1-1-2005

Video content analysis based on statistical modeling

Jian Zhou
Ryerson University

VIDEO CONTENT ANALYSIS BASED ON STATISTICAL MODELING

by

JIAN ZHOU
B.Eng.
Tianjin, P.R. China, 1997

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2005

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: ___ _

BORROWER'S PAGE

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

Name	Signature	Address	Date

Video Content Analysis Based on Statistical Modeling

Master of Applied Science 2005

JIAN ZHOU

Electrical and Computer Engineering

Ryerson University

Abstract

This thesis is aimed at finding solutions and statistical modeling techniques to analyze the video content in a way such that intelligent and efficient interaction with video is possible. In our work, we investigate several fundamental tasks for content analysis of video. Specifically, we propose an online video parsing algorithm using basic statistical measures and an off-line solution using Independent Component Analysis (ICA). A spatiotemporal video similarity model based on dynamic programming is developed. For video object segmentation and tracking, we develop a new method based on probabilistic fuzzy c-means and Gibbs random fields. Theoretically, we develop a generic framework for sequential data analysis. The new framework integrates both Hidden Markov Model and ICA mixture model. The re-estimation formulas for model parameter learning are also derived. As a case study, the new model is applied to golf video for semantic event detection and recognition.

Acknowledgments

I would like to thank all the people who have supported me during the completion of this thesis. Their help has made this thesis possible and an enjoyable experience for me. I am very grateful to my supervisor, Dr. Xiao-Ping Zhang, for his guidance, encouragement, and support. Besides providing constructive discussions, his kind help is crucial for me to build my technical writing skills and the right attitude. I will definitely benefit from those research skills and the knowledge he shared with me.

I would also like to thank every member of my thesis defense committee. I appreciate their time, efforts, and contributions to this work.

Special thanks go to my colleagues in Communications and Signal Processing Applications Laboratory (CASPAL). It has always been a pleasure to discuss technical issues and exchange ideas with them.

I am also very grateful to the department of Electrical and Computer Engineering at Ryerson University for the financial support and the comfortable research environment.

Lastly, I would like to show my gratitude to my family for their caring and support. I dedicate this thesis to my parents and my wife Ying.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Objectives

With technology advances in digital TV, multimedia, and Internet, we have witnessed the amazing growth in the amount of digital image/audio/video data in recent years. As a key element of multimedia computing, digital video has been widely employed in many industries and in various systems. Nowadays, digital video content has been extensively used in entertaining, news, advertising, sport, education, and publishing. The interaction with video has become an important part of our lives.

In the professional TV and movie industry, the migration to digital content has been driven by all kinds of low cost hardware (such as compact disc, Internet, set-top-box, digital TV) and the software part (such as the MPEG video standards [1], the video content editing, authoring and pre-mastering software in digital domain). For example, the success of DVD movies can be considered as a technology that combines both the hardware and the software. In the movie industry, in order to make a DVD movie, many detailed procedures and techniques have been developed to do the media authoring, editing, pre-mastering and manufacturing. However, most of the work, especially the processing of the digital video data, is done manually. The work is boring and tedious, and it is easy to make mistakes. Organizing and processing digital video manually is very time-consuming and inefficient. Thus, it is necessary to develop new technologies and automated tools to model, manage, and process digital videos effectively and efficiently.

In the meantime, digital video production is no longer a privilege for the professionals in the movie industry. Thanks to the popularity of the Internet and powerful personal computers, it has never been easier for ordinary people to record, edit, delivery, and publish their own home-made digital videos. Thus, the past years have seen an explosion in the use of digital home videos. It is not uncommon that even a personal collection of home videos becomes difficult to manage in a short time. To avoid manual indexing and annotation, people hope video data can be structurally categorized, organized and indexed, such that information can be searched and retrieved like Internet search engines. For the general user, desired systems should return results that match the queries such as "find the video clips that were shot during the birthday parties for my son" or "within all of my 2002 FIFA World Cup video collections, return all the intervals where goals are scored from corner kicks". Such tasks are referred as Content Based Video Retrieval (CBVR) [2] [3] [4] [5] in academic research areas. A completely working CBVR search system is not practical at this moment since many challenging problems still remain open. Recently, industry giants such as Google, Microsoft, and Yahoo have launched their own video search engines to index video data that already exist online. However, the indexing is mainly based on filenames and possibly the textual information on web pages that contain the video links. During the writing of the document, even the most ambitious Google Video Search project only makes use of the closed caption transcripts of the TV programs. That is far away from the content based semantic video search. A truly content based video search and retrieval is still an active ongoing research area both in industry and in the academic community.

In some other application fields, such as medical applications, education applications, and surveillance systems, content based analysis of video is also important. For example, by analyzing the real-time video content recorded by highway monitoring cameras, it might be possible to automatically capture the accident scenes or report car breakdowns. Another example is the use of content analysis in surveillance systems. For building and banking surveillance systems, such analysis might allow a machine to automatically report suspicious behaviors without expensive human supervision.

Given the above motivations, this thesis is aimed at finding solutions and modeling techniques to segment, index, and analyze the content of video. We refer 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, filtering, and semantic understanding. Note that content analysis of video including video search and retrieval is not a simple extension of textual search. Compared with textual data, multimedia data, particularly video data, are more random, less structured, and high dimensional. They are generally an integration of multiple modalities. Because of such characteristics, probabilistic approaches are often used to model multimedia signals. In this thesis, we try to view and tackle the problem of video content analysis from a statistical modeling point of view.

Our major concern is not to build a complete content based video search or retrieval system. Instead, we are interested in developing methods, algorithms, and a general framework to address the fundamental problems theoretically or to validate the applications in preliminary implementations. Our objectives for video content analysis include:

- Finding effective and efficient feature representations for video data.
- Modeling spatial and temporal characteristics of video data.
- Finding effective and efficient similarity/dissimilarity measures for video.
- Finding and modeling global and localized patterns and features.
- Achieving semantic video understanding.

1.2 Background and Challenges of Content Analysis of Video

Content based analysis of video is an essential part to achieve multimedia understanding, which, as an emerging interdisciplinary research area, is closely related to digital signal processing, artificial intelligence, data mining, pattern recognition, computer vision, and multimedia database technologies. The goal of video content analysis is to find ways to

better understand video data by combining multiple sources and fusing the information from different modalities. In our view, the most important research tasks include:

1. *Feature extraction.* Techniques and algorithms for signal processing, pattern recognition, and computer vision can be applied to video data to extract meaningful features and patterns. The term "feature" here in the video analysis domain are generally referred as the low level or intermediate level vision-related features. The success of further analysis highly depends on whether the features can effectively or meaningfully represent the original video data. A good feature can make the later tasks easier and help close the gap between low-level features and high-level semantics. Another important direction is the integration and fusion of multiple features. Better performance may be expected if different features can be combined and utilized together.

2. *Video parsing, indexing, and summarization.* Video parsing, indexing, and summarization are challenging and fundamental research tasks. The first step of video content analysis is to segment video into its constituent shots. This process, depending on the context, is generally called video temporal segmentation, video parsing, or video shot detection. The detection of a shot is equivalent to the detection of shot boundaries. The shot boundaries, also known as transitions, are editing effects introduced during video post-production. Thus, by recovering such editing effects, we hope some certain content-based semantics can be achieved since video producers often use transitions to separate two semantically different scenes. Most of the research activities on video indexing are concentrated on the detection of *abrupt transition* (cuts) and *gradual transition* (fades, dissolves, wipes), and the challenges mainly focus on the latter since it is generally very difficult to define and capture the discontinuity patterns of the gradual boundaries. The identification of video shots provides the building blocks for further processing and analysis. As shown in Figure 1.1, a video clip can be regarded as being structurally organized. Based on the video shots, *video indexing* can be performed manually or automatically. Video indexing is a process of attaching content based labels to video shots [6]. *Video summarization* is a further indexing step which provides

content summaries of video like the table of contents (ToC), and thus allows users to quickly browse and access the structured video content. Video summarization can be achieved by selecting one or a few still-frames from each shot that can best represent the content. The selected still-frames are called key-frames. Finding an efficient video indexing and summarization solution is an early but important stage of video content analysis. It not only gives users an important clue about the video content, but also explores the global temporal properties of video and provides the basic elements for further content analysis.

3. *Similarity/dissimilarity model.* The similarity models currently used are mainly based on the evaluation of distance-like functions in the feature space. Strictly speaking, they should be called *dissimilarity model* since a larger value usually implies more dissimilar. New metrics or distance-like functions need to be developed to make the measures consistent with human perception psychologically. Some researchers [7] distinguish between *pre-attentive* and *attentive* human similarity. Attentive similarity usually involves reasoning and previous knowledge. It depends on human interpretation. Pre-attentive similarity instead does not require any interpretation and is simply based on the perceived similarity between stimuli. Attentive similarity is mainly for domain-specific retrieval applications, such as face recognition, and mechanical parts recognition. Pre-attentive similarity is more important for content analysis where features such as color, texture, shape, motion, spatial relationship, and temporal relationship are used. In this thesis, the pre-attentive similarity model is one of our major concerns. The goal for a similarity/dissimilarity model is to define a proximity measure that conform to human similarity perception of sensorial stimuli [8]. For video similarity/dissimilarity model, little research has been done to fully make use of temporal dimension and/or intermediate to high level temporal information.

