQ uses a kD-tree-based structure to index the whole sample set. The worst performance for the nearest-neighbour search is $O(kn^{1-\frac{1}{k}})$, where $k(k = 2)$ is the dimension of the feature vector.

To demonstrate the advantage of GMVQ over other clustering methods, MSE is calculated to compare three methods, SVS, GMVQ, and classic K-Means. The results are shown in Table 3.4, which demonstrates that GMVQ is an optimal solution with the minimum MSE distortion.

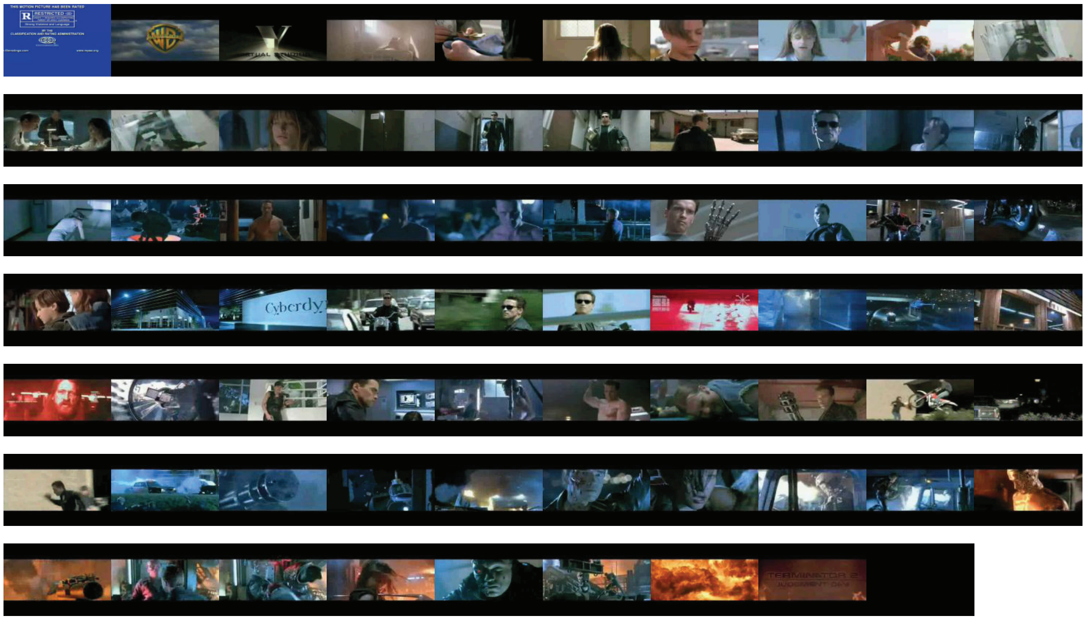


Figure 3.7: Video summary using GMVQ for video 4

Experimental Results on ICAMVQ

Three videos are tested to show the effectiveness and efficiency of ICAMVQ. They are all encoded in MPEG2 with 29.9 frame rate and 352 × 240 frame size. Table 3.5 is the summary of test videos. Note that the total shots are obtained by using existing video shot boundary detection method [98].

First, ICAMVQ is applied to summarize video 1. In Figure 3.8(a), one color represents one ICAM component in the compact 2D ICA feature space. Four ICAM components are used to explore the spatial characteristics of video 1. Figure 3.8(b) shows the BIC chart. From Figure 3.8(b), it can identify 4 as the optimal codebook size. This decision visually matches the spatial characteristics in Figure 3.8(a). A four-frame video summary is laid out in Figure 3.8(c).

Table 3.4: MSE comparisons of three methods for GMVQ

Video Index	SVS	GMVQ	K-Means
Video 1	18.32	10.47	13.41
Video 2	2.72	1.85	2.09
Video 3	7.39	6.84	9.73

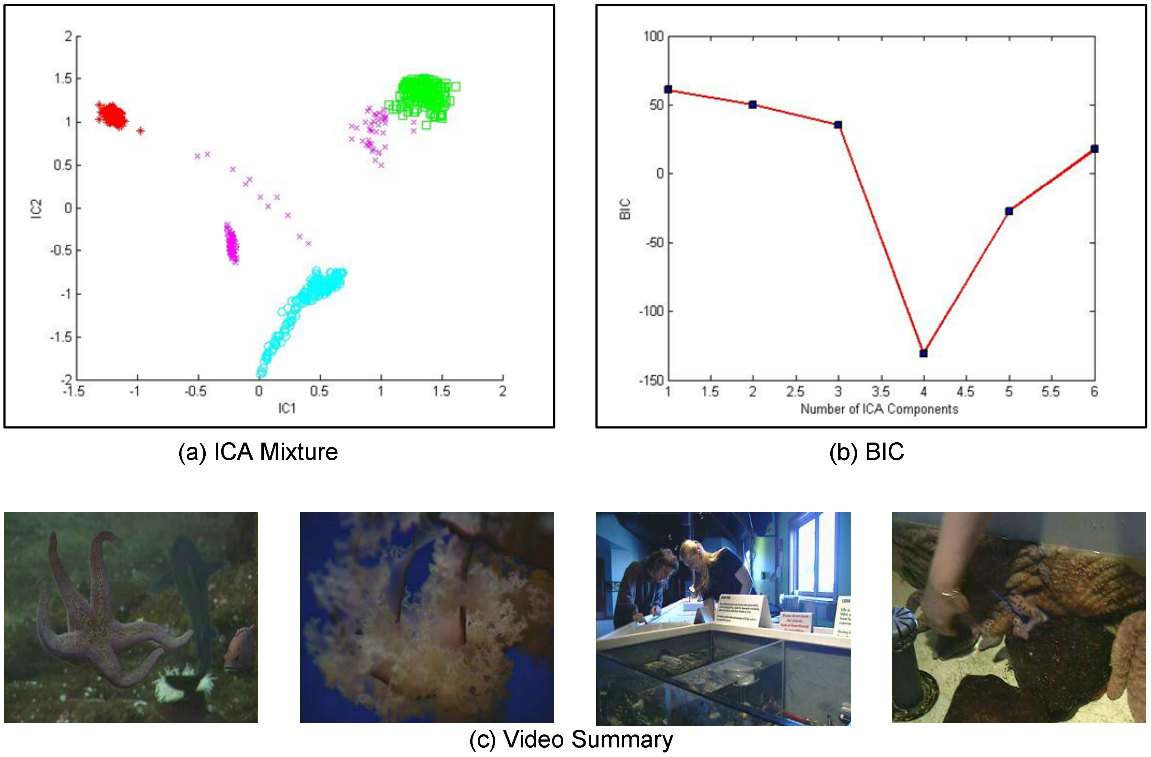
Table 3.5: Summary of test videos for ICAMVQ

Video Index	Total Frames	Total Shots	Codebook Size
Video 1	803	4	4
Video 2	532	5	3
Video 3	250200	3417	59

We can compare ICAMVQ with two methods, one is SVS and another is the above GMVQ. Here video 2 is used to demonstrate. Video 2 is a "full swing" event in golf video, which can be considered as a repetitive scene pattern in golf sports.

Based on the same BIC criterion, three ICAM components and four GM components are applied respectively. Therefore, a more compact video summary is achieved in ICAMVQ. Figure 3.9 shows the ICAM, GM, and the summarization results of three methods. The top one is the summary generated by using ICAMVQ. The middle one is the result obtained by GMVQ and the bottom one is the result of SVS method. Figure 3.9 shows that although the number of sampled frames of ICAMVQ is less than other two methods, it can still grasp the important semantic information (complete shot event) effectively.

Video 3 is the same movie video, "Terminator 2: Judgment Day". A compact summary that consists of only 59 frames can be generated by ICAMVQ.



(a) ICA Mixture

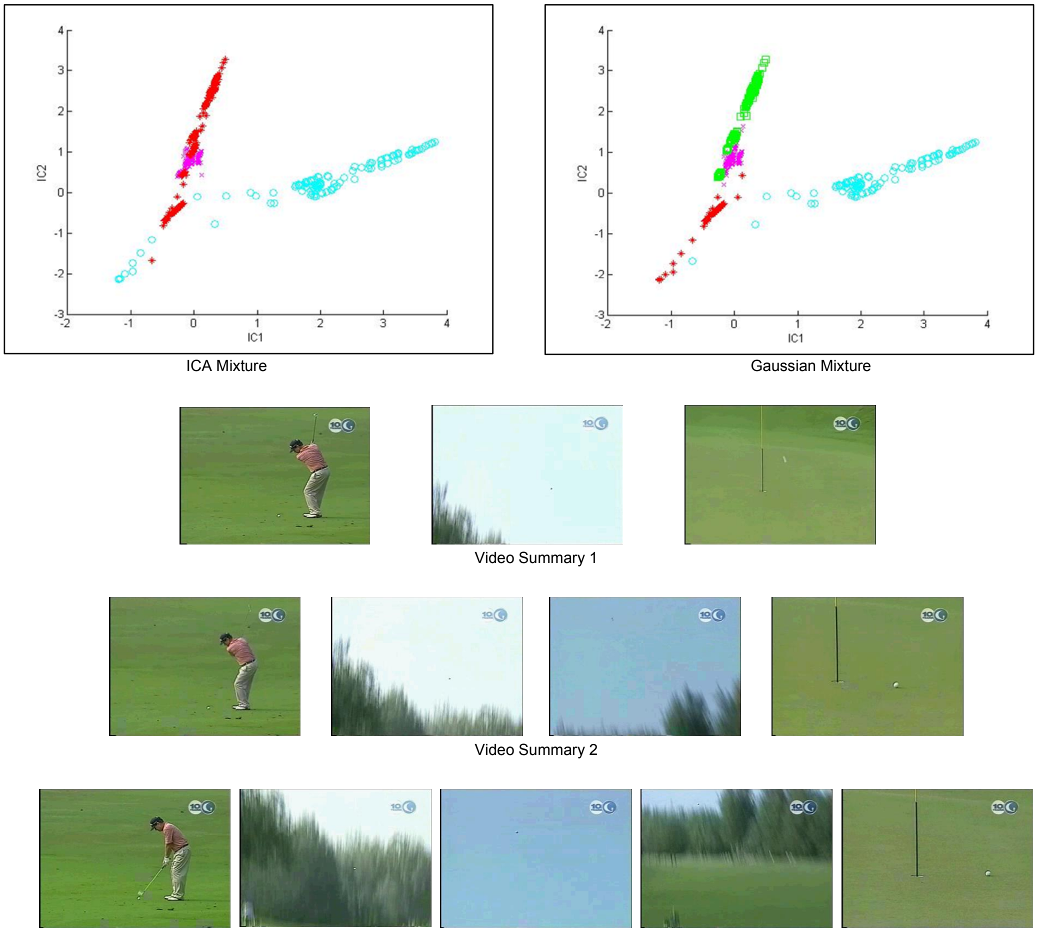
(b) BIC

(c) Video Summary

Figure 3.8: Video summary using ICAMVQ for video 1

3.3 Chapter Summary

Two new solutions on static video summary are proposed in this Chapter: hierarchical key frame tree and statistical model vector quantization. Hierarchical key frame tree enables dynamic key frames layout. Two methods on statistical vector quantization are proposed to generate a compact video summary: GMVQ and ICAMVQ. BIC criterion is integrated to determine the optimal number of sample frames. MSE is employed to calculate the quantization distortion.



ICA Mixture

Gaussian Mixture

Video Summary 1

Video Summary 2

Video Summary 3

Figure 3.9: Video summary using ICAMVQ for video 2

Chapter 4

VTDF-based Temporal Quantization

In this Chapter, considering dynamic video summary is a sampling procedure in nature, a new temporal quantization model is applied to find an optimal solution of dynamic video summary. Then two algorithms are presented to find the optimal sample frames. One is an existing method in literature, repeated weighted boosting search (RWBS). Another is the VTDF-based temporal quantization method. The new method borrows the concept of traditional vector quantization to sample video frames non-uniformly in the time domain. Note that R-D theory is integrated to the video summary framework for summary evaluation. After temporal quantization, all sample frames become a key frame sequence. A video player is designed, developed, and implemented to do fast-forward playback for all key frames. To allow the users to understand the video content better, a triangle-transition function with a set of parameters is employed to sample more frames around each key frame during fast-forward playback in the built video player. Experimental results show that the proposed method is efficient and effective to summarize the video content dynamically.

4.1 Model Description

In rapid video playback, given a video stream with N frames, an optimized summary solution is to sample M frames ($M < N$) to represent the original N frames with the minimum distortion.

Let X be the original sample set (frame indices),

$$X = \{1, 2, ..., N\}\,, \tag{4.1}$$

where t ($1 \leq t \leq N$) is used to index the t-th video frame.

The scheme of temporal quantization in the time domain can be represented by a set of quanta q, a set of partition (boundary) T, and $2M$-1 members,

$$q = \{q_1, q_2, ..., q_M\}\,, \tag{4.2}$$

$$T = \{t_1, t_2, ..., t_{M-1}\}\,, \tag{4.3}$$

$$q_1 < t_1 < q_2 < t_2 < ... < q_{M-1} < t_{M-1} < q_M. \tag{4.4}$$

In the time domain, the class set Q_i can be determined as,

$$Q_i = \{t | t_{i-1} \leq t < t_i\}\,, \tag{4.5}$$

where $t_0 = 1$ and $t_M = N{+}1$.

Note that R-D theory is integrated for performance evaluation. To do that, a reconstructed video sequence X' can be represented as,

$$X' = \{1', 2', ..., N'\}\,, \tag{4.6}$$

where t' $(1 \leq t' \leq N)$ is used to index the t-th video frame in the reconstructed video sequence.

After applying q_i to represent all the frames within each class set Q_i, X' can be represented as,

$$X' = \{q_1, q_1, ..., q_M\}, \tag{4.7}$$

where $\forall t' \in Q_i$, $t' = q_i$.

Quantization performance is evaluated by the following R-D theory,

$$R = M/N, \tag{4.8}$$

$$D = e(Q) = \frac{1}{N}\sum_{i=1}^{N} d(t, t') = \frac{1}{N}\left\{\sum_{i=1}^{M}\sum_{t \in Q_i} d(t, q_i)\right\}, \tag{4.9}$$

where $d(.)$ is an operator to calculate the distortion between each frame and its representative quanta.

The reason to apply R-D theory is that it provides a theoretical foundation for lossy video data compression [72]. From video summary point of view, rate determines the minimal amount of sample frames that can be approximately reconstructed the original video stream without exceeding a given distortion [57]. Therefore, given a predefined rate threshold, $T_r(R \leq T_r)$, the main research problem is to find an optimal solution of q while minimizing distortion, D.

Therefore, the optimized solution of quantization codebook set q is obtained by,

$$q_{opt} = \arg\min_{q}(D). \tag{4.10}$$

A new temporal quantization-based method is proposed to solve the above optimization problem. The new method includes two components: VTDF and temporal quantization. The VTDF is used to model the inter-frame density of activities in the time domain. The VTDF-based video data modeling is an effective and concise way to explore the time density characteristics of video content in the time domain. The VTDF between video frames makes it feasible to detect the segment boundary and sample the representative frames. The temporal quantization is used to explore the characteristics of video using the optimal quanta and partition in the time domain. Different from the traditional quantization method, the proposed temporal quantization makes all frames in one segment consecutive in time.

Similar to the PDF to sample the values based on the probability density changes, the proposed method employs the VTDF to explore the time density changes by assigning a weight to each frame to measure the importance and applies the quantization method in the time domain to find the best sample frames with the minimum distortion.

4.2 VTDF-Based Temporal Quantization

Based on the work in [65], the best partition t_i $(1 \leq i \leq M-1)$ can be determined by the quantization quanta as,

$$\begin{aligned} t_1 &= \tfrac{1}{2}(q_1 + q_2) \\ t_2 &= \tfrac{1}{2}(q_2 + q_3) \\ &\cdots \\ t_{M-1} &= \tfrac{1}{2}(q_{M-1} + q_M) \end{aligned} \tag{4.11}$$

Therefore, the whole sampling procedure is to update quanta and partition iteratively. There are two methods proposed to optimize the quanta and partition in the time domain. One is RWBS and another is temporal quantization using VTDF.

4.2.1 Temporal Quantization Using RWBS

The basic concept of RWBS is to find the "worst" quanta point and generate a new "better" quanta point to replace the worst one iteratively. The reason to employ the RWBS is that the RWBS is an effective and efficient solution to solve such global optimal problem [21]. Therefore, the RWBS can be employed to optimize the quantization quanta.

Similar to the traditional PDF used in conventional vector quantization, the VTDF, $I(t)$ can be used as the weight to measure the semantic importance of frame t and to define the cost function of quantization.

Given a class set Q_i and its quanta q_i, the following VTDF-based cost function is defined to model the quantization distortion-D,

$$e(q_i) = \sum_{t \in Q_i} d(t, q_i), \tag{4.12}$$

$$d(t, q_i) = (t - q_i)^2 I(t). \tag{4.13}$$

Based on the above cost function, the indices of the *best* quanta and the *worst* quanta can be found as,

$$best = \arg\min_i \left(d(t, q_i) \right), \tag{4.14}$$

$$worst = \arg\max_i \left(d(t, q_i) \right). \tag{4.15}$$

The quanta with *best* and *worst* as the indices, denoted by q_{best} and q_{worst}, can be used to generate two new quanta,

$$q_1^* = \sum_{i=1}^{M} \delta_i q_i, \tag{4.16}$$

$$q_2^* = q_{best} + (q_{best} - q_1^*), \tag{4.17}$$

$$\sum_{i=1}^{M} \delta_i = 1, \tag{4.18}$$

where the value of δ_i can be determined as,

$$\delta_i = \frac{N(Q_i)}{N}, \tag{4.19}$$

where $N(Q_i)$ is the size of class set Q_i.