4. *Automatic extraction of semantics and localized features from video.* We regard the frame-based features as global features, which, as mentioned earlier, can be used for

video temporal segmentation, video indexing and summarization. The localized features are referred as the features associated with the meaningful components which are temporally and spatially partitioned within a shot. The research of detecting such meaningful components is known as *video object segmentation and tracking.* Compared with frame-level representation, the object-based representation can provide a finer resolution to support the realization of special tasks such as object-based video query and retrieval. The research of automatic extraction of localized features is significant because the semantics perceived by human beings are generally at the object-based level. The most important and challenging part for localized feature extraction is video object segmentation and tracking. The problem of video object segmentation and tracking is hard in that most of its sub-problems, such as spatial segmentation, motion segmentation, occlusion, video object formation, appearance/disappearance of video objects, and tracking of deformable objects are all nontrivial. Direct applications for the object-based content analysis include video coding, video editing and animation, object-based video indexing and retrieval, and high-level analysis of video contents.

5. *Video event recognition and detection.* Video event recognition and detection can be considered as the final step for content analysis of video. The recognition and detection of events from video is essentially the realization of video understanding. Such tasks generally require further exploration of temporal and spatial characteristics of video data. Desired event recognition and detection system should be able to extract the meaningful semantics and concepts using either global features or localized features. The video event detection algorithms can be categorized into two types: supervised and unsupervised. The supervised event detection is to detect occurrences or reoccurrence of activity based on pre-defined patterns. The research challenges of supervised detection reside within efficiently processing and presenting the video data, modeling the event, and incorporating domain knowledge. The unsupervised event detection is to process the video data and form clusters which are semantically meaningful. The unsupervised solutions can be considered as conceptual-level video clustering. We be-

lieve that the research on unsupervised solutions is still too early since even the major research problems for supervised event detection still remain open. For example, the effective modeling or description of semantic event, and the efficient representation of video data to reflect the spatial-temporal variations are still open problems. Potential applications of supervised event detection may include the analysis of surveillance video, traffic video, sports video, and home video. The applications of unsupervised event detection may include the video data mining and conceptual search.

6. *New theoretical algorithms and models for video signal processing.* Even though the research on multimedia is generally application-oriented, it does not necessarily mean the development of new mathematical algorithms and models is not a major concern. In fact, the multimedia research gives us a strong motivation and interest to derive new theoretical models. The development of new algorithms which can adapt to multimedia data is also an important challenging research direction for content analysis of video.

7. *Implementation issues.* Video data are generally large by nature. Thus, whenever an algorithm is proposed, the time and space issues have to be considered. Preliminary implementations are often prerequisite to verify the correctness of the proposed solutions. The integration and compatibility of industry standards are also important and necessary. For example, the video indexing and summarization can be represented by MPEG-7 meta-data descriptors, and the video object segmentation and tracking results can be integrated into video coding of MPEG-4 industry standard. Generally speaking, the research on multimedia mainly focuses on the provable applications, and the implementation is a non-separable part that can hardly be ignored.

The architecture of video content analysis is shown in Figure 1.2. The gray blocks shown in Figure 1.2 are the research areas covered by this thesis. User subjectivity and interaction modeling may include query by visual examples, query by sketch, and relevance feedback. Besides the research areas mentioned above, there are some other important research challenges and directions such as effective multimedia database model, multimedia

web search and retrieval, and interactive relevance feedback technologies for video search. However, these areas of research are not within the scope of the work in this thesis.

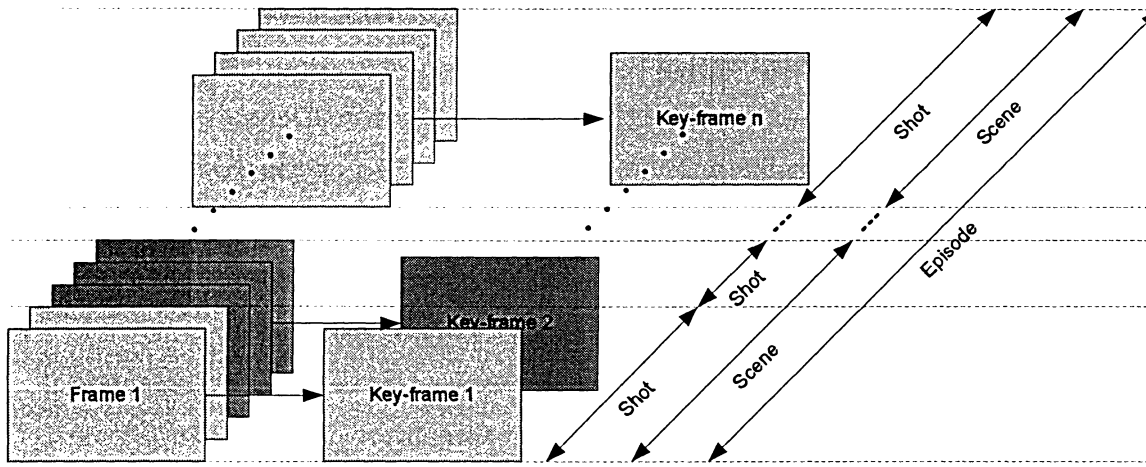


Figure 1.1: The layered structure of video.

1.3 Review of Previous Work

The representation of video content can be categorized into the *perceptual* level and the *conceptual* level [9]. The former focuses on the parsing and representing the video content by episodes, scenes, video shots, key frames, video objects, and other perception properties, such as color, texture, shape, and motion features. The latter, i.e. the conceptual level representation mainly focuses on the conceptual modeling and analysis of video to extract high-level semantic meanings. The conceptual level representation is generally built upon the perceptual level representations. A complete content analysis system is a combination of the two levels.

Early research and studies on content analysis of multimedia mainly focus on the processing of still images from perceptual level, and some also include very preliminary conceptual video processings. Several well-known content-based image and/or video management systems include QBIC [10], PhotoBook [11], Virage [12], VisualSEEk [13], MARS [14], and

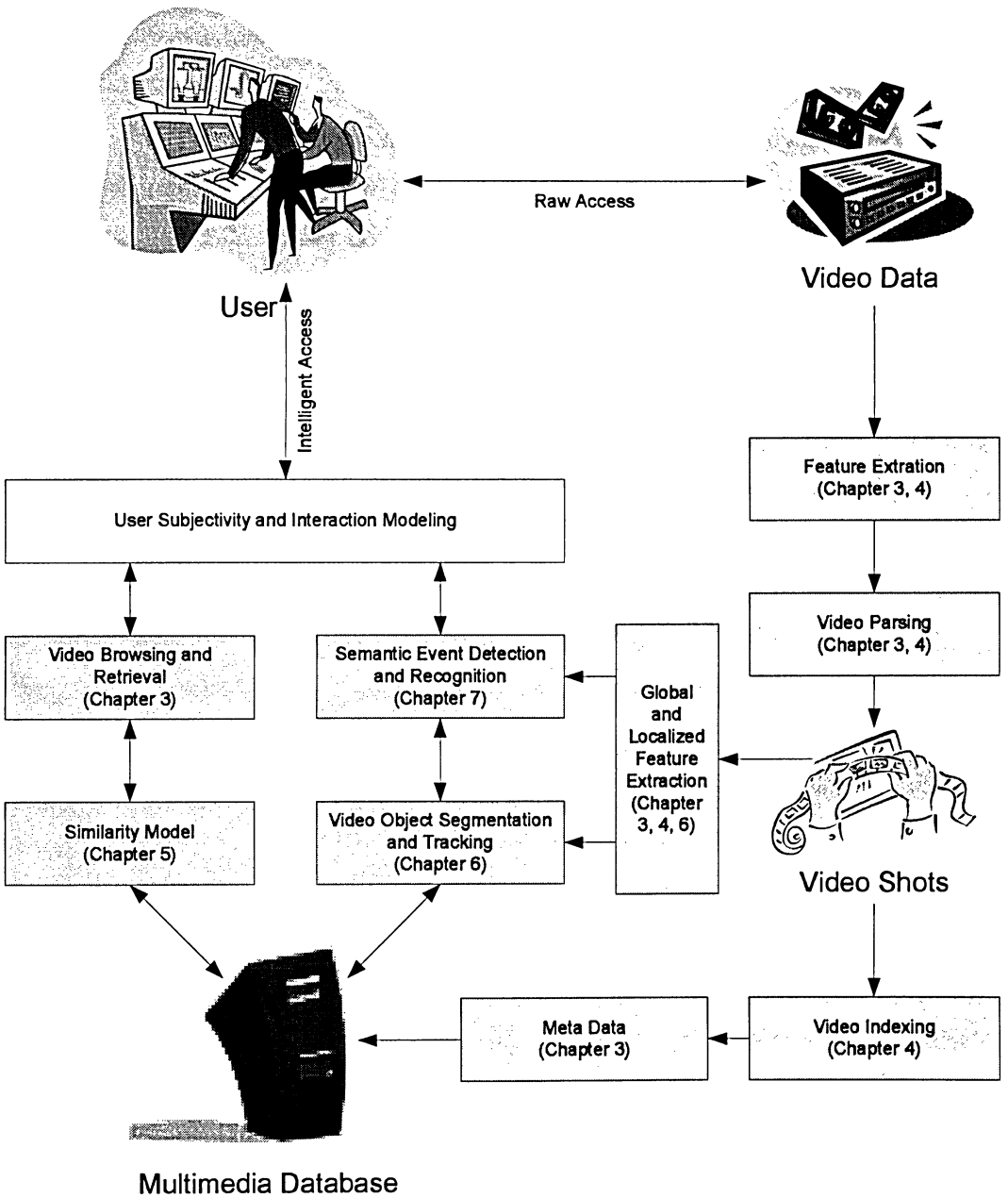


Figure 1.2: The architecture of video content analysis and thesis roadmap.