Similar to the MSE used to model the traditional vector quantization distortion, we define a new TMSE to represent the global partition error, distortion-D as,

$$D = e(Q) = TMSE = \frac{1}{M} \sum_{i=1}^{M} e(q_i). \tag{4.20}$$

In each iteration, q_1^* and q_2^* are used to replace q_{worst}, respectively. After the replacement, the updated quanta set q needs to be sorted first before TMSE calculation can be applied. The one with a smaller error-TMSE will be chosen to replace q_{worst} accordingly in next iteration. The iteration will be repeated until the following termination condition is met,

$$|q_1^* - q_2^*| \leq \epsilon_2, \tag{4.21}$$

where ϵ_2 is a predefined small number.

4.2.2 Temporal Quantization Using VTDF

The same cost function using the VTDF, $I(t)$ as the weight is applied to measure the semantic importance of frame t,

$$e(q_i) = \sum_{t \in Q_i} (t - q_i)^2 I(t). \tag{4.22}$$

In continuous case, the above equation can be converted as,

$$e(q_i) = \int_{t_{i-1}}^{t_i} (t - q_i)^2 I(t) dt. \tag{4.23}$$

To find the optimal solution of above formula, let

$$\frac{\partial(e(q_i))}{\partial(q_i)} = 0. \tag{4.24}$$

The optimal quanta can be obtained by calculating the above equation.

$$q_i = \frac{\int_{t_{i-1}}^{t_i} tI(t)dt}{\int_{t_{i-1}}^{t_i} I(t)dt}. \tag{4.25}$$

The best quanta can be obtained in the discrete case similarly.

$$q_i = \sum_{t_{i-1}}^{t_i} tI(t) \bigg/ \sum_{t_{i-1}}^{t_i} I(t)\ . \tag{4.26}$$

The same TMSE is calculated to represent the global partition error in the above temporal quantization method. During each iteration, the quanta and partition will be updated iteratively until the following termination condition is met.

$$\frac{TMSE^{(i-1)} - TMSE^{(i)}}{TMSE^{(i-1)}} < \epsilon_3, \tag{4.27}$$

where $TMSE^{(i)}$ is the value of TMSE calculation in the i-th iteration ($i > 1$). ϵ_3 is a predefined small number.

The whole temporal quantization algorithm is summarized as follows,

1. Initialization: $q_1, q_2, ..., q_M$.
2. Iteration: calculate the partition T, update the class set Q, and update the quanta set q.
3. Termination: defined above.

The number of sample frames (codebook size) is determined by the users to specify how fast to navigate the video content, denoted by S ($S > 1$). Accordingly, the codebook size can be determined as,

$$M = round(N/S). \tag{4.28}$$

For both methods, if the optimal quanta is not an integer in each partition, a nearest-neighbor-based strategy is applied to find the representative frame t according to its VTDF value,

$$\arg\min_{t \in Q_i} \left(|I(t) - I(round(q_i))| \right). \tag{4.29}$$

As a result, the video frames that are the nearest neighbors to the quanta in the quantization codebook are sampled to navigate the video content.

4.3 Experimental Results

4.3.1 Experimental Results on RWBS-based Method

Three videos are tested to show the effectiveness and efficiency of the proposed method. They are all encoded in MPEG2 with 29.9 frame rate and 352 × 240 frame size.

Video 1 is a bowling video to model a "ball swing" event. Video 2 is a golf video to model a "full swing" scene. Video 3 has four fade-in-fade-out transitions with a still background. Table 5.1 is a summary of all three test videos, which specifies the values of M and N in each video.

To show the effectiveness of the proposed method using RWBS (TQS), we do

Table 4.1: Summary of test videos for RWBS

Video No.	**Total Frames-N**	**Sample Frames-M**
Video 1	77	3
Video 2	532	5
Video 3	616	5

two comparisons. First, we compare it with the uniform sampling (US). The reason is that a video player is built to integrate the proposed method. From the application point of view, it is straightforward to compare with the uniform fast-forward. Second, we compare it with the typical video summarization methods in literature. Different from our work, those methods are from another perspective: physical boundary of video. However, the physical boundary analysis is not necessary to help the semantic video content understanding. We refer to SVS for discussions. In SVS, one key frame is extracted to represent one video shot before hierarchical clustering.

The comparisons of three methods can be addressed in two aspects: theoretical representation error (TMSE) and subjective application evaluation (sampled frames lay out). Table 4.2 shows the TMSE comparisons in all three videos. Table 4.2 indicates that TQS has smaller distortions than US and SVS in all three videos.

Figure 4.1 shows the sampling results for video 1. Figure 4.1(a) shows the efficient iteration procedure of TQS using TMSE criterion for video 1, which demonstrates that the employment of RWBS in temporal quantization can achieve the converge goal in limited iteration times. Figure 4.1(b) is the chart representation of three methods (x-axis: Frame Index, y-axis: Sampled Frame Index). Figure 4.1(b) clearly indicates that TQS is a non-uniform sampling in nature. Figure 4.1(c)(d)(e) show the sample frames layout of three methods in video 1. The results demonstrate that

Table 4.2: TMSE comparisons of three methods for RWBS

Video No.	**TQS**	**US**	**SVS**
Video 1	26.89	91.03	77.04
Video 2	734.07	1577.36	900.74
Video 3	642.61	1077.36	986.43

the TQS can grasp the salient information (ball swing motion) ignored by the US in addition to capturing the semantic information provided by it, which can be easily justified in video 3.

Figure 4.2 shows the sample frames lay out of three methods in two videos (video 2 and video 3). Note that three methods have the same size of sampled frames for better comparisons.

4.3.2 Experimental Results on VTDF-based Temporal Quantization

To show the effectiveness and efficiency of the proposed method, we test the videos downloaded from YouTube. Note that the proposed method is also effective for other types of videos, such as MPEG videos and high definition videos. Here we randomly select 9 test videos from three categories to demonstrate. They are all 320 × 240 in frame size. Table 5.2 shows the summary of the test videos.

First, we do the non-uniform sampling on each video in the "Sports" category and lay out all sample frames sequentially in Figure 4.3. Video 1 is a bowling video to model the "ball swing" scene. Video 2 is a golf video to model a "full swing" scene in golf sports. Video 3 is a "shoot" scene in hockey sports. Figure 4.3 shows that the proposed method can clearly grasp the event scenes in three videos in a concise way.

Table 4.3: Summary of test videos for TVQ

Video Category	**Number of Videos**
Sports	3
Home	4
Others	2

Second, we compare the proposed method using VTDF (TVQ) with two methods. One is the traditional sampling method in current video players (Uniform). Another one is an existing method in video summary (Greed) [58][60]. Since Uniform is a sampling method to sample the frames uniformly, it may not be good enough to capture the important information during rapid video playback. Considering we integrate the TVQ with a video player in practice, it is straightforward to compare it with the uniform fast-forward method from the application point of view. The Greed method selects the frames that would introduce the smallest distortion at each iteration. The reason to compare the TVQ with the Greed is that the Greed uses a similar quantization method to sample frames with a similar distortion function. However, from quantization point of view, it will only sample the first frame as the quanta for each partition. Also the first video frame is always sampled. Considering the different characteristics of videos in nature, the Greed method may not effectively grasp the semantic information of video.

In comparison, the distortion function, TMSE is first applied to all three methods to calculate the quantization distortion and show the results in Figure 4.4. Note that for the same video, the same rate is applied to calculate the distortions of all three methods. Figure 4.4 shows that the TVQ has smaller distortions than other two methods in all test videos.

Figure 4.5 shows the R-D chart for all three methods by sampling the video 1 in the "Others" category. Figure 4.5 also demonstrates that given the same video and the same rate budget, the TVQ has smaller distortion cost compared with other two methods.

We can also compare the sampling speed. The Uniform will take 15 ms to get the sample frames. After the VTDF calculation, the Greed will take 76 ms to get the sample frames and the TVQ will take 104 ms. Although the VTDF calculation has to be done first before applying the TVQ to do non-uniform sampling, the sampling procedure can be done in real time.

Note that the proposed temporal quantization method is integrated in video players for rapid video navigation in practice. Figure 4.6 shows the whole system architecture, which can be described in two modules: processing and playing. In the processing module, a TXT file containing all extract key frame indices can be generated to feed into the playing module. In the playing module, the video player will only play the sampled frames and skip others in its intelligent fast-forward mode.

Note that given one video, only the VTDF calculation part needs to be done offline at once. In other words, the proposed temporal quantization can generate the sample frames in real time.

Figure 4.7 is the video player GUI built in Java. A TXT file contains key frame indices will be fed into the video player as an input for content-based fast-forward video playback. The "Load Video" module will pop up a dialog for the users to choose a video for playback. Note that for one video chosen for playback, its VTDF has been calculated offline. So the temporal quantization can be applied to find the optimal sample frames directly. The "Speed Factor" module is to set how fast to do fast-forward playback, denoted by S (e.g., $S = 4$ is a four times fast-forward). The "ProgressBar" is to model the playback progress of sampled frame sequence. Before committing time to the original video, the users can use TVQ to navigate the video content non-uniformly. It is a relevant practical application. For one video, the users can compare two different sampling effects freely.

4.4 VTDF-based Triangle Transition for Rapid Video Navigation Sampling

In this section, the above key frame sequence is combined with a set of parameters together by using a triangle transition function to generate the sampled frames in a non-uniform way. The reason to apply the parameterized triangle transition function to generate sample frames is that we hope to slow down when the video player is going to access the key frames and speed up after the access. As a result, sampling more frames around each key frame will let the users see more details during the fast-forward video playback.

Given total video frame number (before sampling) is N, the original video frame sequence can be indexed. The frame index at time n ($1 \leq n \leq N$) is denoted by $I(n)$,

$$I(n) = n. \tag{4.30}$$

A parameterized triangle transition is applied to find a transform from the original frame sequence domain to the time-based domain.

Figure 4.8 demonstrates how the triangle transition function works for non-uniform sampling. The non-uniform sampled video frame sequence can be indexed by $I_s(n;\theta)$ (for simple, $I_s(n)$) at time n, which is dependent on a parameter set θ,

$$\theta = (N_w, K, H, S), \tag{4.31}$$

where S is the fast-forward speed factor, $S \geq 1$. (e.g., given $S = 2$, the output for uniform sampling frame sequence is $2, 4, 6...$). N_w (in the original frame sequence domain, we use W) is to model the non-uniform sampling interval for each key frame-how many terms between n_w and n_k. In other words, it is the size of the sliding window. H is non-uniform sampling factor to model the sub-sampling rate (how slow to do playback) at time n_k. In general, $0 \leq H \leq S$.

Note that all parameters are in the time domain. In Figure 4.8, $D_s(n)$ is the sampling period (distance) at time n defined by parameters S, N_w, and H, and K is the position of key frame in the original frame sequence (in the time domain after sampling, we use the value of n_k as the time instance of K).

According to the above definitions of parameters in the time domain, the slope P (left side, the right side can be calculated in a similar way) of non-uniform sampling triangle can be calculated by:

$$P = \frac{H - S}{N_w}. \tag{4.32}$$

Then the $D_s(n)$ (left side, the right side can be calculated in a similar way) at time n can be calculated by:

$$D_s(n) = P(n - n_w) + S. \tag{4.33}$$

In here, the sum of $D_s(t)$ is described by:

$$\sum_{n=n_w}^{n_w+n_k} D_s(n) = K - W. \tag{4.34}$$

Then an equation based on above two equations can be generated as,

$$\sum_{n=n_w}^{n_w+n_k} [P(n - n_w) + S] = K - W. \tag{4.35}$$

Now, the relationship between W and N_w by calculating the above equation is determined as,

$$W = \frac{(S + H)(N_w + 1)}{2} \tag{4.36}$$

Now, we need to decide the value of n_w at time n, the boundary between uniform sampling and non-uniform sampling. The following formula is used to decide how to choose the value of n_w,

$$|I_s(n) - (K - W)| > S/2, \tag{4.37}$$

where if the above condition is true, $n_w = I_s(n) + S$, otherwise, $n_w = I_s(n)$.

Then the value of n_k can be known according to the value of n_w,

$$n_k = n_w + N_w. \tag{4.38}$$

Therefore, based on the above formulas, $I_s(n)$ at time n in non-uniform sampling can be calculated as follows,

$$I_s^{real}(n) = I_s^{real}(n-1) + D_s(n). \tag{4.39}$$

where $I_s^{real}(n)$ is the frame index in the continuous real domain. Then in practice we could take the round integers as the frame indices after sampling,

$$I_s(n) = round(I_s^{real}(n)). \tag{4.40}$$

The whole non-uniform sampling algorithm can be described as follows:

1. For frame at time n (initialization: $n = 1, I_s(1) = 1$) in the indexed video frame sequence and each key frame K, if $I_s(n) \in [n_k - N_w, n_k + N_w]$, go to 3, else go to 2.

2. If $mod(n/S) = 0$, n will be sampled (traditional uniform sampling), then $n = n + 1$, if $I_s(n) < N - 1$, go back to 1.

3. Calculate $I_s(n+1)$, then $n = n + 1$, if $I_s(n) < N - 1$, go back to 1.

Figure 4.10 shows the whole procedure to sample the frames. Figure 4.10(a) is the VTDF. Figure 4.10(b) is the generated key frame sequence. There are total four key frames extracted from the test video. Figure 4.10(c) shows the charts for cumulative $I_s(n)$ and $D_s(n)$ at time n for parameters, $S = 8$, $H = 0$, $N_w = 5$. Figure 4.10(d) is the console output from Eclipse Java IDE.

Figure 4.9 is the new video player GUI with the parameterized non-uniform sampling support. A TXT file contains key frame indices will be fed into video player as input for parameterized fast-forward video playback. In Figure 4.9, the x-axis of sampling chart is to model the sampled frame sequence and y-axis is the sampling speed descriptor, denoted by s. For one video frame at time n (initialization, $n = 1$, $I_s(1) = 1$) in the original video frame sequence, if $I_s(n) \in [n_k - N_w, n_k + N_w]$, $s = D_s(n)$, else $s = S$. As a result, s is to model how slow to sample this frame. In Figure 4.9, the first "ProgressBar" is to model the playback progress of sampled frame sequence. After one frame is played, it will move $I_s(n)$. The second "ProgressBar" is to model the changes of $D_s(n)$ for the sampled frame sequence. At time n, its selection value will be the value of $D_s(n)$.

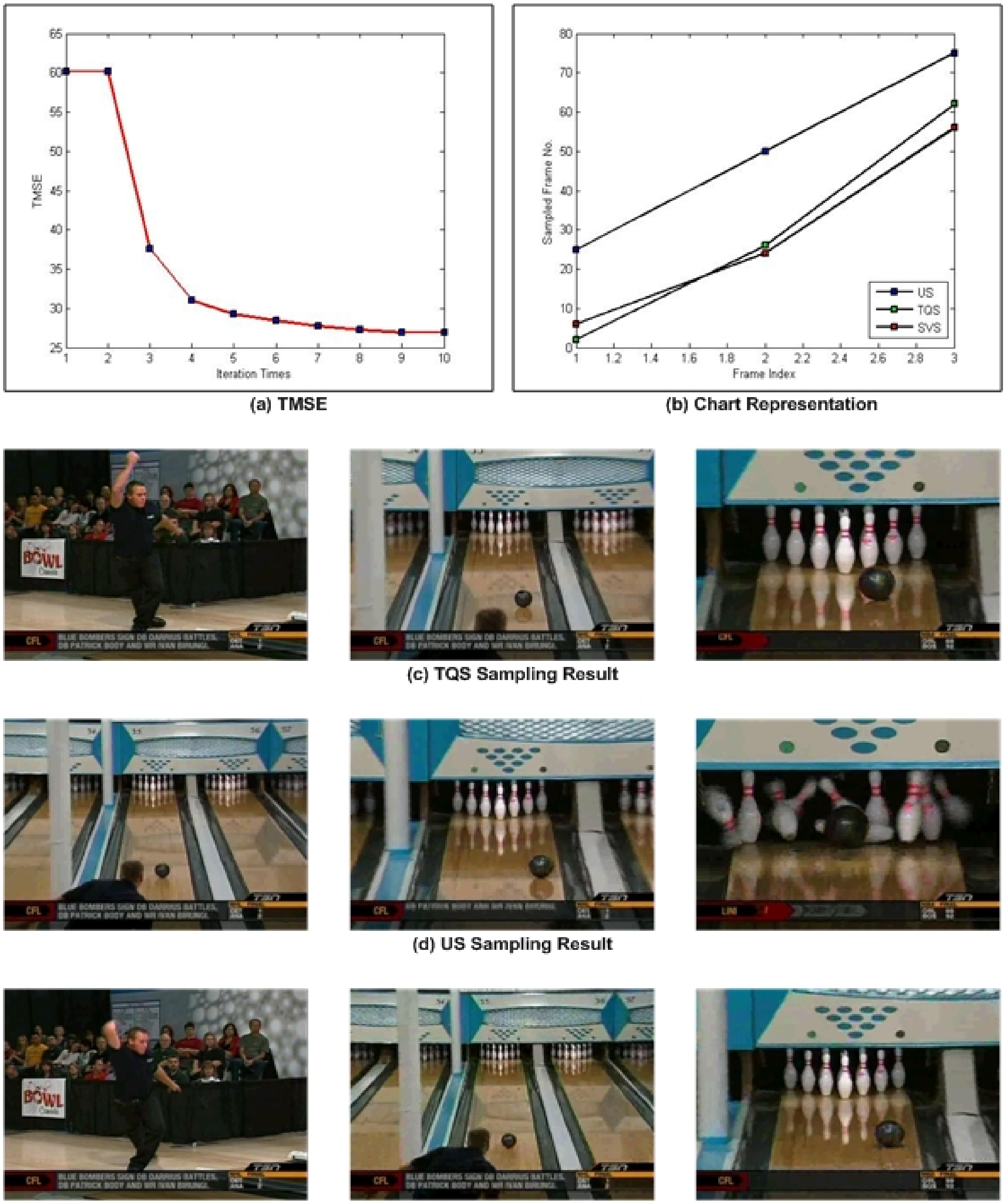
The implementation of this video player demonstrates that the proposed method is useful to the users in content-based rapid video navigation. After arranging a group of users to watch the sampling results with various rapid playback rates, we can conclude that the proposed method can grasp the salient information ignored by the uniform sampling in addition to capturing the semantic information provided by it.