VideoQ [5]. Those systems generally support content-based indexing and queries based on low-level features. They automate the annotation of images and videos, and some also provide limited conceptual visual search.

Specifically, video parsing (video temporal segmentation) has been pioneered by several researchers. Zhang *et al.* [3] used a template matching technique to detect shot boundaries, while gradual transitions are detected by using two thresholds called twin-threshold [4]. In compressed domain, Yeo and Liu [15] used pixel difference and luminance histograms based on DC-images to detect video shots. Lay and Guan [16] chose energy histograms of the DCT coefficients for retrievals. Other approaches such as feature-based [17], model-based [18] and statistical [19] methods are also developed to detect shot boundaries.

Studies have also been conducted on video indexing and summarization. Yeung and Yeo [20] [21] proposed a time-constrained clustering to represent the video content for indexing and fast-browsing. Rui *et al.* [22] explored the video structures and generated a table of contents (ToC) of video for users to quick access and browse the video content. In [23], averaging and the intersection of histograms are used to cluster and identify key frames.

For video content analysis, most classical features in image analysis, such as color, texture, shape and spatial relationship, can be directly applied to video analysis. In addition, because of the temporal information contained in video, features such as global motions and localized motions for each video object can also be extracted and analyzed. Features can also be generated in transformed domain. For example, Sahouria and Zakhor [24] used Principal Component Analysis (PCA) to transform the video frames into a new feature space, and the content analysis related tasks, such as scene description and video sequence classification, were performed in the new feature space.

One important research task for video content analysis is to define similarity/dissimilarity models. Traditionally, the metric models, such as the Euclidean distance, the city-block distance and the Minkowsky distance, are chosen as the dissimilarity models in the feature space. However, depending on the feature space, those metrics might not be consistent with human perceptions. For example, in RGB color space, the difference measured by

Euclidean distance is not consistent with the color difference perceived by human. Thus, we may either need a better distance measure, or we can transform the values from RGB color space to L*u*v color space which is believed to linearize the perceptibility of color difference. Besides the metric model, another very popular dissimilarity measure is the Kullback-Leibler divergence (KL-divergence) when statistical models are used. The KL-divergence measures the "distance" between two distributions [25]. However, because of the intractability of most continuous densities, the powerful KL-divergence has been limited to the calculation between histograms which only coarsely represent the distributions. Approximations of KL-divergence between Gaussian mixtures were studies in [26]. For retrieval applications, Rubner [27] proposed another interesting distance between two distributions based on the minimal cost that must be paid to transform one distribution into the other. The distance, which is called *Earth Mover's Distance* (EMD), has been used for image retrieval and shown better performance than Jeffrey divergence (a symmetric variant of KL-divergence), χ^2 statistics, and quadratic distance [28]. In [29] and [30], a distance measure based on the approximation of the percentage of clusters of similar frames shared between two video sequences was developed. The fusion of multiple measures from different feature set was studied in [31] where a decision-level aggregation of the descriptor-level distances based on logical operators was proposed.

Video object segmentation and tracking is another challenging problem that has attracted many researchers. Semantic image segmentation is ill-posed itself, but for video data, it may be more possible to segment semantic video objects because of the extra temporal information. In [32], a layered representation of images was proposed by estimating and clustering affine parameters. In [33], a multi-resolution iterative refinement algorithm based on Kalman filtering was proposed. Recently, particle filtering based trackers have also been getting popular [34] [35].

All the content analysis tasks described above can be combined and utilized to achieve a higher level multimedia understanding, i.e. semantic event detection and recognition. Event detection and recognition has been an active research area in the past few years.

Xie [36] studied the event detection in soccer domain. The Hidden Markov Model (HMM) was applied to detect play and break event for soccer video. Dominant color ratio and the magnitude of the motion vectors were used as features. The observations were modeled as a mixture of Gaussians with two mixture components per state. In [37] and [38], hierarchical HMM structures were used to model semantic events. In [39], MPEG-7 audio features and entropic prior HMM models were used to recognize common audio events such as applause and cheering. In [40], Gaussian mixture HMM was applied in DCT domain to detect traffic events. Traffic conditions were divided into six events and each was modeled as a hidden state in HMM framework.

In this section, we have reviewed and discussed a variety of existing techniques for video content based parsing, indexing, retrieval, object-based representation, and semantic event detection and recognition. All the tasks are still ongoing research areas. As described in section 1.2, many challenges still exist. Our focus is to develop video analysis algorithms that makes the processing of visual data more intelligently and efficiently.

1.4 Summary of Contributions

In this thesis, we develop new statistical analysis methods for feature extraction and a spatiotemporal modeling framework for video content analysis. The major contributions are summarized as follows:

1. New feature extraction techniques using statistical analysis are developed. The features well preserve the temporal dynamics and are low-dimensional.

2. Theoretically, we develop a new mathematical model that can be used to analyze the signal or sequential data with strong non-Gaussian distributions. We extend the classical continuous HMM to let the observation densities be represented as a mixture of non-Gaussian distributions, which allows a broader range of observation densities to be modeled. In order to obtain the parametric form representation for the new observation models, the Independent Component Analysis (ICA) mixture model is

integrated in HMM framework to describe the observation densities.

3. We further investigate the new proposed framework. A short proof of convergence is given and the re-estimation formulas for model parameter learning are derived.

The proposed statistical analysis and spatiotemporal modeling techniques have been applied to the following applications for video content analysis. The contributions include

1. New video parsing and indexing algorithms using the statistical-based feature extractions.

2. New video dissimilarity models combining both spatial and temporal information is proposed.

3. New algorithms for video object segmentation and tracking based on probabilistic fuzzy c-means and Gibbs random fields.

4. A semantic event detection solution using the proposed feature extraction and the spatiotemporal framework.

For research in multimedia, especially for video, preliminary implementations are as important as theories because implementations validate the feasibility of the algorithms, the performance, and the user interactivity. In this thesis, we also develop a system to validate the use of the *Synchronized Multimedia Integration Language* (SMIL) standard for video indexing.

1.5 Outline of Thesis

The remainder of this thesis is organized as follows. Chapter 2 describes the basic concepts and theoretical background which are used in our work. In this chapter, we introduce the statistical modeling techniques including basic statistical measures, a high-order statistical technique - Independent Component Analysis, and sequential data modeling - Hidden Markov Model.

In Chapter 3, we develop new feature extraction techniques based on statistical analysis for video parsing and indexing. The proposed online algorithms are based on the combined analysis of mean-variance-skewness. The video transitions such as cuts and dissolves are explicitly modeled and analyzed using basic statistical measures. We also validate the possibility of implementing a web-enabled standard Synchronized Multimedia Integration Language (SMIL) to index the video shots. Finally, experimental results are shown and discussed.

In Chapter 4, we represent a new feature extraction technique using higher order statistics for content analysis of video. ICA is used to project video frames from an illumination-invariant color space to a two-dimensional subspace. In the new feature space, a new video temporal segmentation solution using a dynamic clustering algorithm is developed. The video frames are processed in batch mode and thus can be considered as an off-line method. A key-frame selection scheme for video indexing and video summarization is also developed. Experimental results are shown and discussed.

In Chapter 5, we apply the statistical based feature extraction to video data and develop a new video similarity/dissimilarity model based on dynamic programming. The new distance measure makes use of both spatial and temporal information for video sequence matching in shot-level. Experimental results are shown to verify the performance of the measure.

In Chapter 6, we develop a statistical-based solution for video object segmentation and tracking using probabilistic fuzzy c-means and Gibbs Random Fields. Spatial constraints based on Gibbs random fields are integrated into fuzzy c-means framework for spatial segmentation. Motion vectors based on phase correlation are used to located the active segmented regions. Temporal tracking of video objects is achieved by projecting the blocks from current frame to the next frame. Experimental results are shown to test the effectiveness of the proposed algorithm.

In Chapter 7, the statistical analysis methods and the feature extraction techniques developed in previous chapters are applied to video data to build the foundations for the high-level semantic analysis. In this chapter, first, we develop a new theoretical framework to model

sequential data based on HMM and ICA mixture model. We introduce a new continuous observation density model that is based on the mixture of non-Gaussian densities, and each non-Gaussian component is associated with a standard ICA module. The re-estimation formulas of model parameters are derived. We then develop a video event detection system based on the theoretical framework. As a case study, golf video is used for event detection and recognition. The experimental results are analyzed and discussed.

In Chapter 8, we conclude the work presented in the thesis, and give some discussions of potential applications and future work.

Chapter 2

Preliminaries of Statistical Modeling

In this chapter, we introduce some basic concepts and background materials that will be employed later in this thesis. Basic statistical measures, Independent Component Analysis, and Hidden Markov Model are briefly summarized and reviewed.

2.1 Basic Statistical Measures

Suppose that a random variable (r.v.) X has a discrete distribution for which the probability function (p.f.) is p. The *expectation* of X, denoted by $E[X]$, is defined as follows:

$$E[X] = \sum_{x} xp(x). \tag{2.1}$$

If a random variable X has a continuous distribution for which the probability density function (p.d.f.) is p, then the expectation of X is defined as follows:

$$E[X] = \int_{-\infty}^{\infty} xp(x)dx. \tag{2.2}$$

The expectation of X is also called the *mean* of X. The mean of a random variable is the first order statistics which can be regarded as a measure of the center of gravity of that distribution.