We can also compare the built video player with the players in literature. The existing video players can be addressed in two perspectives: commercial and research. The examples of commercial-based players are Apple QuickTime Player [1], CyberLink PowerDVD [3], Microsoft Media Player [5], and Real Network RealOnePlayer

[6]. The commercial players are using a traditional way (uniform sampling) to do fast-forward playback that is not good enough to grasp the important semantic information. There are some players proposed in research literature, such as the players implemented in [22][19]. The problems of existing research players can be addressed in two aspects. First, they are dependent on video shot [19]. Second, they are dependent on domain knowledge, predefined rules, and preferences [22], thus lacking of generic. In addition, different from the emphasized aspects (shot boundary, scene change) in literature, we borrow the traditional vector quantization method to solve the video summarization problem from another perspective: given any specific time constraint (number of sampling frames), find the best samples of video.

4.5 Chapter Summary

A theoretical framework using the VTDF and statistical models is proposed in this Chapter. The VTDF models each video sequence as an one dimensional time series signal. Inspired by the concept of vector quantization, a new VTDF-based temporal quantization is applied in the time domain to summarize the video content dynamically. An existing method, the RWBS and a new VTDF-based method are proposed to find the best quantization quanta for a given time budget or rate budget. A new criterion using the TMSE is to calculate the quantization distortion. The proposed method is integrated into video players for fast-forward playback. One video player is to play the sample frames and skip the others during fast-forward playback. Another video player is to apply a triangle transition function with a set of parameters to do non-uniform sampling for fast-forward playback.



(a) TMSE

(b) Chart Representation

(c) TQS Sampling Result

(d) US Sampling Result

(e) SVS Sampling Result

Figure 4.1: Sampling of video 1 for RWBS

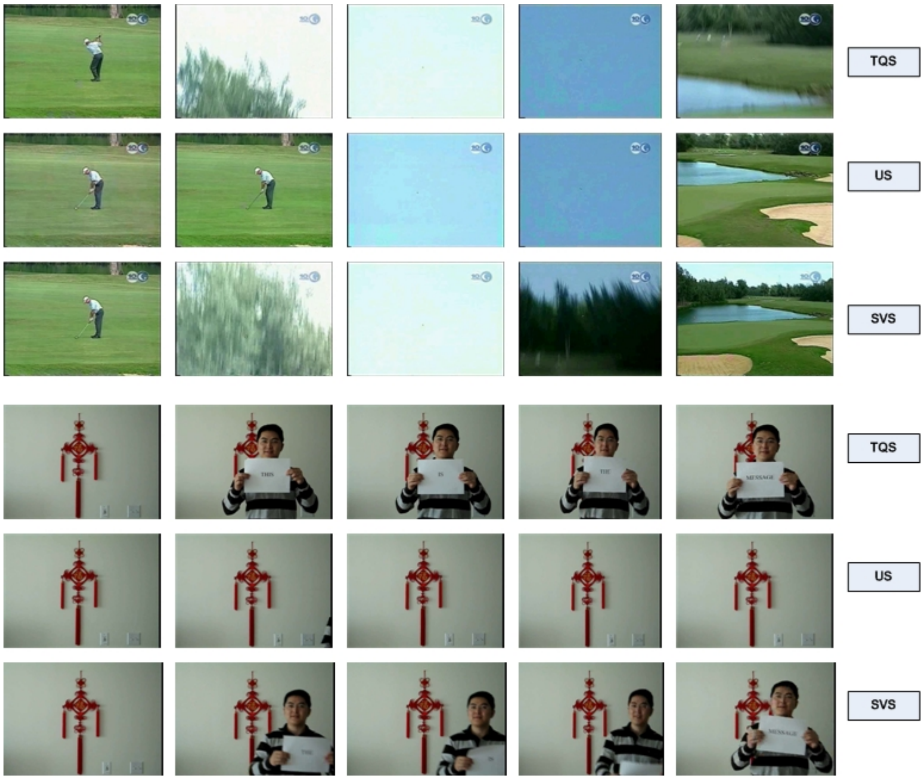


Figure 4.2: Sampling of video 2 and video 3 for RWBS

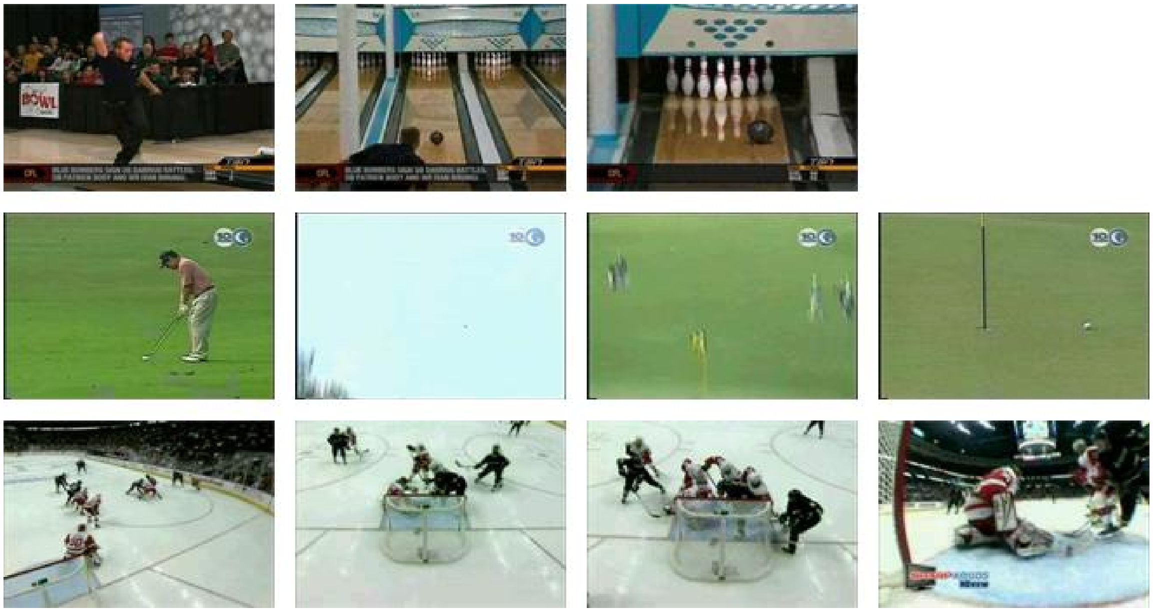


Figure 4.3: Sampling results for three test videos in sports category for TVQ

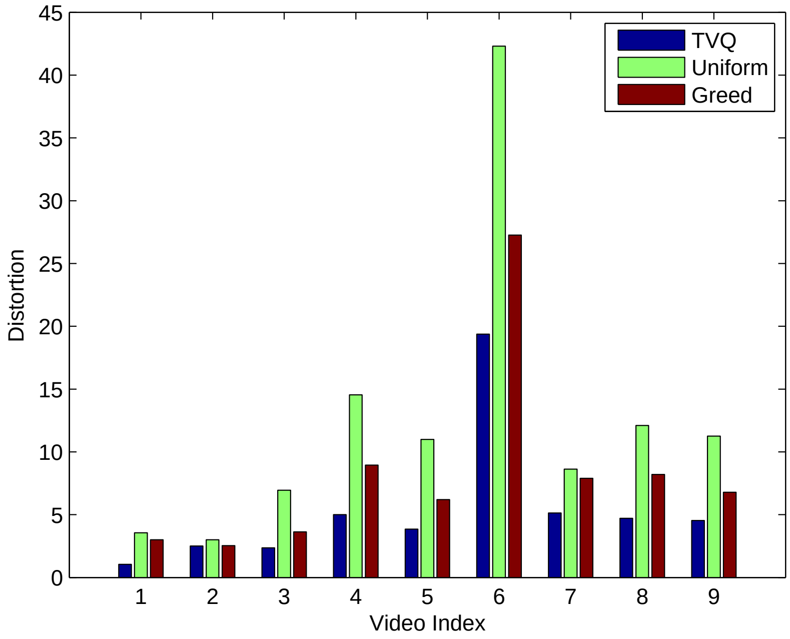


Figure 4.4: TMSE comparisons of three methods for TVQ

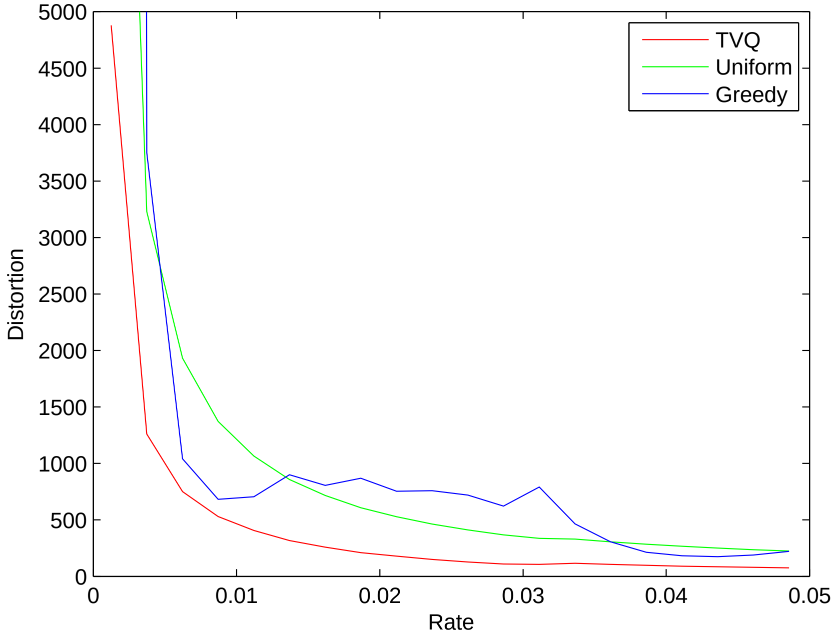


Figure 4.5: Rate-Distortion chart for test video 1 in others category for TVQ

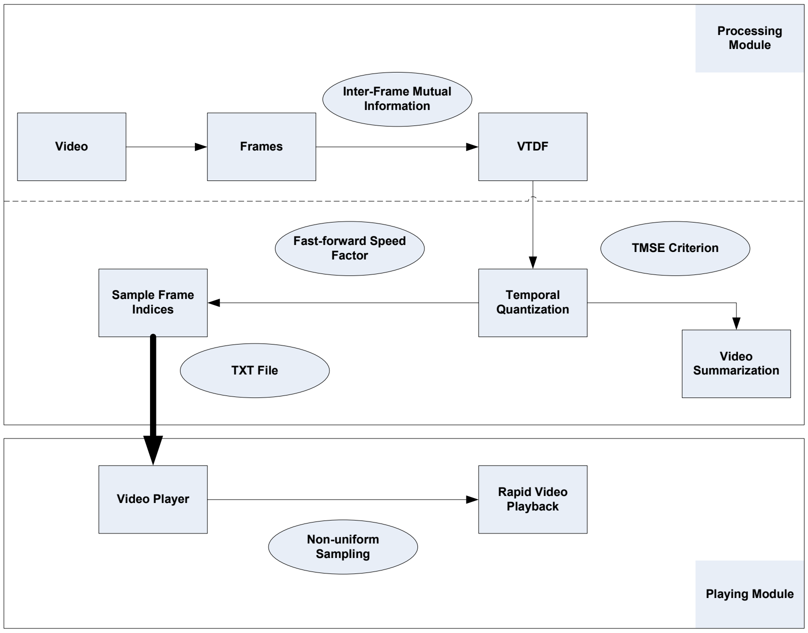
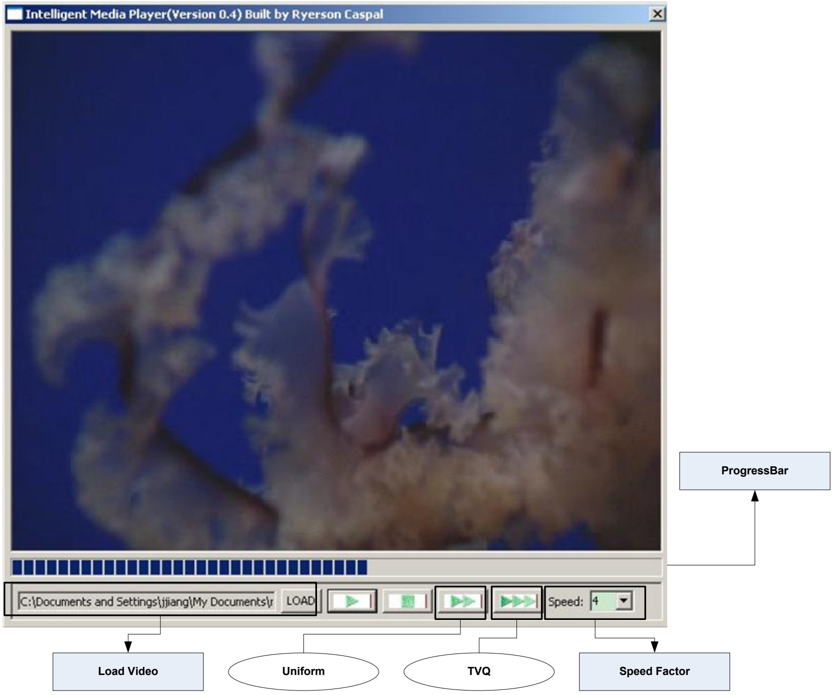


Figure 4.6: Temporal quantization-based rapid video playback system architecture


Intelligent Media Player(Version 0.4) Built by Ryerson Caspal
LOAD
ProgressBar
Load Video
Uniform
TVQ
Speed Factor

C:\Documents and Settings\jjiang\My Documents\

Speed: 4

Figure 4.7: GUI of video player with content-based fast-forward playback mode

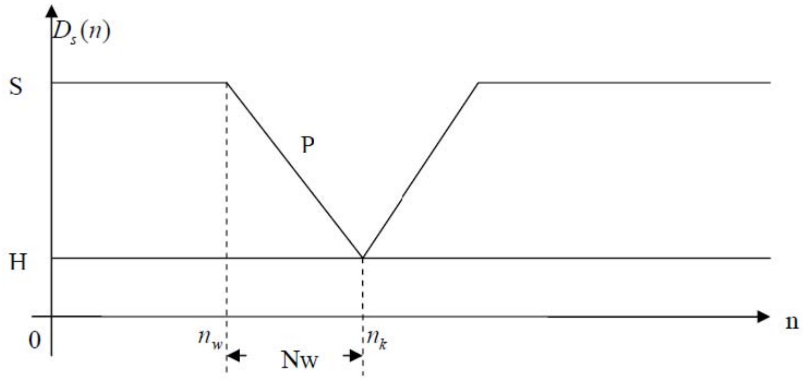


Figure 4.8: Four parameters-based non-uniform sampling

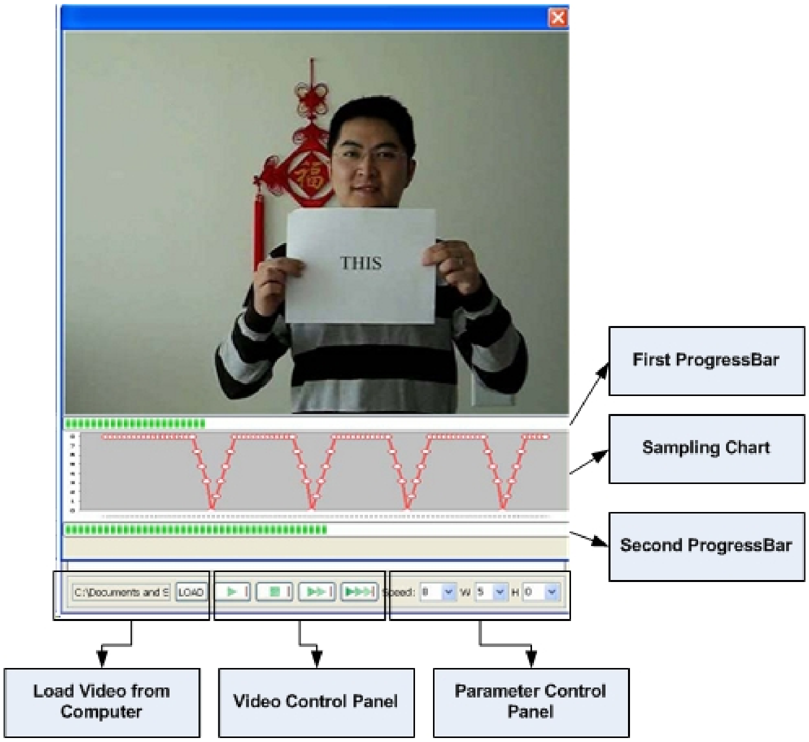
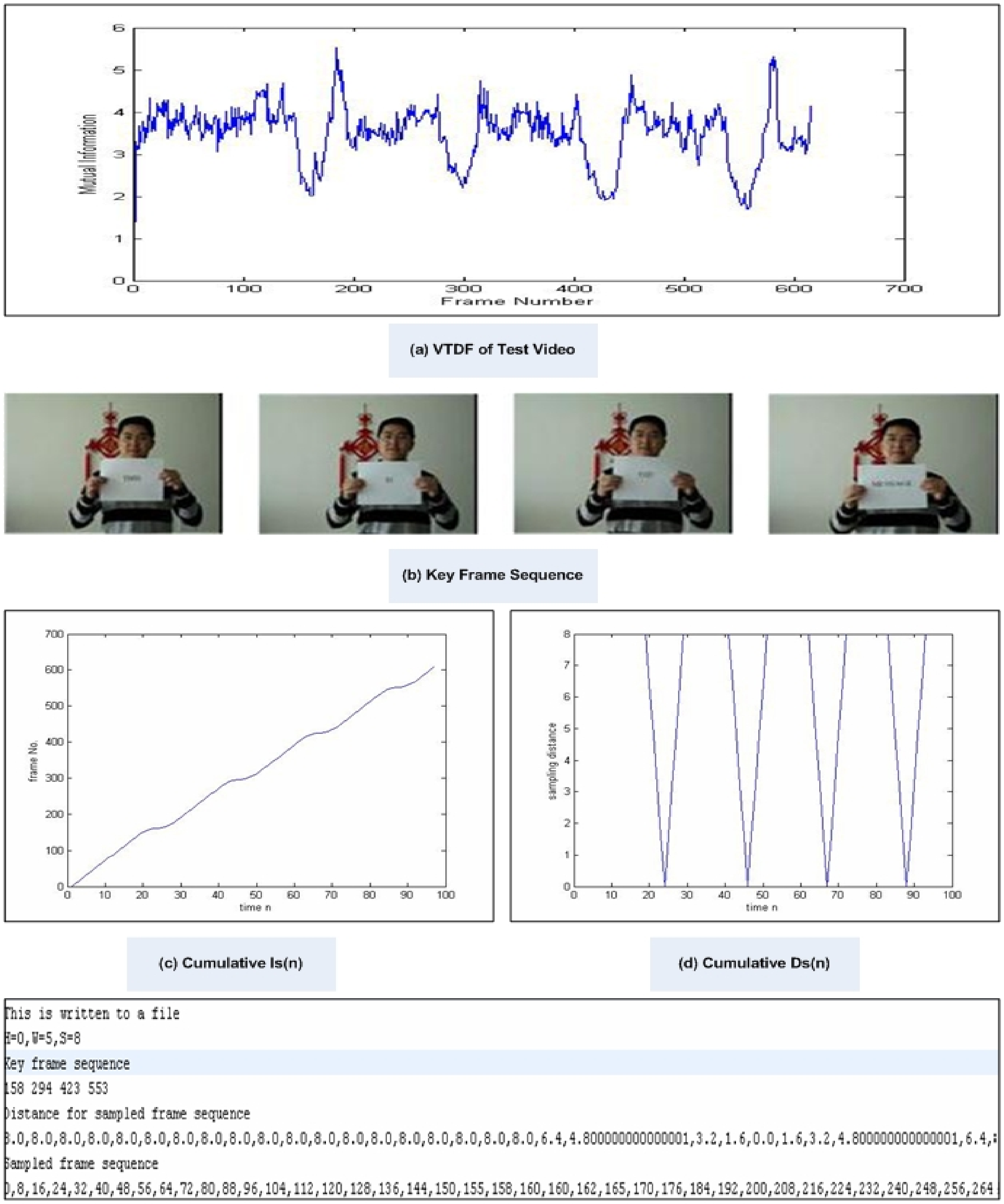


Figure 4.9: Four parameters-based video player



(a) VTDF of Test Video

(b) Key Frame Sequence

(c) Cumulative Is(n)

(d) Cumulative Ds(n)

```
This is written to a file
H=0,W=5,S=8
Key frame sequence
158 294 423 553
Distance for sampled frame sequence
8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,8.0,6.4,4.800000000000001,3.2,1.6,0.0,1.6,3.2,4.800000000000001,6.4,
Sampled frame sequence
0,8,16,24,32,40,48,56,64,72,80,88,96,104,112,120,128,136,144,150,155,158,160,160,162,165,170,176,184,192,200,208,216,224,232,240,248,256,264
```

(e) Eclipse Control Ouput

Figure 4.10: Triangle transition-based non-uniform sampling

Chapter 5

VTDF Mixture Model Video Thumbnail Extraction

In the previous Chapter, a new VTDF-based temporal quantization is applied to sample the video content in a concise and informative way for fast-forward playback in the built video players. One key frame is extracted in each temporal segment to generate the key frame sequence. The number of sample frames is decided by the users. However, the problems can be addressed in two aspects. First, it may not be good enough to sample only one frame to represent the whole temporal segment. Second, the speed factor decided by the users may cause redundancy of sample frames.

To solve the above problems, a new vector quantization method is proposed to extract video thumbnails in this Chapter. First, VTDF-based temporal quantization (VTDFTQ) is employed to segment video data in the time domain. The optimal number of temporal segments is determined by a new TMSE-based criterion. For each temporal segment, the mixture models are then employed to explore its spatial characteristics and determine the number of sample frames. Two mixture models are used: GM and ICAM. Accordingly, the vector quantization methods, GMVQ and ICAMVQ proposed in Chapter 3 are employed to summarize each temporal segment. The video frames that are the nearest neighbors to the quantization codebook are sampled to represent each temporal segment. A compact video thumbnails will be generated by combining the summary of each temporal segment accordingly.

The advantages of the proposed method over existing methods can be summarized in three aspects. First, it is generic and independent of domain features and knowledge. Second, different from the emphasized aspects (shot boundary, scene change) in literature, the VTDF is employed to describe the temporal dynamics of video content. The temporal characteristics of video content are explored by VTDFTQ. Third, the spatial characteristics of video content are explored by GMVQ and ICAMVQ. The video thumbnails are very useful for the users to manage their video collections. Before committing time to the original video, the users can quickly look at the thumbnails for rapid video browsing and navigation.

5.1 VTDF Mixture Model

There are two steps in the proposed video thumbnail extraction method. First, the VTDFTQ is applied to explore the temporal characteristics of video content. The purpose of VTDFTQ is to segment the whole video stream in the time domain. The optimal number of temporal segments is determined by a new TMSE-based criterion. Second, GMVQ and ICAMVQ are employed to each temporal segment and find the best sample frames. The BIC criterion is applied to determine the number of sample frames for each segment. Figure 5.1 shows the whole architecture of video thumbnails extraction system using the VTDF mixture model.

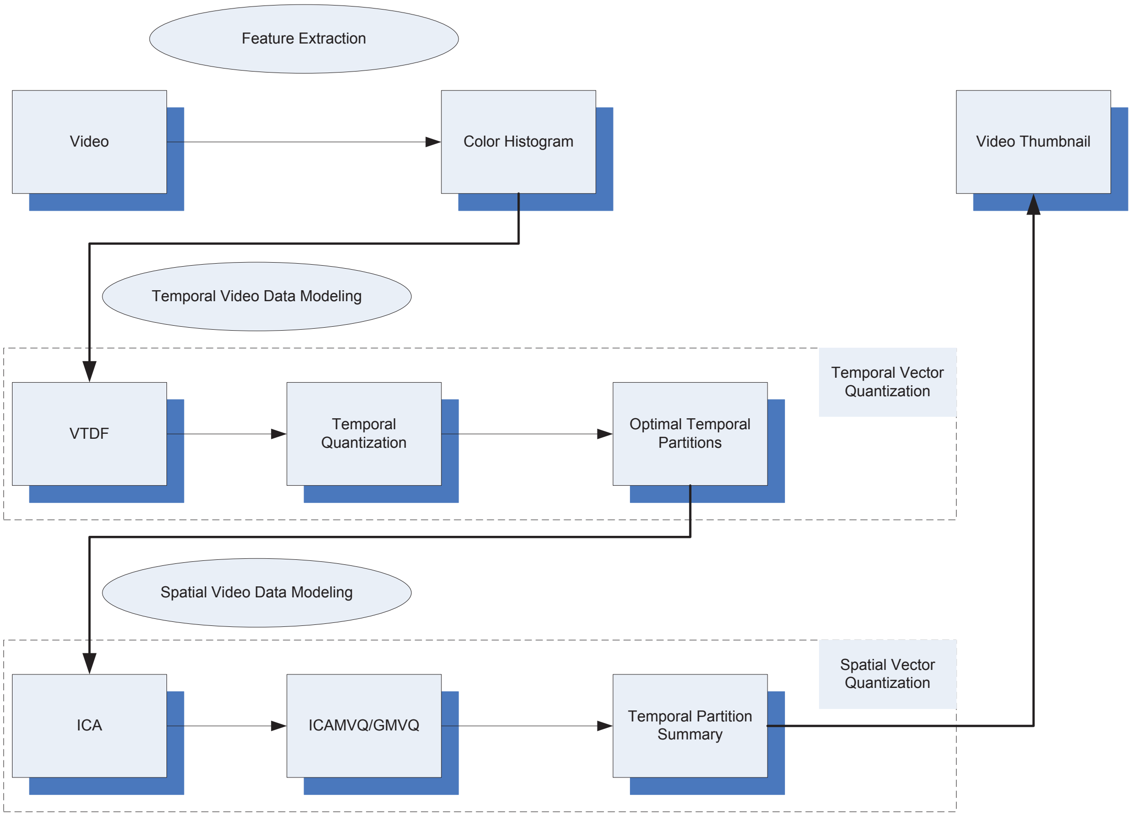

Feature Extraction
Video
Color Histogram
Video Thumbnail
Temporal Video Data Modeling
Temporal Vector Quantization
VTDF
Temporal Quantization
Optimal Temporal Partitions
Spatial Video Data Modeling
Spatial Vector Quantization
ICA
ICAMVQ/GMVQ
Temporal Partition Summary

Figure 5.1: VTDF mixture model architecture for video thumbnail extraction

5.1.1 Model Description

Given a class set Q with all video frames. After the VTDF-based temporal quantization, it can be represented as,

$$Q = \{Q_1, Q_2, ..., Q_M\}\,, \tag{5.1}$$

where M is the optimal number of temporal segments.

A new TMSE-based criterion is presented to determine the optimal number of temporal segments, the value of M.

$$\frac{TMSE(M-1) - TMSE(M)}{TMSE(M-1)} < \epsilon_4, \tag{5.2}$$

where $TMSE(M)$ is the TMSE value given a specific value for the number of temporal segments, $M(M \geq 2)$. ϵ_4 is a predefined small number.

In above criterion, the value of M to terminate the formula will be chosen as the optimal number of temporal segments.

For each temporal segment Q_i, GMVQ and ICAMVQ can be employed to segment it in the spatial domain and summarize it.

$$Q_i = \{Q_{i1}, Q_{i2}, ..., Q_{iK}\}\,, \tag{5.3}$$

where K is the optimal number of mixture components for this segment.

Using mixture model under the 2D ICA feature space, each element x_t within Q_i has a PDF form as,

$$f(x_t|\theta) = \sum_{j=1}^{K} \pi_j p_j(x_t|C_j, \theta_j), \tag{5.4}$$

where π_j represents the probability of the j-th mixture component, $p_j(x_t|C_j, \theta_j)$ is the probability to produce x_t from C_j for given parameter θ_j. θ is the class set of θ_j.

As a result, all x_t produced by C_j (maximum probability) can become a class set Q_{ij} within Q_i.

$$Q_{ij} = \left\{ x_t | j = \arg\min_j \left(p_j(x_t|C_j, \theta_j) \right) \right\}. \tag{5.5}$$

The maximum log likelihood function is used for parameter estimation.

$$\hat{\theta_j} = \arg\max_{\theta_j} \left(\log(p_j(x_t|\theta)) \right). \tag{5.6}$$

In GMVQ, θ_j can be represented as,

$$\theta_j = (\pi_j, \mu_j, \Sigma_j), \tag{5.7}$$

where μ_j and Σ_j are the mean and covariance matrix, respectively.

Therefore, $p_j(x_t|\theta_j)$ can be calculated as,

$$p_j(x_t|\theta_j) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_j|}} e^{-\frac{1}{2}(x_t-\mu_j)^T\Sigma_j^{-1}(x_t-\mu_j)}, \tag{5.8}$$

where $d(d = 2)$ is the dimension of frame vector x_t.

The EM algorithm is used to estimate the parameters iteratively.

$$\pi_j = \frac{N(Q_{ij})}{N(Q_i)}, \tag{5.9}$$

$$\mu_j = \frac{1}{N(Q_{ij})} \sum_{x_t \in Q_{ij}} x_t, \tag{5.10}$$

$$\Sigma_j = \frac{1}{N(Q_{ij})} \sum_{x_t \in Q_{ij}} (x_t - \mu_j)(x_t - \mu_j)^T, \tag{5.11}$$

where $N(.)$ is the operator to get the size of class set.

In ICAMVQ, x_t is modeled by a standard ICA model,

$$x_t = A_j S_j + b_j, \tag{5.12}$$

where A_j is the ICA basis coefficients and b_j is the mean coefficients for the mixtures. And S_j is the hidden source.

Accordingly, $p_j(x_t|\theta_j)$ can be calculated as,

$$\log(p_j(x_t|\theta_j)) = \log(p(S_j)) - \log(det|A_j|). \tag{5.13}$$

Therefore, given a ICAM component, its parameter set θ_j can be represented as,

$$\theta_j = (\pi_j, A_j, b_j). \tag{5.14}$$

Similarly, the EM algorithm is applied to calculate the parameters iteratively.

$$\pi_j = \frac{N(Q_{ij})}{N(Q_i)}, \tag{5.15}$$

$$b_j = \frac{\sum_{t=1}^{N(Q_i)} \pi_j}{\sum_{t=1}^{N(Q_i)} \pi_j x_t}, \tag{5.16}$$

$$\triangle A_j \propto \frac{p(x_t|C_j,\theta_j)\pi_j}{\sum_{j=1}^{K}(p(x_t|C_j,\theta_j)\pi_j)} \frac{\partial}{\partial A_j} \log(p(x_t|C_j,\theta_j)), \tag{5.17}$$

where $N(.)$ is the operator to get the size of class set.

The BIC is employed as an optimal criterion for the codebook size, the value of K.

$$BIC = -2 \times \log f(x_t|\theta) + p \times \log N(Q_i), \tag{5.18}$$

where $p(p = 3K)$ for both GMVQ and ICAMVQ is the number of parameters for estimation.

The MSE is used to measure the above quantization error. The MSE and the termination condition of iteration can be determined as,

$$e(Q) = \frac{1}{K}\sum_{j=1}^{K} \frac{\sum_{x_t \in Q_{ij}} \|q_{ij} - x_t\|_2}{N(Q_{ij})}, \tag{5.19}$$

$$q_{ij} = \frac{1}{N(Q_{ij})} \sum_{x_t \in Q_{ij}} x_t, \tag{5.20}$$

$$\frac{e(Q)^{(l-1)} - e(Q)^{(l)}}{e(Q)^{(l-1)}} < \epsilon_5, \tag{5.21}$$

where $e(Q)^{(l)}$ is the quantization error in the l-th iteration, $l > 1$. ϵ_5 is a predefined small number.

To summarize the class set Q_{ij}, the video frame that is the nearest-neighbor of its quanta needs to be found. The representative video frame of class set Q_{ij}, denoted by s_{ij}, can be determined as,

$$s_{ij} = \arg\min_{x_t \in Q_{ij}} \left\| x_t - q_{ij} \right\|_2, \tag{5.22}$$

The total summarization result for Q_i can be represented by s_i,

$$s_i = \{s_{i1}, s_{i2}, ..., s_{iK}\}. \tag{5.23}$$

As a result, the video thumbnail can be represented as,

$$s = \{s_1, s_2, ..., s_M\}. \tag{5.24}$$

The above VTDF mixture model can summarize the whole video stream in a compact way. Compared with existing video thumbnail extraction methods in literature, the advantages of the proposed method can be emphasized with three aspects. First, the VTDF is an effective and efficient tool for video data modeling. Second, the VTDFTQ is applied to explore the temporal characteristics of video data and find the optimal quanta and partition in the time domain. Third, the GMVQ and ICAMVQ are employed to each temporal segment to find the optimal sampling frames. The GM is an effective density estimation model for multi-modal characteristics of data. In other words, the GM is effective to explore the Gaussian characteristics of video data in the spatial domain. The ICAM is an effective density estimation model for multi-modal data analysis. In other words, the ICAM is effective to explore the spatial characteristics of non-Gaussian structure of video data. Therefore, the new video thumbnail extraction method explores both temporal-spatial characteristics and generates an optimal video thumbnail to abstract video data in a concise and informative way.