Suppose that X is a random variable with mean $\mu = E[X]$. The *variance* of X, denoted by σ_X^2 is defined as follows:

$$\sigma_X^2 = E[(X - \mu)^2]. \tag{2.3}$$

The nonnegative square root of the variance, i.e., σ_X is defined as the *standard deviation* of X. The variance of a random variable is a measure of the spread or dispersion of the distribution around its mean. A large value of the variance typically indicates that the distribution has a wide spread around the mean, while a small value of the variance indicates that the distribution is tightly concentrated around the mean [41].

The *moments* of a random variable X are defined as:

$$m_i = E[X^i], \quad i = 1, 2, \ldots. \tag{2.4}$$

The expectation $E[X^k]$ is called the k-th moment of X. For any positive integer k, the k-th *central moment* of X, denoted by cm_k, is defined as:

$$cm_k = E[(X - E[X])^k]. \tag{2.5}$$

Obviously, the mean is the first moment, i.e. $cm_1 = \mu$, and the variance is the second central moment, i.e. $cm_2 = \sigma_X^2$. Another frequently used central moment is cm_3 which is known as the *skewness* of X (sometimes the skewness is normalized by σ_X^3). The skewness, denoted by s^3, is defined as:

$$s^3 = E[(X - E[X])^3]. \tag{2.6}$$

The skewness of a random variable is a measure of the degree of the asymmetry of the distribution around the mean.

2.2 Independent Component Analysis and Higher Order Statistics

Independent Component Analysis (ICA) is a recently developed statistical technique which captures the higher order statistics of signals [42]. ICA can be considered as an extension of principal component analysis (PCA). PCA aims to decorrelate the signals, while ICA's task is to blindly separate the signals such that the output signals are mutually independent or as independent as possible. ICA is also closely related to Blind Source Separation (BSS) problem [43] since originally ICA was developed to blindly solve source separations. ICA

later was found to be useful in many other applications, and several algorithms from different views have been developed to solve the ICA problem. In this section, we briefly introduce the ICA model and several estimation algorithms to solve the ICA problem.

2.2.1 ICA Model

In this section, we review the general concepts for ICA. The ICA model, and the identifications of ICA problem are discussed. The potential applications of ICA are also reviewed.

ICA Model

The ICA model assumes the n-dimensional observed random vector $\boldsymbol{o} = [o^{(1)}, o^{(2)}, \ldots, o^{(n)}]^T$ is a linear static transformation of m-dimensional random vector $\boldsymbol{s} = [s^{(1)}, s^{(2)}, \ldots, s^{(m)}]^T$ where the elements are statistically independent to each other. Note that both random vectors $\boldsymbol{o}$ and $\boldsymbol{s}$ are column vectors. Each element in $\boldsymbol{s}$ is called a component or source, and each element in $\boldsymbol{o}$ is called a sensor or observation. Denote the n-by-m mixing matrix as M. The ICA model is described as follows:

$$\boldsymbol{o} = M \cdot \boldsymbol{s}. \tag{2.7}$$

The ICA task is to find a m-by-n filtering matrix W such that the transformed outputs

$$\boldsymbol{y} = W \cdot \boldsymbol{o} \tag{2.8}$$

are mutually independent, given only the observed random vectors. In ICA literature, the matrix M is generally called the *mixing matrix*. However, because we will introduce a mixture model using ICA later in this thesis, we refer the mixing matrix M as the *basis matrix* to avoid potential ambiguities. The ICA model is shown in Figure 2.1. The matrix W is referred as the *filtering matrix*. The outputs $\boldsymbol{y}$ can be considered as estimates of the sources. ICA is often applied to blind separation since it blindly separates the observed signals and creates the independent sources without knowing the basis matrix and the sources. One example which is always used to demonstrate the power of ICA is the famous cocktail party problem. Imagine that there are two persons talking at the same time at a cocktail party. The voices

(sources) are mixed together and form the observed signals, which can be recorded by two devices at different locations, and thus, each recorded (observed) signal can be considered as a linear combination of the sources. The goal is to separate the voices given only the recorded (observed) signals.

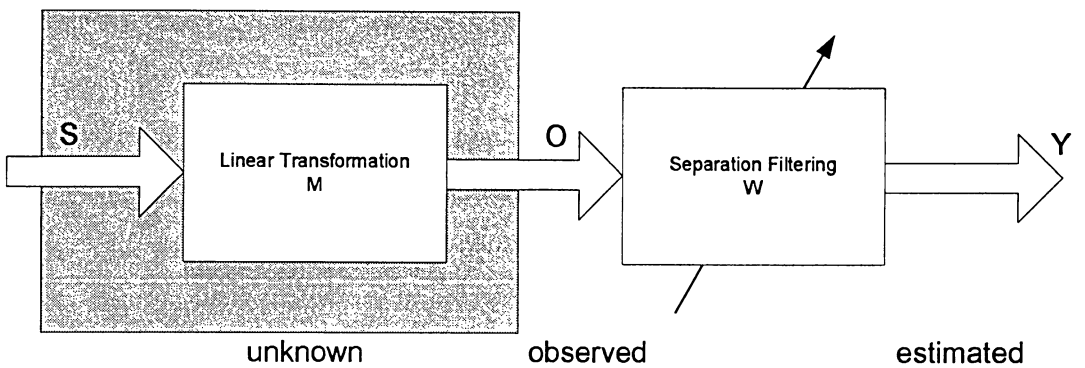


Figure 2.1: ICA Model.

Identifications of ICA Problem

ICA problem may seem to be ill-posed at the first glance. However, by introducing several assumptions, the ICA problem can be solved by using different estimation principles. Below we list the key assumptions and the identifications of ICA problem:

- *Statistically independent sources*: statistical independence is the core assumption on which ICA model holds. The assumption of statistical independent sources means that one source does not give any information about another. For example, in the cocktail party problem mentioned earlier, it is reasonable to assume that one voice (the source) is independent to the other since one voice does not provide any statistical information about the other. Compared with the second-order techniques such as decorrelation, ICA not only decorrelates the data, but also explores the higher order statistics since statistical independence is a stronger condition compared with being uncorrelated.

- *Non-Gaussian sources*: This assumption requires that all sources (independent components), with the exception of only one component, must be non-Gaussian distributions.

- *Linear transformation*: The classical ICA is a non-orthogonal linear transformation that can find and recover the source signals which are linearly mixed.

- *The number of sensors should not be less than the number of sources*: The number of observed signals should be equal to or greater than the number of sources. This condition implies the basis matrix must be a square matrix or a thin matrix. Otherwise, the ICA model becomes an over-complete ICA problem which is not in the scope of our research. The reason is that the dimension of the feature space for video signals is already very high and we are not interested in projecting them into an even higher dimensional space.

- *The basis matrix must be full column rank*: Based on the previous assumption, this condition means each column of the basis matrix should be linearly independent, i.e. the dimension of the column space is equal to the number of the columns. If two columns of the basis matrix are linearly dependent, their corresponding source signals can not be completely separated simply from one unique combination.

- *Sign, scaling, and order ambiguity*: In terms of the statistical characteristics, ICA has sign ambiguity and scaling ambiguity. If we multiply an independent component by -1 or any scalar number, the model is not affected. Also, ICA has order ambiguity, i.e. multiplying a permutation matrix or its inverse does not affect the model either. Thus, when using ICA, we generally drop the time index, and do not consider the order of the sources.

Potential Applications of ICA Problem

Because ICA exploits the higher order statistics, it has shown its power in many potential applications. In [43], the ICA method was used for image noise reduction. ICA takes advantage of the statistical information from images. The additive white Gaussian noise

can be added to the classical model. Based on a maximum likelihood solution, signals can be decomposed into two components, a Gaussian noise component and a non-Gaussian component. Next a "shrinkage" operation is performed in the rotated space, and after that the estimate of the image can be obtained by rotating back.

ICA has also become a very popular signal processing technique in biomedical research. The assumption here is that brain activity and artifacts might be produced by separate processes, and thus separating the sources can be done by exploring the statistical independence between the observed signals. ICA has been applied in Magneto-Encephalography (MEG), Electro-Encephalograph (EEG) [44], and Functional Magnetic Resonance Imaging (fMRI) [45] signals to reduce artifacts and provide better understanding of brain functioning.

ICA can be considered a data dependent decomposition method using non-orthogonal bases. Thus, ICA can be used for data compression, feature extraction [46] [47], and pattern recognition [48]. Note that in order to better understand the signals, ICA can always be applied to extract some "meaningful" independent components beneath the observed signals. For example, in [43], ICA was used on financial data to find underlying structure of stock data and the common factors for cash flow data.

Because the ICA technique is "blind" by nature, it is also a good tool to estimate the convolving filter. Thus, it is also used in blind de-convolution and other system identification problems.

2.2.2 Review of Algorithms and Estimation Principles

After Comon [42] clearly stated the ICA problem and the general framework in 1994, many algorithms based on different estimation principles have been developed. In this section, we review some major estimation methods for the ICA problem.

Non-Gaussian Based Estimation Methods

The Central Limit Theorem says the summation of independent random variables tends to a Gaussian distribution. For example, in Figure 2.2, the left plot and the middle plot are approximations of two uniformly distributed random variables; the plot on the right is the

distribution of the summation of the two random variables. As it can be seen that the summation tends to a Gaussian distribution.

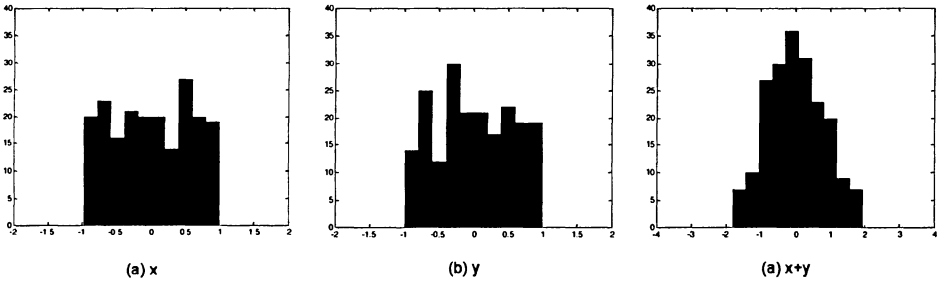


Figure 2.2: Summation of two uniform random variables (approximated by histograms).

Recall that the ICA model, the observed signals are modeled as a linear combination of independent non-Gaussian sources. Thus, intuitively speaking, the observed signals are more Gaussian than the sources. Therefore, maximizing the non-Gaussianity reverse this process and gives us the independent component. There are several estimation methods based on non-Gaussian measures.

- *Kurtosis*: In statistics, kurtosis, the 4-th order cumulants, is the classical measure of non-Gaussianity. It can be easily verified that the kurtosis for Gaussian random variable is zero. For most non-Gaussian random variables, kurtosis is not zero. As a measure of non-Gaussianity in ICA, kurtosis can be either positive or negative. Super-Gaussian has positive kurtosis and sub-Gaussian has negative kurtosis [43]. Generally, we add the constraint that the observed signals have unit variance. Thus, the optimization problem becomes finding the maxima of the kurtosis function on the unit circle. The drawbacks of using kurtosis are that kurtosis is sensitive to outliers and its value may depend on only a few observations [43].

- *Negentropy*: Negentropy is the another method to measure non-Gaussianity. Negentropy J, a modified definition of differential entropy, is defined as

$$J(\boldsymbol{y}) = H(\boldsymbol{y}_{gauss}) - H(\boldsymbol{y}), \tag{2.9}$$

where $H(\cdot)$ is the differential entropy defined as

$$H(\boldsymbol{y}) = -\int p(\boldsymbol{y}) \log p(\boldsymbol{y}) d\boldsymbol{y}. \tag{2.10}$$

The objective function J defined above essentially measures the distance of source probability distributions from a Gaussian distribution. The advantage of this method is that negentropy is well justified in statistical theory. The disadvantage is that negentropy is computationally expensive. In practice, we might want to void the direct calculation of negentropy. Alternatively, we can find and calculate approximations of negentropy.

- *Approximations of Negentropy*: Traditionally, higher order moments are used to approximate the negentropy, for example,

$$J(y) \approx \frac{1}{12}(E[y^3])^2 + \frac{1}{48} kurtosis(y)^2, \tag{2.11}$$

where the random variable y is assumed to be of zero mean and unit variance. In [49] [50], a new approximation method to compute negentropy was proposed. The approximation of negentropy is

$$J(y) \propto [E[G(y)] - E[G(\nu)]]^2, \tag{2.12}$$

where $G(\cdot)$ is a non-quadratic function, and ν is a Gaussian variable of zero mean and unit variance. By carefully choosing the non-quadratic functions, Equation (2.12) can provide better approximations that are robust than moment-based approximations.

Maximum likelihood Estimation Method

Maximum likelihood is another popular method for estimating the ICA model. Given N independent observations $\boldsymbol{O} = \{\boldsymbol{o}(t)\}$, $1 \leq t \leq N$, the likelihood function is given by

$$p(\boldsymbol{O} \mid M) = \prod_{t=1}^{N} p(\boldsymbol{o}(t) \mid M) = \prod_{t=1}^{N} \int p(\boldsymbol{o}(t) \mid M, \boldsymbol{s}(t)) \cdot p(\boldsymbol{s}(t)) d\boldsymbol{s}(t). \tag{2.13}$$

Estimating the basis matrix M or the filtering matrix $W = M^{-1}$ can be done by maximizing the above objective function. Generally we take the log and maximize the log-likelihood $\log(p(\boldsymbol{O} \mid M))$.

The Infomax Estimation Method

The infomax estimation method for ICA problem comes from the principle of network entropy maximization, or "infomax" in neural network community. Suppose that a sigmoid nonlinear function $g(\cdot)$ is chosen properly, let us define $z = g(y)$. Then, for a nonlinear infomax, the optimization function is given by

$$J = H(z) = -\int p(z) \log p(z) dz. \tag{2.14}$$

In practice the nonlinear functions are chosen as the cumulative distribution function (c.d.f.) corresponding to the densities, i.e., the derivative of the nonlinear functions are the probability density functions.

The infomax method for ICA was first introduced in [51], and later [52] improved by using the natural gradient. However, the original infomax ICA algorithm with sigmoidal nonlinearities was only suitable for super-Gaussian source estimations. In [53], an extended version of the infomax ICA algorithm that is suitable for both super-Gaussian and sub-Gaussian was developed.

Connections Between All Princples

In [43], the equivalence between the mutual information and the negentropy as a measure of non-Gaussianity is discussed. In [54], Cardoso showed a surprising result that factorial coding, maximum likelihood estimation, and nonlinear infomax are identical principles in ICA.

2.2.3 Higher Order Statistics

Statistical independence is the core assumption for the ICA model. That leads to the exploration of higher order statistics which are not contained in the covariance matrix. Sources have to be assumed to be non-Gaussian since all the information of Gaussian variables is contained in the covariance matrix and the mean vector. For Gaussian cases, the independence and the uncorrelation are equivalent and thus ICA is equivalent to principle component

analysis. Therefore, in order to exploit the higher order information, we are more interested in non-Gaussian random variables. Also, because independence implies uncorrelated, PCA is often performed as a pre-processing step before ICA.

2.3 Hidden Markov Models

The ICA model described above can exploit higher order statistics from the observed data. When performing ICA tasks, we generally drop the time index and we are only interested in the overall statistics instead of the order. However, for some time series data, such as speech signals and video signals, the temporal characteristics and the order cannot be ignored. Also, such signals often show non-stationary properties since the probability densities may change over time. Thus, a statistical modeling that is suitable for capturing temporal statistics is required for better analyzing observed signals. Hidden Markov Model (HMM) techniques, which are well-known stochastic modeling methods, model non-stationary stochastic sequences by using distinct random transitions among a set of different stationary processes. An HMM model can be considered as a *stochastic finite state automation* [55], which generates a sequence of observations vectors from its hidden states. HMM has been successfully applied in many fields, such as speech recognition [56] [57], handwriting recognition [58], texture classification [59], and blind equalization [60].

2.3.1 Introduction

A classical discrete HMM model assumes the observations can be chosen from finite symbols defined as $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\}$ where M is the number of symbols. We denote the given observation sequence as $\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T$, where T is the length of the sequence, and $\boldsymbol{o}_t, 1 \leq t \leq T$, is the L dimensional feature vector for the t-th sample. Let $q = q_1, q_2, \cdots, q_T$ be the hidden state sequence. A discrete HMM model with N states is determined by the parameters $\lambda = (A, B, \pi)$, where each parameter is explained as follows:

- *State transition matrix*: $A = \{a_{ij}\}, 1 \leq i, j \leq N$ is the state transition probability matrix, where $a_{ij} = P(q_{t+1} = j \mid q_t = i)$ is the probability of state j at time $t+1$ given

the state is i at time t.

- *Observation densities*: $B = \{b_j(k)\}$ is the observation symbol probability distribution for the discrete model, where $b_j(k) = P(\boldsymbol{o}_t = \boldsymbol{v}_k \mid q_t = j), 1 \leq j \leq N, 1 \leq k \leq M$ is the probability of observing $\boldsymbol{v}_k$ given the current state is j at time t.

- *Initial state distribution*: $\pi = \{\pi_i\}, 1 \leq i \leq N$ is the initial state distribution where $\pi_i = P(q_1 = i)$.

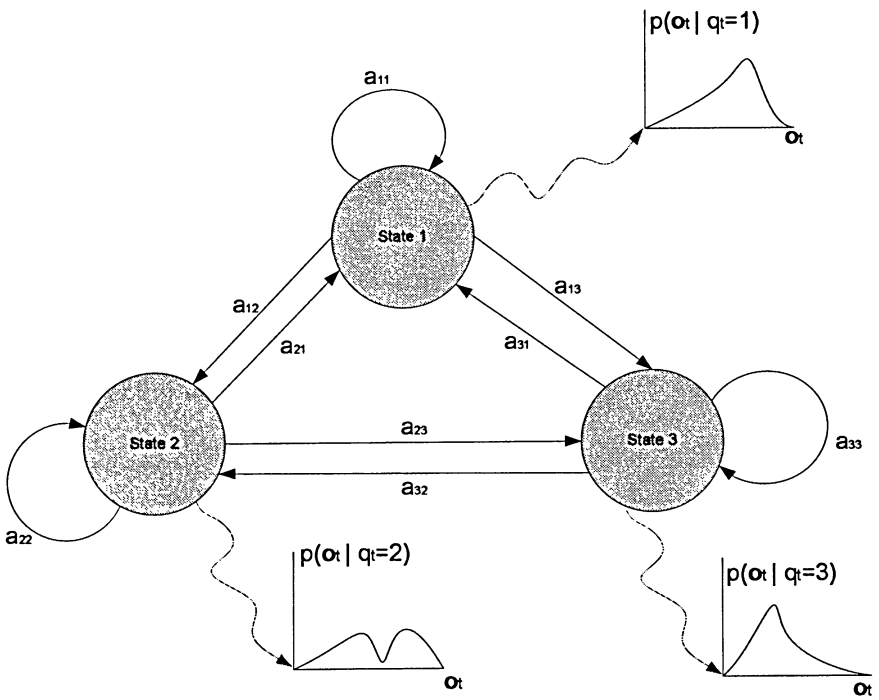


Figure 2.3: Model parameters for a three-state Hidden Markov Model.