5.2 Experimental Results

5.2.1 Experimental Results on VTDF-GM

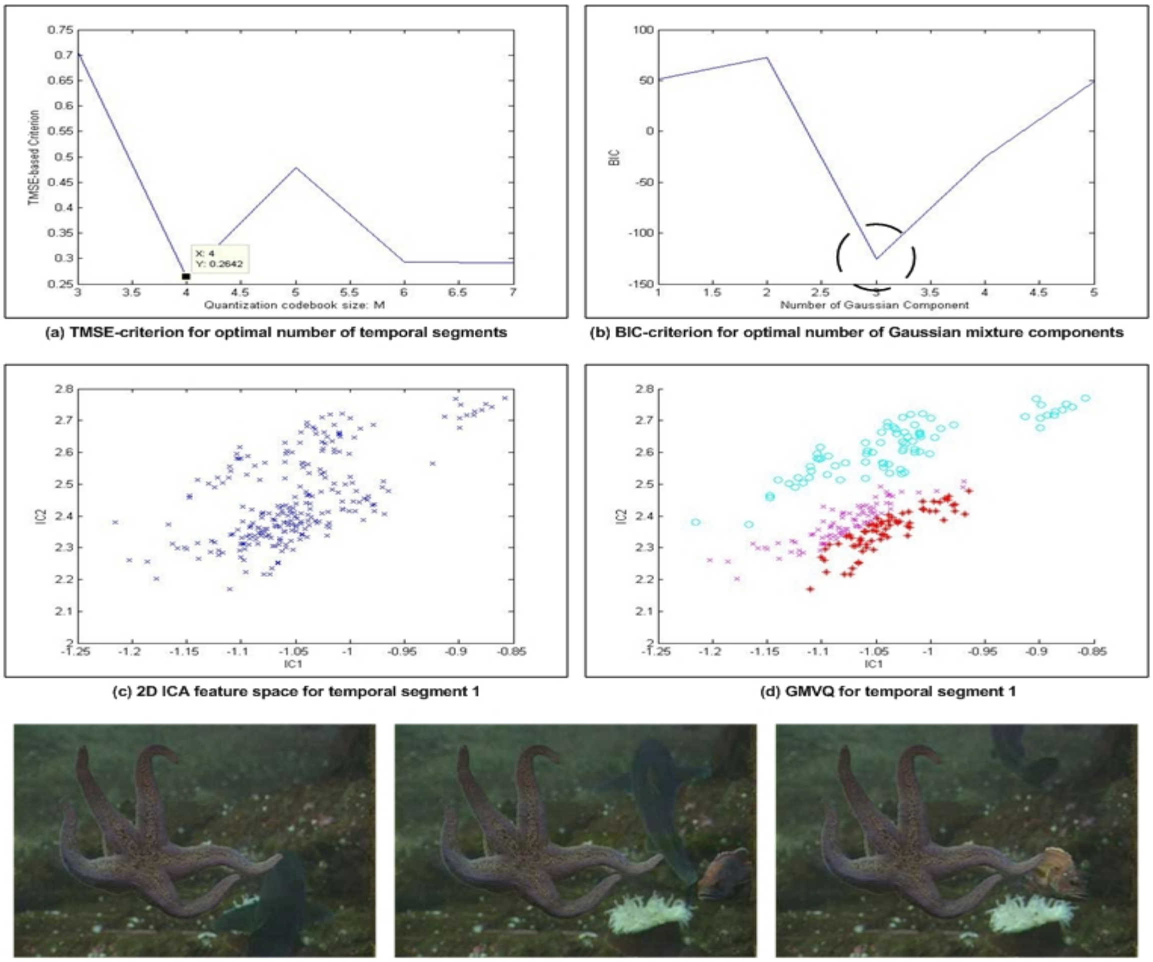
Three videos are tested to show the effectiveness and efficiency of our method. They are all MPEG2 with 320 × 240 frame size and 29.9 frame rate. Table 5.1 shows the summary of test videos. "Total Shots" column lists the shots number of each test video obtained by a literature method [98]. "Sample Frames" column lists the size of each video thumbnail.

First, we demonstrate the performance of VTDFTQ and GMVQ for video

Table 5.1: Summary of test videos for VTDF-GM

Video No.	**Total Frames**	**Total Shots**	**Sample Frames**
Video 1	803	4	7
Video 2	396	5	10
Video 3	342	1	4

1. The TMSE-based criterion is calculated and shown in Figure 5.2(a), from which it can conclude that the optimal number of temporal segments is 4. For the first temporal segment, the GMVQ is applied to explore its spatial characteristics. Figure 5.2(c) is the raw 2D ICA feature space. The optimal number of GM components is determined by the BIC-based criterion. From Figure 2(b), it can acquire 3 is the optimal number. As a result, all frames in the first temporal segment are quantized into three GM components. In Figure 5.2(d), one color represents one component. The summary of this segment with three frames is shown in Figure 5.2(e). In Figure 5.2(e), the complete fish fade-in-fade-out movement is captured effectively.



(a) TMSE-criterion for optimal number of temporal segments

(b) BIC-criterion for optimal number of Gaussian mixture components

(c) 2D ICA feature space for temporal segment 1

(d) GMVQ for temporal segment 1

(e) Video thumbnail for temporal segment1

Figure 5.2: Video thumbnail extraction using VTDF and GMVQ

Second, we compare with the SVS method and use video 2 and video 3 to demonstrate. Video 2 is a golf video to model a "full swing" scene in golf sports. In Figure 5.6, the first two rows are the video thumbnail result of VTDF-GM, while the last one row is the results of SVS. From Figure 5.6, we can conclude that the VTDF-GM method can grasp the salient information ("full swing" event) ignored by the SVS method in addition to capturing the semantic information provided by it. This conclusion can be justified easily by using video 3.

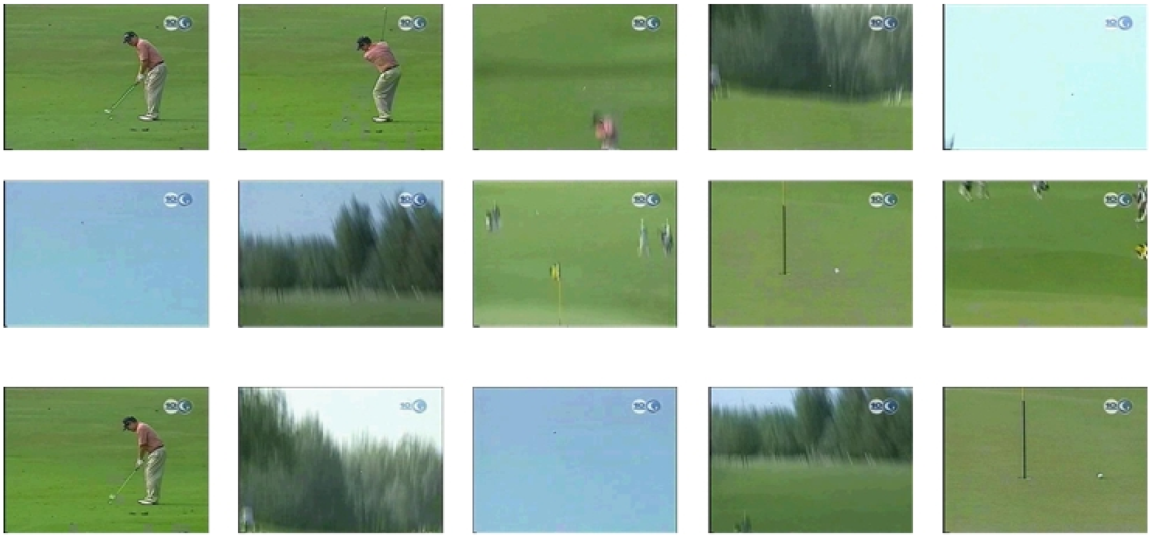


Figure 5.3: Comparisons of two thumbnails of video 2 for VTDF-GM

The video 3 is a home-recorded fireworks video. The video thumbnails of two methods are shown in Figure 5.7. In Figure 5.7, the four frames in the first row is the result of VTDF-GM. The one frame on the bottom is the SVS method result. Compared with only one frame in the SVS method, the thumbnails obtained by VTDF-GM not only models the whole fireworks scene completely, but also captures the highlight effectively. From the SVS thumbnail, we cannot even know that it is a video on light or fireworks.

5.2.2 Experimental Results on VTDF-ICAM

Three videos are tested to show the effectiveness and efficiency of our method. They are all downloaded from YouTube with 320 × 240 frame size. Table 5.2 shows the summary of test videos.

First, we demonstrate the performance of VTDF-ICAM for video 1. The TMSE-based criterion is shown in Figure 5.5(b), from which it can conclude that the optimal

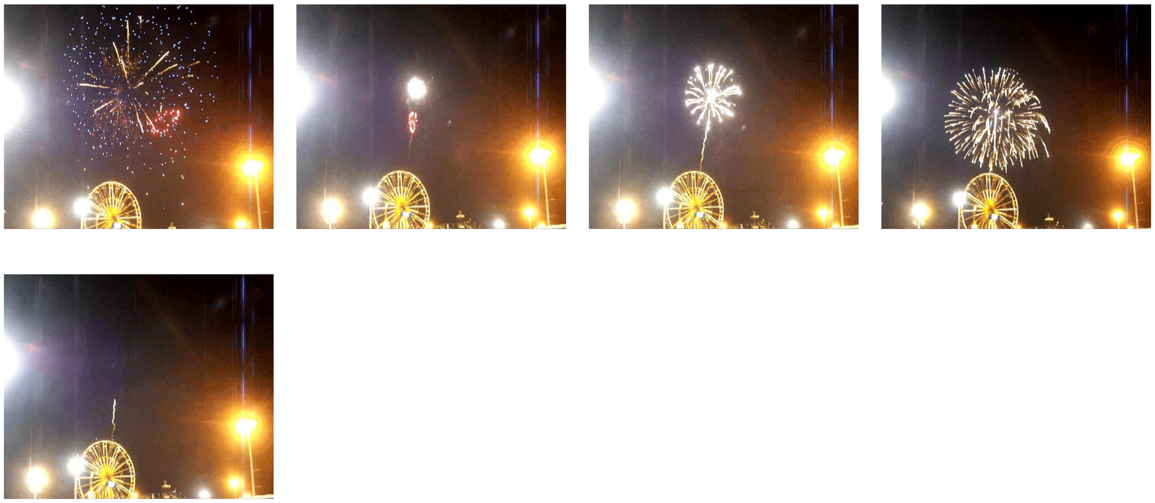


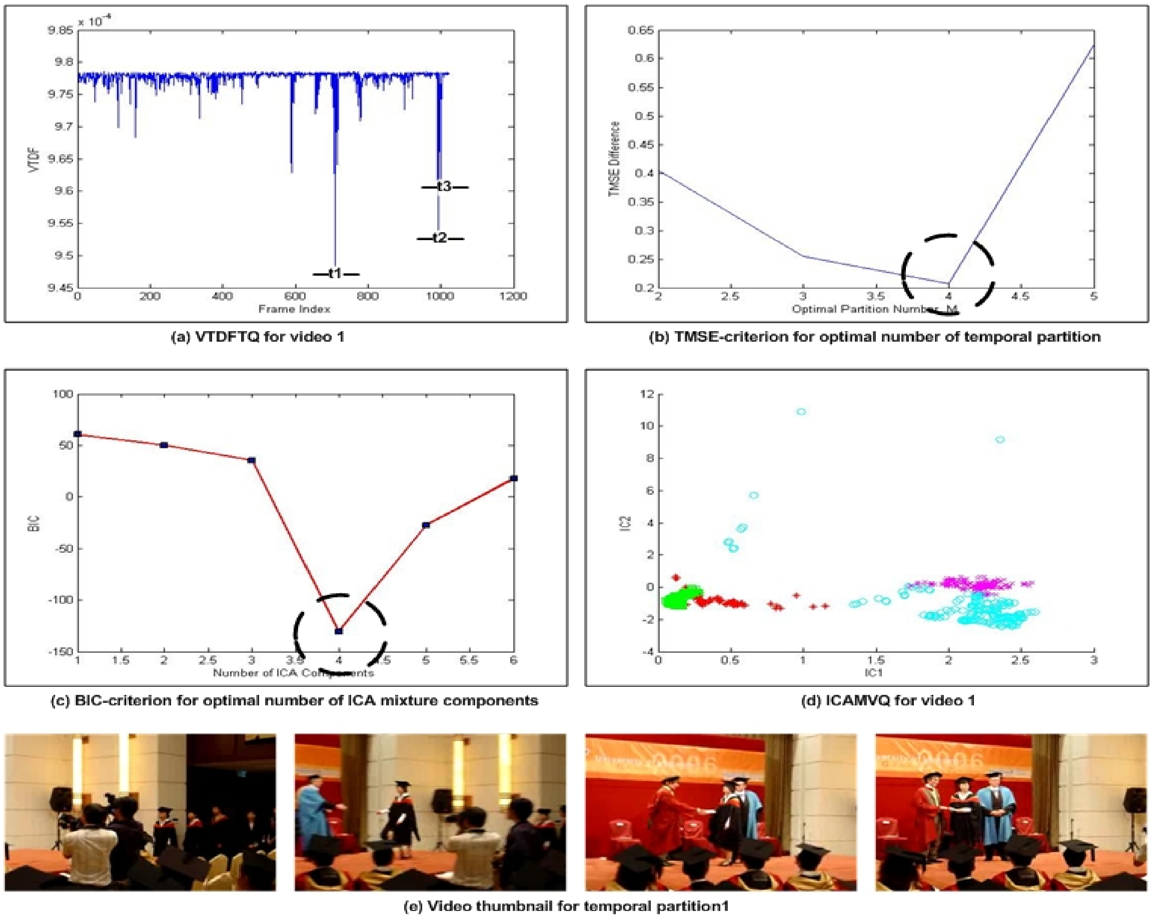
Figure 5.4: Comparisons of two thumbnails of video 3 for VTDF-GM

Table 5.2: Summary of test videos for VTDF-ICAM

Video No.	**Frame Rate-fps**	**Total Frames**	**Total Shots**	**Sample Frames**
Video 1	25	1024	1	8
Video 2	30	1151	7	9
Video 3	6	234	1	6

number of temporal segments is 4. In Figure 5.5(a), three boundaries are labeled. ICAMVQ is applied to each segment and the optimal number of ICAM component is determined by the BIC-based criterion. From Figure 5.5(c), it can acquire 4 is the optimal number of sample frames in the first temporal segment. In the 2D ICA subspace, all frames in the first segment are quantized into four ICAM components. In Figure 5.5(d), one color represents one component. Figure 5.5(e) shows a 4-frame thumbnail for this segment.

Second, we compare the VTDF-ICAM with the SVS and use video 2 and video 3 to demonstrate. Video 2 is a curling video. In Figure 5.6, the first two rows are the



(a) VTDFTQ for video 1

(b) TMSE-criterion for optimal number of temporal partition

(c) BIC-criterion for optimal number of ICA mixture components

(d) ICAMVQ for video 1

(e) Video thumbnail for temporal partition1

Figure 5.5: Video thumbnail extraction using VTDF and ICAMVQ

result of VTDF-ICAM, while the last two rows are the results of SVS. From Figure 5.6, it can conclude that our method can grasp more salient information ignored by the SVS method in addition to capturing the semantic information provided by it. This conclusion can be justified easily by using video 3.

The video 3 is a home-recorded fireworks video. The video thumbnails of two methods are shown in Figure 5.7. In Figure 5.7, the first six frames are the video thumbnails of VTDF-ICAM. Compared with only one frame in the SVS method, it not only models the whole fireworks scene completely, but also captures the highlight

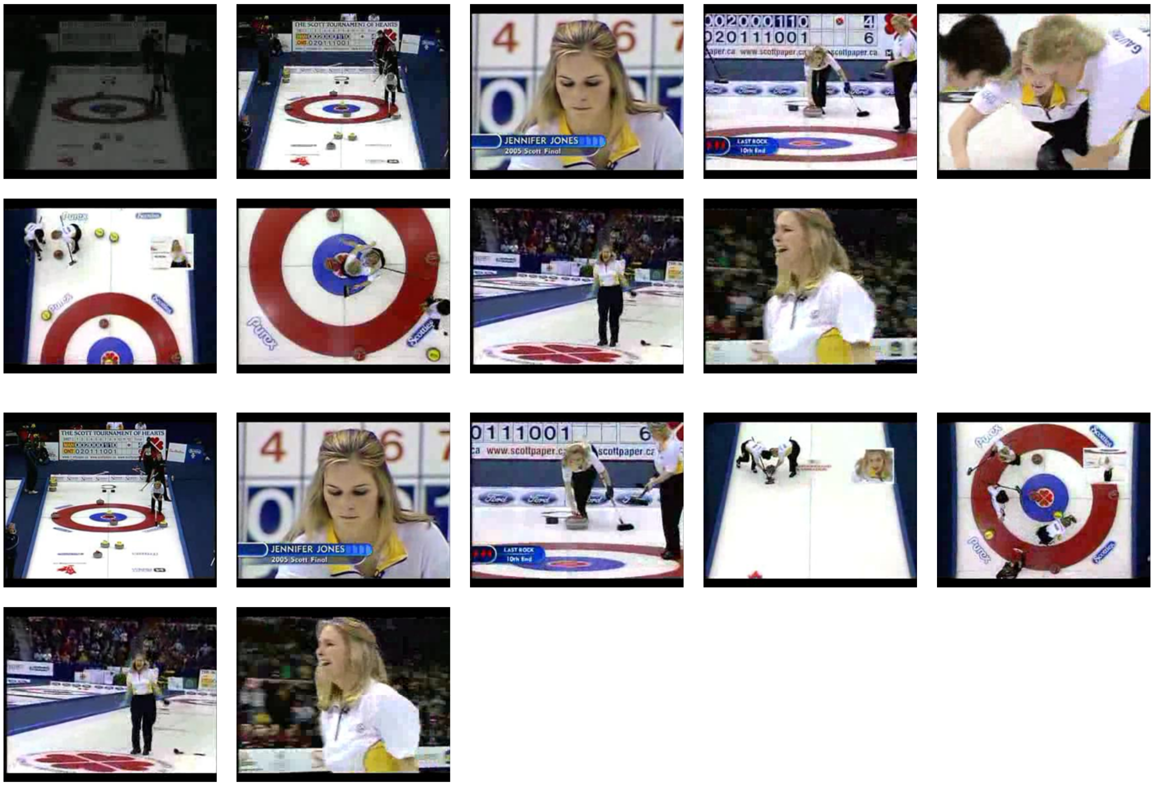


Figure 5.6: Comparisons of two thumbnails of video 2 for VTDF-ICAM

effectively.