Note that the model described above is a discrete model. For continuous observations, the symbol probability distribution $B = \{b_j(k)\}$ will be replaced by $B = \{b_j(\boldsymbol{o})\}, 1 \leq j \leq N$, where $b_j(\boldsymbol{o})$ is the probability density function of the observation at state j. Figure 2.3 shows model parameters for a three-state Hidden Markov Model with continuous observation models.

2.3.2 Basic Problems for HMMs

In order to apply the HMM to real-world applications, there are three basic problems of interest [57]:

1. *Evaluation problem*: The evaluation problem is to measure how well a given model matches the given observation sequence, i.e., given the model parameters $\lambda = (A, B, \pi)$, we need to find the efficient solutions to compute $P(\boldsymbol{O} \mid \lambda)$.

2. *State path recovering*: This problem is to recover the hidden state path, i.e., given the observations, we need to find a state sequence $q = q_1, q_2, \cdots, q_T$ based on some optimality criteria. For example, in speech recognition, the recovered hidden path of speech signals could be the corresponding written words. However, it is worth pointing out that not all recovered hidden states have associated physical meanings. Thus, the hidden state path may or may not be verified.

3. *Model parameter estimation problem*: The model parameter estimation problem is to optimize the model parameters such that the estimated model can best describe how the given observation sequence is generated.

In this thesis, we are only interested in the evaluation problem and the model parameter estimation problem. Recovering the state path is not directly utilized as a modeling method since we generally cannot directly associate a meaningful or generic physical units to the state path for video signals. Also, the state path differs depending on different optimality criteria. For speech recognition, the state path may be important because most speech signals can be mapped to basic linguistic units such as phonemes, or written language units such as words. However, for video signal processing or video content analysis, such meaningful hidden states often do not exist or at least are not easy to validate. Most of the existing research, as well as the research conducted in this thesis, uses the observations and the model structures instead of the state path to model the spatial and temporal characteristics. Note that the model parameters are mainly driven by the hidden state sequence, but we just do not use the state

path as an explicit modeling technique. Therefore, in order to utilize the HMM framework for video content analysis, we mainly consider problem 3 and problem 1. Problem 3, i.e., model parameter estimation, is the core process for most HMM based applications. This process is to adjust model parameters such that the model can best represent how the observations are created. The observation sequence used during the parameter estimation process is also called "training sequence". After the model parameters are determined, we can compute the probability of the observed sequence given the model parameters, i.e., $P(\boldsymbol{O} \mid \lambda)$. This probability gives us the solution to problem 1 and allows us to choose the best match among several candidate models.

HMM's three basic problems have been well studied. The general framework has also been developed, and the solutions for specific observation model and structures have been derived. *The forward-backward procedure* [57] is an efficient procedure to iteratively calculate the model parameters using a few intermediate variables.

Some definitions that are required for forward-backward procedure are reviewed and discussed as follows. The forward variable $\alpha_t(i)$ and the backward variable $\beta_t(i)$ are defined as [57]

$$\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = i \mid \lambda), \tag{2.15}$$

$$\beta_t(i) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T, \mid q_t = i, \lambda). \tag{2.16}$$

Using the forward variable and the backward variable defined above, two probabilities of the joint event can be defined [57]:

$$\xi_t(i,j) \equiv \ P(q_t = i, q_{t+1} = j \mid \boldsymbol{O}, \lambda) = \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(\boldsymbol{o}_{t+1}) \cdot \beta_{t+1}(j)}{P(\boldsymbol{O} \mid \lambda)}, \tag{2.17}$$

$$\gamma_t(i) \equiv P(q_t = i \mid \boldsymbol{O}, \lambda) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(\boldsymbol{O} \mid \lambda)}, \tag{2.18}$$

where (2.17) defines the probability of the joint event: a path passes through state i at time t *and* through state j at time $t+1$, given the available sequence of observations $\boldsymbol{O}$ *and* the parameters of the model λ. Equation (2.18) defines the probability of being in state i at time t, given the observation sequence $\boldsymbol{O}$, and the model λ.

Based on the above intermediate variables, the evaluation problem can be solved iteratively [57] [61] as

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \quad 1 \leq i \leq N. \tag{2.19}$$

2. Induction:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij}] b_j(\boldsymbol{o}_{t+1}), \quad 1 \leq t \leq T-1, 1 \leq j \leq N. \tag{2.20}$$

3. Termination:

$$P(\boldsymbol{O} \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{2.21}$$

The explanation of the above iterative estimation formulas is straightforward based on the definitions of the intermediate variables. The advantages are that the computation complexity is greatly reduced, compared with the direct calculation using the exhaustive search for all possible state sequences.

For the third problem, i.e., the model parameter estimation problem, the solutions using forward-backward procedure are [57]

$$\pi_i^* = \gamma_1(i), \tag{2.22}$$

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{i=1}^{T-1} \gamma_t(j)}, \tag{2.23}$$

$$b_j^*(k) = \frac{\sum_{t=1, \boldsymbol{o}_t = \boldsymbol{v}_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{2.24}$$

The above solutions were developed by [61] [62]. The equivalence between forward-backward procedure and the EM algorithm was noticed by [63]. The model parameter estimation problem can also be considered as an optimization problem with stochastic constraints. Thus, standard gradient techniques can be used to solve the problems.

2.3.3 Types of HMMs

In terms of the observation model, the HMM can be divided into discrete HMM models and continuous HMM models. The former directly takes the finite discrete observations as inputs or uses vector quantization to convert the signals from continuous values to a finite number of values.

In terms of the model structure or the state configuration, the types of HMM can be categorized as ergodic models, left-right models, etc. Generally speaking, the state configuration is application-dependent, and it also provides a way to include *a prior* knowledge into the framework. Note that the configuration is determined by the state transition matrix A in the HMM framework. For an ergodic model which has the property that every state can be reached from every other state, all elements in the state transition matrix should be positive. Examples of ergodic HMM and left-right HMM are shown in Figure 2.4 and 2.5. For some applications, if some state transitions are not possible, such as the left-right model, we may directly set the corresponding entries in the state transition matrix to zeros. The above constraints can be used as initializations when estimating the model parameters. Note that not only the state configuration can be changed, the number of states can also be changed. Decomposing one state into multiple states or merging multiple states into one state can be done through manipulating the state transition matrix according to certain criteria.

Other variants of HMM models are also developed. For example, in [64], an explicit state duration modeling was developed. This extension allows the state duration to be modeled by any density instead of exponential distribution used in standard HMM. The variable duration HMM does improve the flexibility of describing the state duration; however, the computation load is also greatly increased. Furthermore, the number of free parameters is large and the parameter estimations become very complicated. For some types of HMMs, such as the left-right HMM, the average duration is already modeled in the model structures, therefore, the explicit state duration modeling may not be necessary for those cases [57].

Another extension of the standard HMM is to introduce the dynamic learning of model structure from data. In [65], a merging process was proposed using "neighboring merging"

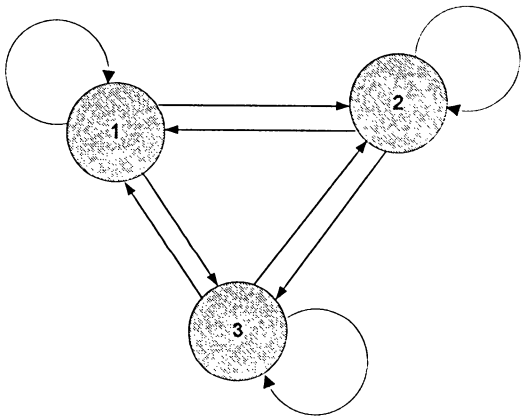


Figure 2.4: An example of 3-state ergodic HMM.

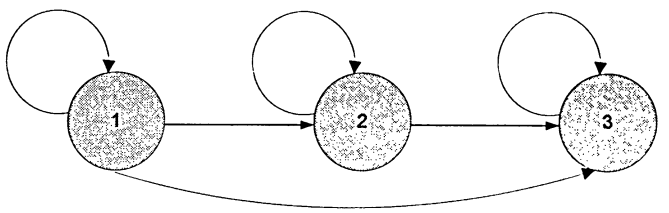


Figure 2.5: An example of 3-state left-right HMM.

and "v-merging". A "neighboring merging" merges states that share a link and have the same class label, and the "v-merging" merges the states that have the same class label and share transitions from or to a common state. A more generic framework for structure learning in dynamic Bayesian networks was developed in [66], and HMM can be considered as a special case of dynamic Bayesian network. Hierarchical Hidden Markov Models were used in [37] to model the semantic events in soccer video. However, learning the structure not only introduces more free parameters, but also increases the complexity and computation load of the system.