5.3 Chapter Summary

This Chapter tries to solve the problem in dynamic video summary from another point of view: a compact video thumbnails. In particular, the VTDF is applied to do video segmentation. The optimal number of temporal segments is determined by the TMSE-based criterion. For each temporal segment, GMVQ and ICAMVQ can be employed to find the optimal sample frames and then generate the video

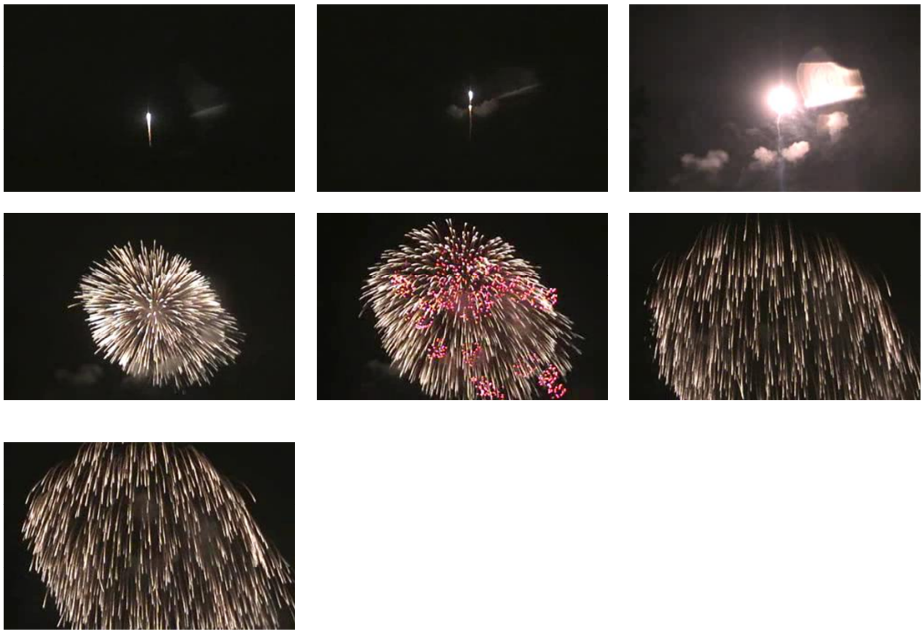


Figure 5.7: Comparisons of two thumbnails of video 3 for VTDF-ICAM

thumbnails. The temporal characteristics of video data are explored by the VTDF-based segmentation. The spatial characteristics of video data are explored by the GMVQ and ICAMVQ. The proposed method is useful for the users to manage their video collections. Given a video, the users can look at the generated video thumbnails to understand what is the video about before committing time to the full video.

Chapter 6

Video Similarity Measure and Event Detection

In this Chapter, a new video semantic similarity measure model using the VTDF and dynamic programming is presented first. In particular, the VTDF is calculated by using inter-frame dependency. The VTDF is used to explore the time density of video activities and model the video data in a compact and effective way. A temporal partition is then applied to divide each video stream into equal sized segments in the time domain. The VTDF-based correlation coefficient is calculated to measure the similarity between two equal sized temporal segments. As a result, the video similarly measure is a combination of pair-to-pair similarity of temporal segments. Dynamic programming is employed to solve the above multiple-to-multiple mapping problem.

Compared with the existing methods in literature, the advantages of the proposed method can be addressed in three aspects. Firstly, the VTDF is employed to model each video sequence in a compact and effective way. It is uncommon that two videos are semantically similar with different time density rhythms. In other words, the

VTDF is effective to index a video by using the density changes in time. Secondly, the whole video stream is considered to measure the video similarity rather than only several key frames. Thirdly, the distance measure combines both visual characteristics and temporal information together to measure the video similarity effectively.

A generic-oriented method to detect and recognize semantic video events is then proposed. Specifically, the video event detection problem is formulated as a generic supervised learning problem. The VTDF is employed as the video data modeling tool. A similar temporal quantization is employed to find the optimal VTDF-based feature vectors for a given video sequence and makes all video sequences have the same computational dimensions. Correlation is calculated to measure the similarity between two video sequences. Given a video sequence for event detection and classification, the predefined event sequence with the highest correlation coefficient value can be considered as the detection result in the proposed method.

An one-hour golf video is used as the case study to demonstrate the detection performance of the proposed method. Note that the proposed method can also be applied to detect the events in other sports videos, such as bowling and hockey.

Compared with existing methods, the advantages of the proposed method can be addressed in three aspects. First, the VTDF is an effective video data modeling tool to explore the essential temporal characteristics of video content. Second, temporal quantization is employed to generate the same dimensional compact VTDF-based feature vectors for each video sequence. The VTDF-based features are generic-oriented and independent of domain knowledge. Third, it does not need model training, learning, and parameter estimation in general.

6.1 VTDF-based Video Similarity Measure Model

6.1.1 Model Description

In this thesis, video similarity measure is modeled as a generic multiple-to-multiple matching process. To do that, one video stream is divided into equal sized temporal segments in the time domain first.

Let X be one video sequence with M frames, and Y be the other video sequence with N frames. Both of them are divided into equal sized segments in the time domain as:

$$X = \{x_1, x_2, ..., x_U\}, \tag{6.1}$$

$$Y = \{y_1, y_2, ..., y_V\}, \tag{6.2}$$

where x_i $(1 \leq i \leq U)$ is the i-th segment in video X and y_j $(1 \leq j \leq V)$ is the j-th segment of video Y. And U and V is the number of temporal segments of X and Y, respectively.

$$U = M/S, \tag{6.3}$$

$$V = N/S, \tag{6.4}$$

where S is a predefined parameter to specify the length of each temporal segment.

The segment x_i can be represented by its VTDF as,

$$x_i = \{I(t)|t \in [(i-1)S + 1, iS]\} \tag{6.5}$$

Note that the segment y_j can be represented in a similar way.

Note that all temporal segments have the same length. In other words, they are equal sized. As a result, the similarity between X and Y can be determined based on the combination of similarities between two temporal segments,

$$score(X, Y) = f(d(x_i, y_j)), \tag{6.6}$$

where $d(.)$ is a function to model the similarity between two temporal segments. And $f(.)$ is a function to measure the sequence similarity based on the segment similarity.

6.1.2 Correlation-based Similarity of Temporal Segments

Based on the model description in last section, the similarity of two individual temporal segments needs to be measured first.

Given two equal sized temporal segments x_i and y_j, correlation can be calculated to measure their dissimilarity as,

$$d(x_i, y_j) = 1 - \left| \frac{cov(x_i, y_j)}{\sigma(x_i)\sigma(y_j)} \right|, \tag{6.7}$$

where σ is standard deviation and $cov(.)$ means covariance operator.

The reason to apply correlation coefficient to measure the semantic similarity between two temporal segments can be addressed in two aspects. First, all temporal sequences have the same length of VTDF-based features. Second, two segments with high correlation coefficient mean that they are semantic correlated. In other words, two temporal segments with a small value obtained in the above formula mean that they are highly similar in content.

6.1.3 Dynamic Programming-based Similarity of Video Sequences

Dynamic programming is developed to measure the video similarity by combining multiple similarity scores of temporal segments. It is an optimal multiple-to-multiple mapping in nature. The effectiveness of dynamic programming to solve this kind of mapping problem is demonstrated in [59][82].

Two alignment functions $p(.)$ and $q(.)$ are defined for video X and video Y, respectively, where $1 \leq p(i) \leq U$ and $1 \leq q(i) \leq V$. Note that $p(i)$ and $q(i)$ are to index the temporal segment for similarity calculation at the i-th iteration, $1 \leq i \leq L$, where L is the length of optimal path.

The following constraints are applied to generate an optimal path based on the ordered set formed by the two alignment functions.

$$p(1) = 1, q(1) = 1, \tag{6.8}$$

$$p(L) = U, q(L) = V, \tag{6.9}$$

$$0 \leq p(i) - p(i-1) \leq 1, 0 \leq q(i) - q(i-1) \leq 1, \forall i \geq 1, \tag{6.10}$$

$$p(i) - p(i-1) + q(i) - q(i-1) \geq 1. \tag{6.11}$$

Therefore, the similarity score of two video sequences can be defined as,

$$score(X, Y) = 1 - \frac{1}{L} \sum_{i=1}^{L} \left(d(x_{p(i)}, y_{q(i)}) \right). \tag{6.12}$$

Table 6.1 is an example used to demonstrate how the above dynamic programming works. In Table 6.1, the columns V11-V16 and V21-V24 represent the segment indices of test video 1 (six segments) and video 2 (four segments). In Table 6.1, the table cell is the dissimilarity score calculated using formula (6.7). And all bold cells become an optimal path using dynamic programming formula group (6.8)-(6.11). According to the formula (6.12), it can obtain 0.41 as the similarity score between the video 1 and video 2.

The above similarity measure function combines both visual characteristics and

Table 6.1: Dissimilarity matrix between video1 and video2

	V11	V12	V13	V14	V15	V16
V21	**0.25**	0.81	0.84	0.82	0.81	0.81
V22	0.80	**0.73**	**0.50**	0.81	0.90	0.78
V23	0.81	0.80	0.83	**0.45**	0.81	0.80
V24	0.75	0.81	0.83	0.82	**0.78**	**0.80**

temporal information together to evaluate the video similarity effectively. Given two video sequences, the higher similarity score means that they are more semantically similar.

6.1.4 Experimental Results

To show the effectiveness of the proposed method, one golf video recorded from TV is used to test the detection performance. By applying different coding formats and parameters, two test videos are created: MPEG and AVI. The MPEG video is encoded in MPEG2 with 29.9 frame rate, 352×240 frame size, and $101,144$ frames in total. The AVI video has the same content as the MPEG video with different parameters: AVI encoding, a frame rate of 25, frame size of 352×240, aspect ratio $16 : 9$, and simple color correction.

The sequences in both videos are predefined into three classes, “full swing”, “non-full-swing”, and “irrelevant”. “Full swing” class can be considered as a repetitive scene pattern in golf sports, which begins with a zoom-in to capture the player’s preparation for his hit, and then followed by a quick camera motion to track the ball. The last scene locates the slowly moving ball. “Non-full-swing” class describes all soft ball hits in golf sports, such as fairway shot. “Irrelevant” class includes all other scenes rather than play, such as talk and break, audience scene, TV scene transition effect, global still view and so on.

For better evaluation, the two videos have been pre-edited to remove all advertisements. And all video sequences are manually annotated and classified into one of three predefined classes. There are 209 video sequences for each test video in total. Table 6.2 lists the summary of test video sequences.

For each class, six video sequences are randomly selected from two videos to test.

Table 6.2: Summary of test video sequences

	Sequence Numbers
Class 1-Full Swing	35
Class 2-Non-Full-Swing	141
Class 3-Irrelevant	33
Total sequences	209

Table 6.3 shows the video similarity score between any two test video sequences that belongs to the Class 1 in the MPEG video and AVI video. Table 6.4 shows the video similarity score between any two test video sequences that belongs to the Class 2 in the MPEG video and AVI video. Table 6.5 shows the video similarity score between any two test video sequences that belongs to the Class 3 in the MPEG video and AVI video.

In Table 6.3, Table 6.4, and Table 6.5, the columns V111-V116, V121-V126, and V131-V136 represent the sequence indices of the MPEG video for three classes. For example, V116 means the sixth video sequence in the class 1. Similarly, the columns V211-V216, V221-V226, and V231-V236 represent the sequence indices of the AVI video for three classes.

Note that two sequences with the same class No. and index are same in content with different parameters. For example, V111 and V211 are same in content and number of frames, but different in parameters. Table 6.3, Table 6.4, and Table 6.5 all demonstrates the effectiveness of our video similarity measure method. In experiments, each temporal segment consists of 60 ($S = 60$) video frames.

The proposed method can also be applied to detect the near-duplicate videos.

Table 6.3: Similarity score for golf video Class 1

	V111	V112	V113	V114	V115	V116
V211	**0.79**	0.50	0.47	0.47	0.33	0.46
V212	0.27	**0.78**	0.38	0.44	0.41	0.29
V213	0.34	0.34	**0.74**	0.38	0.39	0.49
V214	0.45	0.44	0.47	**0.74**	0.36	0.30
V215	0.44	0.33	0.46	0.54	**0.69**	0.37
V216	0.43	0.42	0.55	0.63	0.44	**0.77**

Table 6.4: Similarity score for golf video Class 2

	V121	V122	V123	V124	V125	V126
V221	**0.78**	0.49	0.28	0.24	0.31	0.17
V222	0.43	**0.68**	0.48	0.16	0.42	0.19
V223	0.36	0.66	**0.84**	0.43	0.43	0.37
V224	0.45	0.44	0.47	**0.74**	0.27	0.41
V225	0.34	0.23	0.34	0.44	**0.81**	0.26
V226	0.47	0.52	0.50	0.34	0.43	**0.78**

Four video sequences are used to demonstrate. As a ground truth, video 1 and video 2 have the same semantic content (same video with different parameters). Video 1 and video 3 have the same content but video 1 is 25 frames less. Video 1 and video 4 belong to the same ”full swing” class. Therefore, the similarity score between video

Table 6.5: Similarity score for golf video Class 3

	V131	V132	V133	V134	V135	V136
V231	**0.75**	0.27	0.19	0.43	0.27	0.52
V232	0.47	**0.73**	0.60	0.43	0.18	0.16
V233	0.46	0.46	**0.85**	0.50	0.44	0.54
V234	0.33	0.54	0.57	**0.78**	0.29	0.32
V235	0.37	0.17	0.44	0.44	**0.80**	**0.79**
V236	0.42	0.28	0.25	0.36	0.17	**0.76**

1 and video 4 should be the lowest, whereas video 1 and video 2 are the most similar. Table 6.6 lists the similarity scores and shows that the proposed method is effective to measure the variable-length similarity comparisons between two video sequences.

Table 6.6: Video similarity scores for near-duplicate detection

	Video 2	Video 3	Video 4
Video 1	0.78	0.68	0.41

6.2 VTDF-based Video Event Detection

6.2.1 Model Description

Given a video sequence, the main research problem in video event detection is to find a solution to classify it into one predefined video event.

Considering X is one video sequence with N frames and E is a class sets including all predefined M events,

$$X = \{x_1, x_2, ..., x_N\}, \tag{6.13}$$

$$E = \{E_1, E_2, ..., E_M\}, \tag{6.14}$$

where x_i is the feature vector of the i-th frame ($1 \leq i \leq N$). E_j is the feature vector to model the j-th event ($1 \leq j \leq M$).

Let $C(X)$ be the event index of X, the optimized solution to classify X can be obtained by,

$$C(X)_{opt} = \arg\max_j \left(f(X, E_j)\right), \tag{6.15}$$

where $f(.)$ is a predefined decision function to measure the semantic similarity between X and E_j.

To solve the above optimization problem, a new event detection method using the VTDF and temporal quantization is proposed. The new method includes three components: the VTDF, the VTDF-based temporal quantization, and the correlation-based similarity measure. The VTDF is used to explore the inter-frame activities of video data in the time domain. By using VTDF, one video sequence can be represented as an one dimensional time series signal. The VTDF-based temporal quantization is then employed to model the temporal characteristics of video with the optimal quanta and partition. After quantization, all video sequences are quantized into the same dimensional feature vectors. The correlation is calculated as the decision function to find the event index with the maximum correlation coefficient value. From the semantic point of view, two video sequences with the similar temporal rhythm of VTDF can be considered that they are semantically correlated.

6.2.2 Quantization and Similarity

The previous work on VTDF and temporal quantization are applied to model each video sequence. After that, a VTDF-based feature vector can be generated to represent each video sequence.

Let V be a video sequence. $\forall 1 \leq i \leq L$, its VTDF-based feature vector, $F(V)$, can be represented as,

$$F(V) = [I(v_1)\ I(v_2)\ ...\ I(v_L)], \tag{6.16}$$

$$v_i = round(q_i), \tag{6.17}$$

where L is the dimension of VTDF-based feature vector.

Given two video sequences U and V, their similarity can be measured by their correlation coefficient as,

$$\rho_{U,V} = \left| \frac{cov(F(U), F(V))}{\sigma_{F(U)}\sigma_{F(V)}} \right|, \tag{6.18}$$

where σ is standard deviation and $cov(.)$ means covariance operator.

The reason to apply correlation coefficient to measure the semantic similarity between two video sequences is that two sequences with a high correlation coefficient mean that they are semantic correlated.

Accordingly, the decision function in the model description section can be defined as,

$$C(X)_{opt} = \arg\max_j \left(\rho_{F(X),F(Y_j)})\right), \tag{6.19}$$

where $1 \leq j \leq M$. And M is the number of predefined events.

The whole event detection procedure in the proposed method has three steps.

First, all video sequences are modeled by the VTDF in a compact way. Second, temporal quantization is employed to reduce the computational dimension and make all sequences have the same dimensional compact VTDF-based feature vector. Third, correlation is applied to find the predefined event sequence with the maximum correlation coefficient value. Figure 6.1 shows the video event detection procedure.

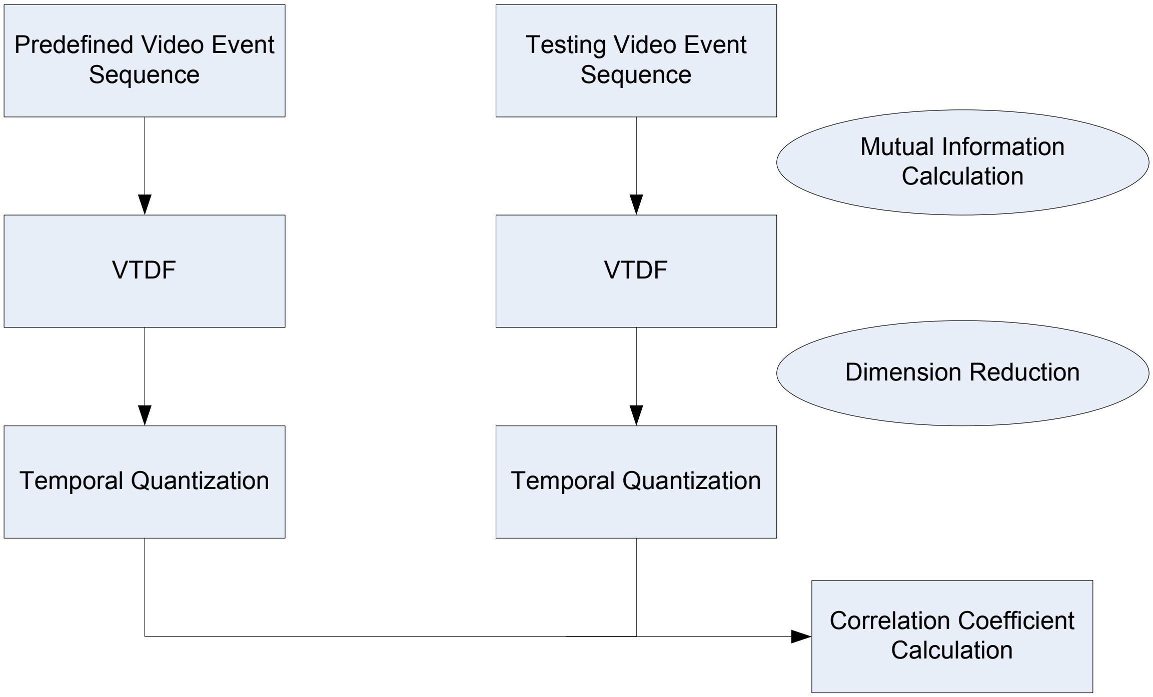
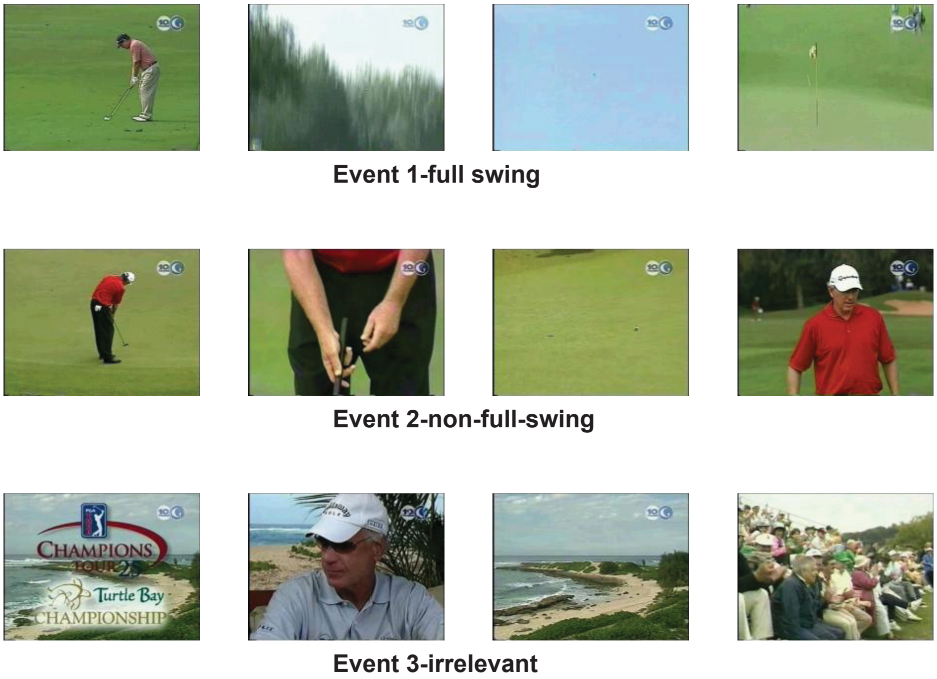

Predefined Video Event Sequence
Testing Video Event Sequence
Mutual Information Calculation
VTDF
VTDF
Dimension Reduction
Temporal Quantization
Temporal Quantization
Correlation Coefficient Calculation

Figure 6.1: Video event detection architecture

6.2.3 Experimental Results

To show the effectiveness of the proposed method, we use the same golf video in the last section to test the detection performance. Similarly, the events in golf video are predefined into three classes ($M = 3$), "full swing", "non-full-swing", and "irrelevant". Figure 6.2 shows the examples of three events. In Figure 6.2, we sequentially lay out four video frames per event.

Table 6.2 shows the ground truth of the test video sequences for golf video event



Event 1-full swing

Event 2-non-full-swing

Event 3-irrelevant

Figure 6.2: Three events in golf sports

detection. Note that the golf video has been pre-edited to remove all advertisements for better evaluation. And all video sequences are manually annotated and classified into one of three predefined events as the ground truth. There are 209 event sequences in total.

One video sequence from each event (three sequences in total) is preselected as the pattern/template to evaluate the detection performance. Given a video sequence for testing (as ground truth, it belongs to the event 1, “full swing”), the original VTDF patterns, VTDF-based features after quantization for three predefined event sequences and the testing sequence are shown in Figure 6.3. Figure 6.3 visually demonstrates that two VTDF-based video sequences with the similar temporal rhythm of the VTDF are semantically correlated.

Table 6.7 gives the event detection results of the golf video. Total 159 events are detected successfully. Table 6.8 gives the whole confusion matrix on golf video event detection.

Given a event E_i, its detection rate $D(E_i)$ can be calculated as,

Table 6.7: Golf video event detection results

	Detected event numbers
Event 1-full swing	26
Event 2-non-full-swing	110
Event 3-irrelevant	23
Total detected events	159

Table 6.8: Confusion matrix on golf video event detection

	Event 1	**Event 2**	**Event 3**
Event 1	26	21	3
Event 2	7	110	7
Event 3	2	10	23

$$D(E_i) = \frac{N_{(detected)}(E_i)}{N_{(total)}(E_i)}, \tag{6.20}$$

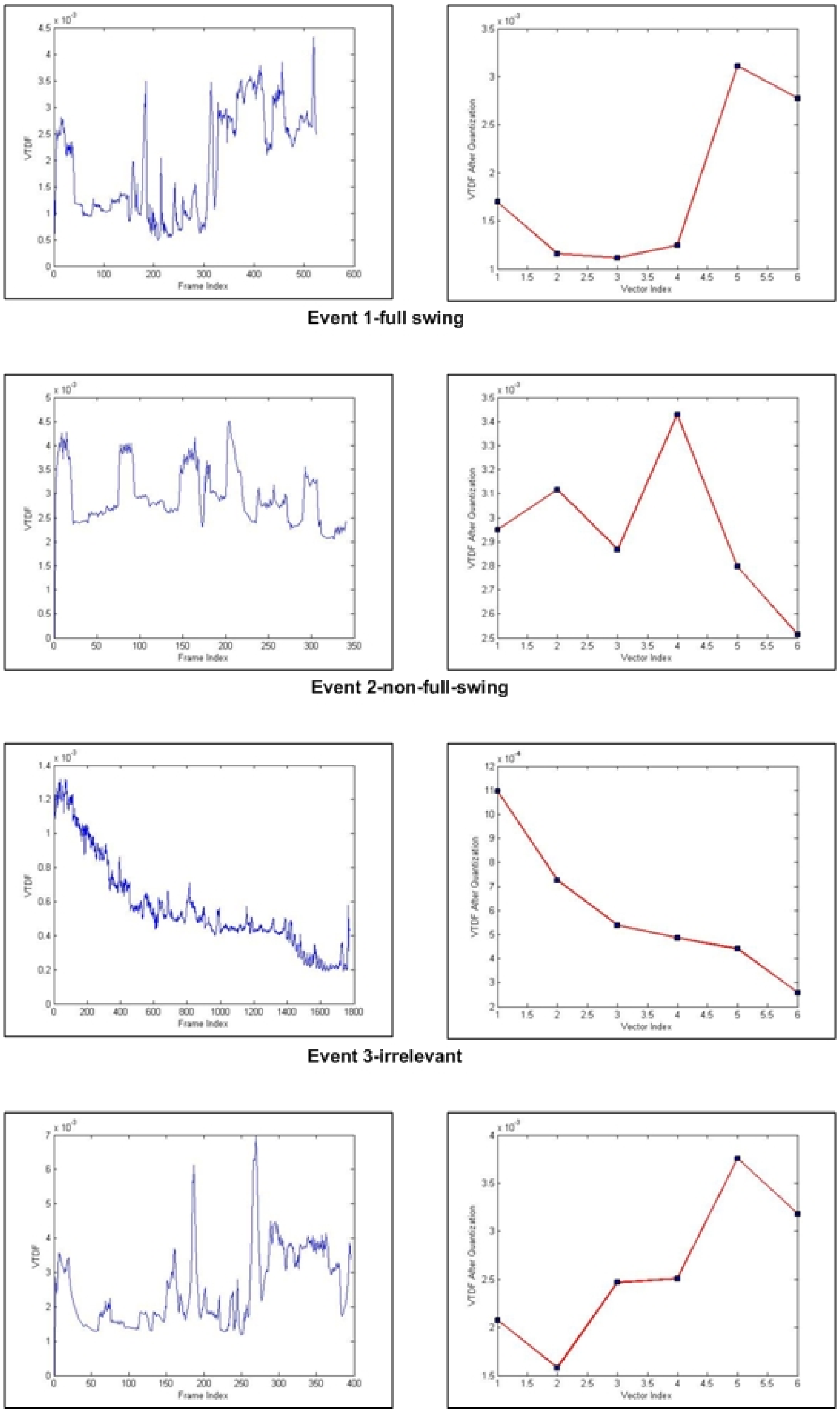
where $N_{(detected)}(E_i)$ is the number of correctly detected events, and $N_{(total)}(E_i)$ is the number of total events.

According to the above formula, the detection rate in golf video for “full swing”, “non-full-swing”, and “irrelevant” is 74.28%, 78.01%, and 69.69%, respectively. The total detection rate in average is 76.07%.

We refer to existing methods [90] [99] for comparisons. An overall classification accuracy of 83.5% for “play” and “break” is obtained in [90]. In [99], a 70.79% rate in golf sports event detection is reported. The discussions can be addressed in two aspects. First, the features used in [90] mainly focus on soccer domain, and thus lacking of general applications. Whereas the proposed detection method makes use of color features and employs inter-frame dependency to VTDF-based video data modeling to relieve some domain knowledge to a certain extent. To demonstrate that it is generic-oriented and independent of domain knowledge, we also test the detection performance on hockey and bowling sports videos. A comparable detection rate (both around 70%) is obtained. The reason to test on sports video is that it has good structure for event predefinition. Second, in [99], the parameter set with five parameters makes it complex in model learning. Whereas the proposed VTDF-based temporal quantization method combines with the correlation-based decision function overcomes the constraint in time duration and provides a generic-oriented solution on semantic similarity measure between two video sequences. Compared with the method in [99], the proposed method does not need model learning and parameter estimation in general. In addition, for the same kind of golf sports test video, a higher detection rate is obtained by our method. We test the method in [99] on the same golf video and a 77.42% detection rate is obtained.

6.3 Chapter Summary

In this Chapter, the proposed theoretical framework using the VTDF and statistical models are extended to solve the problems in video similarity measure and video event detection. In particular, to measure the similarity between two video sequences, each sequence is modeled using several temporal segments. After modeling each segment using VTDF-based features, correlation is calculated to measure the similarity between two segments. Dynamic programming is employed to combine the similarity scores of two segments and measure the similarity of two videos. Video event detection can be considered as a problem of video similarity measure. After applying the VTDF-based temporal quantization, all video sequences are modeled by equal sized VTDF-based features. Correlation is applied to measure the similarity of two videos. For a video for classification, the predefined video event that has the highest similarity score can be considered as the detection result. A golf video is used to test the performance of the proposed methods. Experimental results show that the proposed normalized similarity function can measure the similarity of two videos effectively. As a generic video event detection method, the proposed method does not need model training and parameter learning to detect video events effectively and efficiently.



Event 1-full swing

Event 2-non-full-swing

Event 3-irrelevant

Testing event-full swing

Figure 6.3: VTDF patterns for three events and test sequence

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis addresses video content analysis problems based on the spatial and temporal dynamics of video data. Multimedia signal processing research is experiencing a rapid surge because of the advance of new consumer electrical devices and the Internet. However, most technologies on video content analysis have limited performance, this is due to that fact that they are only using a few low-level features such as color, texture, shape, and motion. The reason is that there is a semantic gap between high-level meanings and low-level features. The direct solution of this problem is to understand the multimedia content and bridge the semantic gap.

In this thesis, two kinds of video data modeling tools are presented to explore the semantic information of video data, the ICA-based feature extraction and the VTDF. Based on the low-level feature using color histogram, the ICA features are effective to explore the trajectory characteristics of the whole video stream in the compact 2D ICA subspace. For a video, its time density carries a lot of information and makes it unique in general. The proposed VTDF is to explore the time density changes in the time domain by modeling the inter-frame dependency.

Based on the two video data modeling tools, a theoretical framework using the VTDF and statistical models is proposed to analyze the video content. Several new solutions are proposed to solve the problems in video content analysis based on specific applications: video summary, video similarity measure, and video event detection.

Two new solutions on static video summary are proposed. Hierarchical key frame tree enables a dynamic key frames layout. Statistical model vector quantization can generate a compact video summary.

The theoretical framework using the VTDF and statistical models is to solve the problems in dynamic video summary, video similarity measure, and video event detection. For dynamic video summary, based on the inter-frame dependency calculation, the VTDF is to explore the temporal dynamics of video data in an effective and efficient way. The VTDF is to give a weight to each video frame to evaluate its importance for sampling in dynamic video summary. Compared with a video frame with a small value of VTDF, the video frame with a large value of VTDF is more possible to be sampled. Inspired by the traditional vector quantization concept, a new VTDF-based temporal quantization is developed to segment the whole video stream in the time domain. As a result, an optimal video summary with the minimum distortion can be generated for rapid video navigation.

After combining with statistical models, such as GM model and ICAM model, a compact video summary can be created as video thumbnails. Both rapid video navigation and video thumbnail extraction are very useful for the users to manage their video collections. Different from the methods in literature to address the physical structure analysis (e.g., shot boundary detection, scene change), we do content-based video frame sampling based on the time constraint. All sampled frames will be fed into the built video players for fast-forward playback.

Considering two videos with the similar VTDF rhythms can be considered that they are semantically correlated, the proposed framework is also extended to measure video similarity and detect video events. Experimental results show that the framework can solve the problems in video similarity measure and video event detection effectively and efficiently.

Given the richness of literature in effective and efficient information coding and representation using PDF, we expect that VTDF will serve as a foundation of video content representation and more video content analysis methods can be developed based on the VTDF framework.

7.2 Future Work

According to the progress of this thesis on video content analysis, there are still several problems that need to be addressed in the future.

- To combine multiple features of videos for analysis: Multimedia is a combination of multiple form of content. Multi-modality needs to be investigated thoroughly for representing video content. It is also very important to understand the role of each feature and modality for combining them naturally in the video content analysis framework.
- To justify the performance of the proposed methods: Considering complex characteristics of video data in nature, there is no a standard way to justify and

evaluate the performance of methods. We try to solve this problem in several ways. For example, three kinds of criteria are applied to evaluate the video summarization results. First, we lay out the sample frames sequentially and hierarchically and compare with others from visual point of view. Second, we feed sample frames into the video players to do fast-forward playback. Third, we combine the R-D theory to calculate the value of TMSE to evaluate the quantization distortion. In the future, the research direction of video content analysis evaluation should be connected with human visual system together. In other words, the objective of video content analysis is to towards extracting semantically meaningful information to match human visual system effectively and efficiently. We plan to arrange a group of users to do annotations on the quality of video summary to build a ground truth for future evaluations and comparisons.

- To be aware of existing approaches in this area: We can gauge our research work with the state-of-art methods. Meanwhile, some of the applications we would like to enable include semantic search and automatic extraction of “interesting” or “representative” segments from video.

- To connect with practical applications: We can work with our research partners to apply and extend the proposed methods to the real applications on both software and hardware. Doing some kind of “semantic” key frame extraction could be the first step. It is interesting to see how we can summarize or compact these consumer type video clips into something that are concise and meaningful to the users. We can also explore the techniques from other fields and see if they can be effectively applied here for this research problem.

- To build a standard video database: The video database used in this thesis is limited. We need to test much more videos with different formats, categories, and types to make sure it is valuable to deploy the proposed methods in practice.

Appendix: Video Sources

The videos tested in this thesis are from different sources.

- Two videos from Microsoft are standard videos (aquarium and downtown).
- Three videos are recorded from TV. One video is about rocket, which is tested in hierarchical key frame tree. One video is about golf sports, which is parsed into 209 video sequences in video similarity measure and event detection. One is the movie video, Terminator 2.
- Five videos from a research partner are high definition videos. They are from different categories: music, sports, news, movie, and TV series.
- Three videos from Kodak Research Lab (research partner) are home videos, piano playing, fireworks, and graduation ceremony.
- One self-recorded video is the video with still background and four fade-in-fade-out transitions.
- Total thirty-five home videos are downloaded from Youtube from five different categories: sports, wedding, performance, graduation ceremony, and fireworks.

VITA

NAME:	Junfeng Jiang
PLACE OF BIRTH:	Daqing, CHINA
POST-SECONDARY	Harbin Institute of Technology, Harbin, CHINA
DEGREES:	BEng, MEng
HONORS AND AWARDS:	Ryerson Graduate Award, 2006-2010
	Ryerson Access to Opportunity Program, 2006-2010
	Ryerson Research Stipend, 2006-2011
	Ryerson Graduate Research Excellence Award, 2010

PUBLICATIONS

1. J. Jiang, X.-P. Zhang, "Trends and opportunities in consumer video content navigation and analysis", invited position paper, accepted to appear in Proc. of International Conference on Computing, Networking and Communications (ICNC) 2012, Jan.30-Feb.2, Maui, Hawaii, USA.