Chapter 3

Video Parsing and Indexing Using Basic Statistical Measures

Video parsing and video indexing is an important early stage of content-based video analysis. In this chapter, we present new statistical analysis methods to extract features and analyze the properties for video transitions. Based on the new methods, we develop a new web-enabled video parsing and indexing system that integrates Synchronized Multimedia Integration Language (SMIL) standard. Abrupt transition detection is achieved by an enhanced histogram-based method that is robust to illumination changes. For gradual transition detection, new features are introduced for dissolve detection. The proposed dissolve detector is based on a combined analysis of mean-variance-skewness of intensity. Compared with existing variance-based approaches, the introduced new features improve the discrimination ability on shot boundaries. We also describe methods for eliminating false positives. Experimental results show that the proposed algorithms can effectively detect shot boundaries. Detected scenes and other cinematic attributes are structured and organized by integrating HTML and SMIL. For each video file, the system generates a table-of-contents indexing file. The user-friendly interface provides web-based interaction, browsing and previewing of video content.

3.1 Introduction

Content based video analysis and retrieval has become an area of active research in recent years. The first step of video content analysis is to segment video into its constituent shots, and high-level scenes or episodes. As the building blocks of video projects, video shots need to be identified effectively and efficiently. The research of video parsing focuses on the detection of both abrupt shot transition (i.e., cut) and gradual shot transition (fade/dissolve). Many automatic techniques have been developed to detect video shot boundaries in both the compressed and the decompressed domains. Zhang *et al.* [3] proposed pair-wise pixel comparison, likelihood ratio and histogram comparison for abrupt transition detection. Edge changes can also be used as a good feature for shot detection [67]. Yeo and Liu [68] detected scene changes by using pixel difference and luminance histograms based on DC-images in compressed domains. In [69], recently developed methods for shot detection were reviewed in detail, and a statistical detector was proposed based on motion compensation. Shot detection techniques can be categorized into feature-based [67], model-based [18] and statistical [19] methods. Most of the above techniques can achieve good performance on hard cut detection.

However, compared with abrupt transition detection, gradual transition still remains a challenging problem. When dealing with gradual transition, Zhang *et al.* [4] used a twin threshold mechanism based on histogram difference metric. Frame differences were accumulated when inter-frame difference was above the lower threshold but smaller than the higher threshold. When the accumulated difference exceeded the higher threshold, a gradual transition was defined. In [67], edge and contour changes were used for gradual transition detection. Another feature that is commonly used for dissolve detection is intensity variance. During a dissolve transition, the intensity variance curve forms a downwards-parabolic shape. The variance-based approach was first introduced by Alattar [18], and many other researchers have used this feature to build their dissolve detectors [69] [19]. Alattar [18] proposed to take the second order difference of intensity variance, and then check two large negative spikes. However, such a pattern may not be pronounced due to object/camera motion and noise. Truong *et al.* [19] proposed an improved version by adding more constraints. Lienhart [70]

introduced a somewhat different approach that includes a transition synthesizer and a neural network classifier, and dissolves are detected by a multi-resolution search.

Most of the existing techniques require careful selection of thresholds to achieve good performance. Some key factors that affect the performance of shot detection are illumination changes and object/camera motion. Since histograms do not carry spatial information, they are expected to be robust to object and camera motion. However, illumination changes can cause serious problems. Some feature-based methods, for example, based on the appearance of intensity edges, are less sensitive to illumination changes, but they are not robust to the motions of large objects and extra computations are also introduced. In this chapter, new algorithms are proposed for shot detection. We present a cut detection algorithm that is robust to illumination changes. Dissolve detection is achieved by a combined analysis of intensity moments. In addition to the variance feature, the introducing of mean and skewness adds more constraints and improves the discrimination ability of frame distances on shot boundaries, and thus provides a more robust way for shot detection. Experimental results show that the proposed mean-variance-skewness approach can capture those dissolves whose intensity patterns are not so obvious if using only variance feature.

To structure and describe video content, media description scripts or templates need to be developed. In [71], MPEG-7 and MPEG-21 were used to design a video personalization and summarization system. While MPEG-7 defines several levels of abstraction and provides a standard set of tools for describing multimedia content, its scope does not focus on the storage, delivery or presentation of digital video. Considering the growing amounts of online digital video with the success of the Internet, it is desirable to incorporate web-based technologies in content based video analysis projects. In this work, we describe the use of the Synchronized Multimedia Integration Language (SMIL) standard for building the web-enabled video indexing system. In the proposed system, SMIL bridges the gap between the structure of video content and its web-based presentations. User friendly web-enabled interaction, integration and synchronization of video segments are realized by hybrid documents combining HTML and SMIL.

3.2 Overview of the Proposed Video Parsing and Indexing System

Figure 3.1 shows a web-enabled video indexing system which includes frame-level playback, feature extraction, cut detection, dissolve detection, a SMIL generator and Graphic User Interface (GUI).

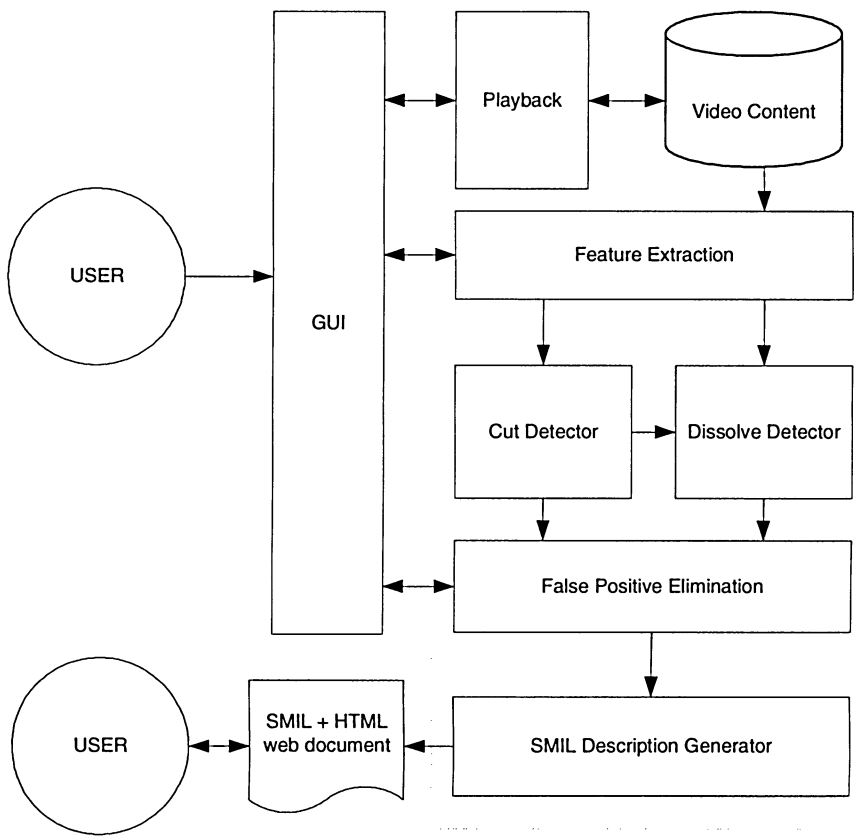


Figure 3.1: Architecture of a web-enabled video parsing system.

After features are extracted from video content, cut detection is performed, followed by dissolve detection. Then, false positives are eliminated by a validation process. SMIL and HTML are used to describe the structures of video content including shot boundaries,

video format, frame rate and other cinematic information. For each video clip, the system generates a web document as a table-of-contents, which can be used for previewing and browsing video content online.

3.3 Abrupt Transition Detection

3.3.1 Improved Cut Detection

Histogram based methods are widely used for cut detection. Compared with other techniques, a histogram is robust to object motion, and it is simple and fast to calculate intensity or color histogram difference between two consecutive frames. Given a video sequence with N frames, denote the n-th frame as f_n, $n = 1, 2, \ldots, N$. Let $H(f_n, k)$ denote the value of the k-th bin of the histogram for the frame f_n. Suppose there are total K bins in the histogram. Then the histogram difference at time n, denoted by $D_h(n)$, can be defined as follows