2. J. Jiang, X.-P. Zhang, "A smart video player with content-based fast-forward playback", accepted to appear in Proc. of ACM Multimedia 2011, Nov.28-Dec.1, Scottsdale, Arizona, USA.

3. J. Jiang, X.-P. Zhang, Alexander C. Loui, "A content-based video fast-forward playback method using video time density function and rate distortion theory", in Proc. of IEEE International Conference on Multimedia and Expo (ICME) 2011, July 11-15, Barcelona, Spain.

4. J. Jiang, X.-P. Zhang, “A novel vector quantization-based video summarization method using independent component analysis mixture model”, in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2011, May 22-27, Prague, Czech Republic.

5. J. Jiang, X.-P. Zhang, Alexander C. Loui, “A new video similarity measure model based on video time density function and dynamic programming”, in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2011, May 22-27, Prague, Czech Republic.

6. J. Jiang, X.-P. Zhang, “Video thumbnail extraction using video time density function and independent component analysis mixture model”, in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2011, May 22-27, Prague, Czech Republic.

7. J. Jiang, X.-P. Zhang, “A content-based rapid video playback method using motion-based video time density function and temporal quantization,” in Proc. of International Workshop on Social, Adaptive and Personalized Multimedia Interaction and Access (SAPMIA 2010) in conjunction with ACM Multimedia 2010, October 25-29, Florence, Italy.

8. J. Jiang, X.-P. Zhang, “A novel video thumbnail extraction method using spatiotemporal vector quantization,” in Proc. of the 3rd International Workshop on Automated Information Extraction in Media Production (AIEMPro’10) in conjunction with ACM Multimedia 2010, October 25-29, Florence, Italy.

9. J. Jiang, X.-P. Zhang, “Gaussian mixture vector quantization-based video summarization using independent component analysis,” in Proc. of IEEE International Workshop on Multimedia Signal Processing (MMSP), October 4-6, 2010, Saint-Malo, France.

10. J. Jiang, X.-P. Zhang, “A new hierarchical key frame tree-based video representation method using independent component analysis,” in Proc. of the 6th International Conference on Intelligent Computing (ICIC 2010), August 18-21, 2010, Changsha, China.

11. J. Jiang, X.-P. Zhang, “A new player-enabled rapid video navigation method using temporal quantization and repeated weighted boosting search,” in Proc. of the 7th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision (POCV) in conjunction with IEEE CVPR 2010, June 13-18, 2010, San Francisco, USA.

12. Z. Wei, X.-P. Zhang, J. Jiang, Z. Zhao, “A video navigation model based on a genetic algorithm,” in Proc. of the 23rd IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), May 2-5, 2010, Calgary, Canada.

References

[1] Apple Quicktime Player, a video player from Apple Corporation Inc. http://www.apple.com/quicktime/, accessed 2011-08-10.

[2] BBC Rushes. http://www-nlpir.nist.gov/projects/tv7.acmmm/, accessed 2011-09-08.

[3] Cyberlink PowerDVD, a video player from Cyberlink Corporation Inc. http://www.cyberlink.com/multi/products/main_1_ENU.html/, accessed 2011-08-10.

[4] Kinect for xbox 360. http://www.xbox.com/en-US/kinect/, accessed 2011-08-10.

[5] Microsoft Media player, a video player from Microsoft Corporation Inc. http://www.microsoft.com/windows/windowsmedia/default.mspx/, accessed 2011-08-10.

[6] Real Networks Realone Player. http://www.real.com/, accessed 2011-08-10.

[7] TRECVID. http://www-nlpir.nist.gov/projects/trecvid/trecvid.tools/, accessed 2011-08-10.

[8] B. Adams, S. Greenhill, and S. Venkatesh. Temporal semantic compression for video browsing. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, 2008.

[9] M.-W. Bang, S.-J. Kang, J. H. Park, S.-Y. Park, and W. H. Cho. Shot boundary detection of video sequence using hierarchical hidden Markov models. In *Proceedings of International Conference Computers and Their Applications*, pages 471–474, 2002.

[10] A. J. Bell and T. J. Sejnowski. The independent component of natural scenes are edge filters. *Vision Research*, pages 3327–3328, 1997.

[11] S. Benini, A. Bianchetti, R. Lenoardi, and P. Migliorati. Hierarchical summarization of videos by tree-structured vector quantization. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2006.

[12] S. Benini, A. Bianchetti, R. Lenoardi, and P. Migliorati. Hierarchical video summaries by dendrogram cluster analysis. In *Proceedings of the 14th European Signal Processing Conference (EUSIPCO)*, 2006.

[13] S. Benini, P. Migliorati, and R. Leonardi. Hidden Markov models for video skim generation. In *Proceedings of the 8th International Workshop on Image Analysis for Multimedia Interactive Services*, June 2007.

[14] J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. In *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, pages 170–179, 1996.

[15] J. S. Boreczky and L. D. Wilcox. A hidden Markov model framework for video segmentation using audio and image features. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3741–3744, 1998.

[16] T. Butz and J.-P. Thiran. Shot boundary detection with mutual information. In *Proceedings of International Conference Image Processing (ICIP)*, pages 219–227, 2001.

[17] Z. Cernekova, C. Nikou, and I. Pitas. Entropy metrics used for video summarization. In *Proceedings of the 18th Spring Conference on Computer Graphics*, 2002.

[18] Z. Cernekova, I. Pitas, and C. Nikou. Information theory-based shot cut/fade detection and video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 82–91, 2006.

[19] L. Chang, Y. Yang, and X.-S. Hua. Smart video player. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2008.

[20] B.-W. Chen, J.-C. Wang, and J.-F. Wang. A novel video summarization based on mining the story-structure and semantic relations among concept entities. *IEEE Transactions on Multimedia*, 11(2):295–312, 2009.

[21] S. Chen, X. Wang, and C. J. Harris. Repeating weighted boosting search for optimization in signal processing application. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(4):682–693, 2005.

[22] K.-Y. Cheng, S.-J. Luo, B.-Y. Chen, and H.-H. Chu. Smartplayer: user-centric video fast-forwarding. In *Proceedings of 27th International Conference on Human Factors in Computing Systems*, 2009.

[23] S.-C. S. Cheung and A. Zakhor. Fast similarity search and clustering of video sequences on the world-wide-web. *IEEE Transactions on Multimedia*, 7(3):524–537, 2005.

[24] S.-S. Cheung and A. Zakhor. Efficient video similarity measurement with video signature. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 59–74, January 2003.

[25] P. Comon. Independent component analysis - a new concept? *Signal Processing*, 36:287–314, 1994.

[26] C. Cotsaces, N. Nikolaidis, and I. Pitas. Video shot boundary detection and condensed representation: a review. *IEEE Signal Processing Magazine*, 23(2):28–37, March 2006.

[27] L. Deng, L.-Z. Jin, and S.-M. Fei. A novel method for video shot similarity measures. In *Proceedings of the 5th International Conference on Machine Learning and Cybernetics*, August 2006.

[28] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor. Application of video-content analysis and retrieval. *IEEE Multimedia*, 9(3):42–55, July 2002.

[29] A. Divakaran, R. Radhakrishnan, and K. A. Peker. Motion activity-based extraction of key-frames from video shots. In *Proceedings of International Conference Image Processing (ICIP)*, pages 932–935, 2002.

[30] C. Dorai, B. T. Truong, and S. Venkatesh. New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *Proceedings of ACM International Conference on Multimedia*, pages 219–227, 2000.

[31] A. D. Doulamis and N. D. Doulamis. Optimal content-based video decomposition for interactive video navigation. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(6):757–775, 2004.

[32] M. S. Drew, J. Wei, and Z.-N. Li. Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalize images. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 1998.

[33] F. Dufaux. Key frame selection to represent a video. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2000.

[34] A. Divakaran et al. *Video mining*. Kluwer Academic Publishers, USA, 2006.

[35] R. Laganiere et al. Video summarization from spatio-temporal features. In *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, 2008.

[36] R. Glasberg, S. Schmiedeke, M. Mocigemba, and T. Sikora. New real-time approaches for video-genre-classification using high-level descriptors and a set of classifiers. In *Proceedings of International Conference on Semantic Computing)*, 2008.

[37] Y. Gong and X. Liu. Generating optimal video summaries. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2000.

[38] Y. Gong and W. Xu. *Machine learning for multimedia content analysis.* Springer-Verlag, USA, 2007.

[39] R. M. Gray. Gauss mixture vector quantization. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1769–1772, 2001.

[40] P. O. Gresle and T. S. Huang. Gisting of video documents: a key frames selection algorithm using relative activity measure. In *Proceedings of the 2nd International Conference on Visual Information Systems*, 1997.

[41] A. Hanjalic. Shot-boundary detection: unraveld and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):90–104, 2002.

[42] A. Hanjalic and H.-J. Zhang. Optimal shot boundary detection based on robust statistical models. In *Proceedings of International Conference on Multimedia Computing and Systems*, pages 710–714, 1999.

[43] S. Hasebea and M. S. Mustafa. Constructing storyboards based on hierarchical clustering analysis. In *Proceedings of Visual Communications and Image Processing, SPIE*, 2005.

[44] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Press, USA, 2001.

[45] T. S. Huang, Y. Zhang, Y. Rui, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 1998.

[46] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[47] A. K. Jain, A. Vailaya, and X. Wei. Query by video clip. *Multimedia System*, pages 369–384, 1999.

[48] S. Jeong, C. S. Won, and R. M. Gray. Histogram-based image retrieval using gauss mixture vector quantization. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2003.

[49] P. Jiang and X.-L. Qin. Keyframe-based video summary using visual attention clues. *IEEE Multimedia*, 17(2):64–73, 2009.

[50] A. Kankanhalli, H.-J. Zhang, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia System*, 1:10–28, January 1993.

[51] K. Kashino, T. Kurozumi, and H. Murase. A quick search method for audio and video signals based on histogram pruning. *IEEE Transactions on Multimedia*, 5(3):348–357, 2003.

[52] R. Lancini, F. Mapelli, and A. Mucedero. Automatic identification of compressed video. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2004.

[53] T. Lee, M. S. Lewicki, and T. J. Sejnowski. Ica mixture models for unsupervised classification and automatic context switching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1078–1089, 2000.

[54] T.-W. Lee and M. S. Lewicki. Unsupervised image classification, segmentation, and enhancement using ica mixture models. *IEEE Transactions on Image Processing*, 11(3):270–279, 2002.

[55] Y. Li and C.-C. J. Kuo. A robust video scene extraction approach to movie content abstraction. *Imagining Systems and Technology*, 13:236–244, May 2004.

[56] Y. Li and B. Merialdo. Vert: automatic evaluation of video summaries. In *Proceedings of ACM Multimedia*, 2010.

[57] Z. Li, A. K. Katsaggelos, and B. Gandhi. Temporal Rate-Distortion based optimal video summary generation. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2003.

[58] Z. Li, A. K. Katsaggelos, and B. Gandhi. Optimal video summary generation and coding. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2005.

[59] Z. Li, G. M. Schuster, A. K. Katsaggelos, and B. Gandhi. Rate-Distortion optimal video summarization: a dynamic programming solution. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.

[60] Z. Li, G. M. Schuster, A. K. Katsaggelos, and B. Gandhi. Rate-Distortion optimal video summary generation. *IEEE Transactions on Image Processing*, 14(10):1550–1560, 2005.

[61] Z.-N. Li and J. Wei. Spatio-temporal joint probability images for video segmentation. In *Proceedings of International Conference on Image Processing (ICIP)*, 2000.

[62] T. Lin and H.-J. Zhang. Automatic video scene extraction by shot grouping. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR)*, pages 39–40, 2000.

[63] 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*, 13(10):1006–1013, 2003.

[64] T. Liu, X. Zhang, J. Feng, and K.-T. Lo. Shot reconstruction degree: a novel criterion for key frame selection. *Pattern Recognition Letters*, pages 1451–1457, 2004.

[65] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[66] J. Luo, C. Papin, and K. Costello. Towards extracting semantically meaningful key frames from personal video clips: from humans to computers. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2):289–301, 2009.

[67] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang. A generic framework of user attention model and its application in video summarization. *IEEE Transactions on Multimedia*, 7(5), 2005.

[68] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *Proceedings of the 10th ACM Multimedia*, pages 533–542. ACM, December 2002.

[69] C. D. Manning, P. Raghavan, and H. Schutze. *An Introduction to Information Retrieval*. Cambridge University Press, 2008.

[70] M. R. Naphade and T. S. Huang. Extracting semantics from audiovisual content: the final frontier in multimedia retrieval. *IEEE Transactions on Neural Networks*, pages 793–810, 2002.

[71] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang. Automatic video summarization by graph modeling. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2003.

[72] P. V. Pahalawatta, Z. Li, F. Zhai, and A. K. Katsaggelos. Rate-Distortion optimization for Internet video summarization and transmission. In *Proceedings of IEEE the 7th Workshop on Multimedia Signal Processing*, 2005.

[73] A. Pardo. Probabilistic shot boundary detection using inter frame histogram differences. In *Proceedings of Iberoamerican Congress Pattern Recognition*, pages 726–732, 2006.

[74] H. Permuter, J. Francos, and I. H. Jermyn. A study of Gaussian mixture models of colour and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.

[75] X. Shi and D. Schonfeld. Video classification and mining based on statistical methods for cross-correlation analysis. In *Proceedings of 14th IEEE Workshop on Statistical Single Processing (SSP)*, 2007.

[76] S. Shipman, A. Divakaran, and M. Flynn. A highlight scene detection and video summarization system using audio feature for a personal video recorder. *IEEE Transactions on Consumer Electronics*, 51(1):112–116, 2007.

[77] J. R. Smith. Videozoom spatio-temporal video browser. *IEEE Transactions on Multimedia*, 1(2):157–171, 1999.

[78] M. A. Smith and T. Kandade. Video skimming and characterization through the combination of image and language understanding. In *Proceedings of International Workshop on Content-Based Access Image Video Data Base*, pages 61–67, 1998.

[79] S. W. Smoliar, H. Zhang, C.Y. Low, and D. Zhong. Video parsing, retrieval and browsing: an integrated and content-based solution. In *Proceedings of ACM Multimedia*, 1995.

[80] X. Song. Selecting salient frames for spatiotemporal video modeling and segmentation. *IEEE Transactions on Image Processing*, 16(12):3035–3046, 2007.

[81] F. E. Streib and M. Dehmer. *Information theory and statistical learning.* Springer, USA, 2009.

[82] S. Theodoridis and K. Koutroumbas. *Pattern Recognition (2nd edition).* Academic Press, USA, 2003.

[83] B. T. Truong and S. Venkatesh. Video abstraction: a systematic review and classification. *ACM Transactions on Multimedia Computing, Communications and Applications*, 3(1), February 2007.

[84] S. Uchihashi and J. Foote. Summarizing video using a shot importance measure and a frame-packing algorithm. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[85] V. Valdes and J. M. Martinez. Binary tree based on-line video summarization. In *Proceedings of the 2nd ACM TRECVID Video Summarization Workshop*, 2008.

[86] X. Wang and X.-P. Zhang. Ica mixture hidden conditional random field model for sports event classification. In *Proceedings of IEEE Workshop on Video Oriented Object and Event Classification in Conjunction with ICCV*, 2009.

[87] X. Wang, X.-P. Zhang, I. Clarke, and Y. Yakubovich. A new Gaussian mixture conditional random field model for indoor image labeling. In *Proceedings of ACM Multimedia Workshop on Interactive Multimedia for Consumer Electronics*, 2009.

[88] I. H. Witten, E. Frank, and M. A. Hall. *Data mining: practical machine learning tools and techniques.* Morgan Kaufmann, USA, 2011.

[89] W. Wolf. key frame selection by motion analysis. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1996.

[90] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun. Structure analysis of soccer video with hidden Markov models. *Pattern Recognition Letters*, pages 767–775, May 2004.

[91] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang. Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003.

[92] Z. Xiong, R. Radhakrishnan, A. Divakaran, Y. Rui, and T. S. Huang. *A unified framework for video summarization, browsing and retrieval.* Elsevier Academic Press, USA, 2006.

[93] M. M. Yeung, B. L. Yeo, and B. liu. Extracting story units from long programs for video browsing and navigation. In *Proceedings of IEEE Conference on Multimedia Computing and Systems*, 1996.

[94] H. Yi, D. Rajan, and L.-T. Chia. A new motion histogram to index motion content in video segments. *Pattern Recognition Letters*, pages 1221–1231, 2005.

[95] L. Zhao, W. Qi, S. Z. Li, S.-Q. Yang, and H.-J. Zhang. Key-frame extraction based on improved nearest feature line (NFL) classification algorithm. *Computers*, 23(12), 2000.

[96] J. Zhou and X.-P. Zhang. A web-enabled video indexing system. In *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2004.

[97] J. Zhou and X.-P. Zhang. Automatic identification of digital video based on shot-level sequence matching. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, November 2005.

[98] J. Zhou and X.-P. Zhang. Video shot boundary detection using independent component analysis. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2005.

[99] J. Zhou and X.-P. Zhang. An ica mixture hidden Markov model for video content analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 18:1576–1586, 2008.

[100] S. K. Zhou and R. Chellappa. From sample similarity to ensemble similarity: probabilistic distance measure in reproducing kernel Hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 917–929, 2006.

[101] X. Zhu, J. Fan, A. K. Elmagarmid, and X. Wu. Hierarchical video summarization and content description joint semantic and visual similarity. *ACM Multimedia System*, 9(1), 2003.