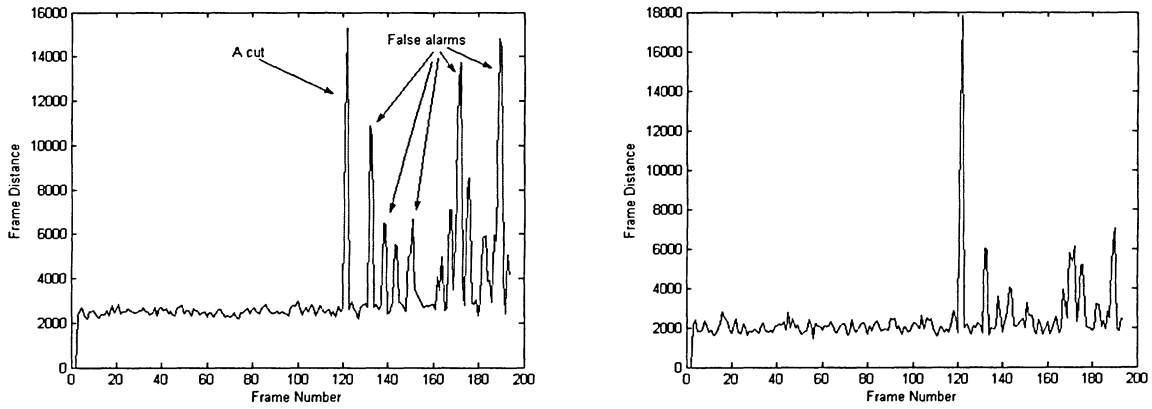
$$D_h(n) = \frac{1}{U} \sum_{k=1}^{K} |H(f_{n+1}, k) - H(f_n, k)|, n = 1, 2, \ldots, N, \tag{3.1}$$

where U is the normalization factor. Histogram difference is a measure of dissimilarity between two consecutive frames. For cut transitions, visual characteristics are expected to change sharply at short boundaries, and thus the histogram difference can capture the visual discontinuities between shots. In practice, histogram based methods are the most common approach to shot detection, since they provide a good trade-off between accuracy and computational efficiency [3] [72] [2]. However, most histogram based methods are very sensitive to lighting changes. An example is shown in Figure 3.2. The luminance component is used to calculate the histogram difference. The first peak at frame 120 represents a real shot cut, and other peaks are caused by camera flash scenes. It can be seen that these illumination changes cause serious problems for cut detection. In this section, we propose a new histogram-based algorithm that is robust to lighting changes. The three channels in RGB color space are converted to the opponent color space. Only the red-green (R-G) component is used to compute the histogram difference. The opponent color representation

of the RGB color space is defined as [73]

$$(R+G+B, R-G, 2B-R-G), \tag{3.2}$$

where R, G and B are red, green and blue channels respectively. By choosing the opponent color space, the proposed cut detection algorithm is less sensitive to lighting changes. As shown in Figure 3.2 (b), the performance is significantly improved. Experimental results show that our method performs better, compared with working only with chromatic color components in YCbCr color space. The conversion from RGB color space to its opponent color representation is computationally efficient. The advantage of this representation is that the last two chromaticity axes are invariant to changes in illumination intensity and shadows. The same video sequence is used to calculate the histogram differences in Figure 3.2. It can be readily seen that the effects caused by camera flash scenes are significantly reduced.



(a) (b)

Figure 3.2: Improved cut detection: (a) before the improvement (b) after the improvement.

Because the distribution of frame distances varies from video to video, an adaptive threshold is more reasonable than a globally fixed threshold. It is worth mentioning that the normalization factor U defined in (3.1) is crucial for robust adaptive threshold, since the video

data may have different image sizes. A temporal sliding window of the length $2w + 1$ with $w = 8$ and centered at current frame n is used to capture the local characteristics. A hard cut candidate is detected at frame n if the following conditions are satisfied:

1. $D_h(n)$ has the maximum value inside the sliding window, *i.e.*, $D_h(n) \geq D_h(k), \forall k \in [n - w, n + w]$.

2. The difference between $D_h(n)$ and the median value of the sliding window is larger than a given threshold τ_1 .

An example based on the proposed cut detection algorithm is shown in Figure 3.3.

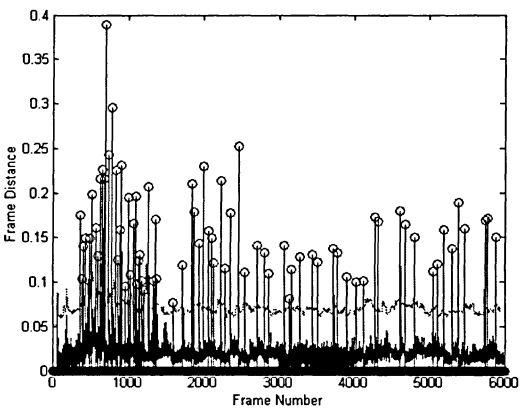


Figure 3.3: Cuts detected by adaptive threshold in the opponent color space (TV show "Friends").

3.3.2 False Positive Elimination

Due to camera/object motion and noise, some frames might be mistakenly identified as shot boundaries. Thus, a false positive elimination process is necessary. The validation process includes two criteria. First, a hard cut candidate at time n is declared as a false positive if frame $(n + 2)$ is similar as frame $(n - 2)$, i.e.

$$\sum_{k=1}^{K} |H(f_{n+2}, k) - H(f_{n-2}, k)| < \tau_2, \tag{3.3}$$

where τ_2 is a given threshold. Second, excluding the maximum value $D_h(n)$ of the sliding window, we calculate the left half maximum denoted as M_L and the right half maximum M_R. Let M_V denote the average of the two maximums. If the difference between $D_h(n)$ and M_V is less than a given threshold τ_3, the cut candidate $D_h(n)$ is deemed as a false positive; that is

$$M_L = \max\{\forall\, D_h(i), i \in (n-1-w, n-1)\}, \tag{3.4}$$

$$M_R = \max\{\forall\, D_h(i), i \in (n+1, n+1+w)\}, \tag{3.5}$$

$$M_V = (M_L + M_R)/2, \tag{3.6}$$

$$|D_h(n) - M_V| < \tau_3. \tag{3.7}$$

The above criteria can effectively eliminate the false positives. The first criterion defined in (3.3) is to measure the difference of visual characteristics between two shots. The second criterion compares the maximum value at the center with the second and third peaks, and such criterion can remove the false positives caused by dissolves.

3.4 Gradual Transition Detection

3.4.1 Mean-Variance-Skewness

Among the existing methods, the detection of gradual transitions is less mature compared with cut detection. One of the main reasons is that it is difficult to define and capture the visual discontinuities for gradual transitions. Most of the recent research work focuses on dissolve detection, since dissolve is the dominant editing style in gradual transitions. In this section, we also focus on dissolves instead of other types of gradual transitions. During a dissolve, intensity variance has been proved to show a parabolic shape [18]. However, such patterns might not be so obvious due to motion and noise. In the proposed algorithm, in addition to intensity variance, mean and skewness are introduced as new features. The first order (mean - μ), the second order (variance - σ^2) and the third order (skewness - s^3)

intensity moments for the n-th frame are defined as

$$\mu(n) = \frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} f(i,j,n), \tag{3.8}$$

$$\sigma(n) = \{\frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} [f(i,j,n) - \mu(n)]^2\}^{1/2}, \tag{3.9}$$

$$s(n) = \{\frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} [f(i,j,n) - \mu(n)]^3\}^{1/3}, \tag{3.10}$$

where $f(i,j,n)$ is the intensity value of the image pixel at location (i,j) for the n-th frame. M and N are image width and height respectively. The above moments are functions of time. Assuming a frame at time t is defined as $f(x,y,t)$, and x, y, t are continuous variables. A dissolve transition with duration of T can be considered as a mixture of two shots $f_1(x,y,t)$ and $f_2(x,y,t)$. During dissolve transition, the intensity of one shot decreases, and the intensity of the other increases. The dissolve editing style can be approximated by choosing two linear scaling functions $g_1(t)$ and $g_2(t)$ as

$$g_1(t) = \frac{T-t}{T}, \quad g_2(t) = \frac{t}{T}, \quad t \in [0,T]. \tag{3.11}$$

The dissolve sequence $D(x,y,t)$ for $t \in [0,T]$ can be defined as [74]:

$$D(x,y,t) = g_1(t) f_1(x,y,t) + g_2(t) f_2(x,y,t). \tag{3.12}$$

Assuming two shots $f_1(\cdot)$ and $f_2(\cdot)$ are statistically independent and roughly ergodic random processes [18] [74]. The intensity variance is given by

$$\sigma^2(t) \equiv Variance(D(x,y,t)) = B_2 t^2 + B_1 t + B_0, \tag{3.13}$$

where coefficients B_2, B_1, and B_0, are independent of t. Thus, ideally, before and after a dissolve transition, the variance is roughly constant, and during the transition, the variance curve forms a parabolic shape. Based on the same assumptions, the mean can be calculated

as

$$\begin{aligned}
\mu(t) &= E[D(x,y,t)] \\
&= E[g_1(t)f_1(x,y,t) + g_2(t)f_2(x,y,t)] \\
&= g_1(t)\mu_1(t) + g_2(t)\mu_2(t) \\
&= (\frac{T-t}{T})\mu_1 + (\frac{t}{T})\mu_2 \\
&= A_1 t + A_0,
\end{aligned} \tag{3.14}$$

where coefficients, A_1 and A_0, are independent of t. Equation (3.14) shows that the mean curve forms a line during a dissolve. Similarly, the skewness can be derived as

$$\begin{aligned}
s^3(t) &= Skewness(D(x,y,t)) \\
&= E[(D(x,y,t) - \mu_D)^3] \\
&= E\{[g_1(t)f_1(x,y,t) + g_2(t)f_2(x,y,t) - \mu_1(t) - \mu_2(t)]^3\} \\
&= C_3 t^3 + C_2 t^2 + C_1 t + C_0,
\end{aligned} \tag{3.15}$$

where the coefficients C_3, C_2, C_1, and C_0 are independent of t. It can be seen that the skewness forms a cubical curve during a dissolve. Skewness characterizes the degree of asymmetry of the distribution around its mean. Two shots with different visual characteristics are expected to have different skewnesses, which are connected by a cubical curve during the dissolve. Another interpretation is that we can easily make connections between moments and distance. The first moment actually defines the 1-norm, and the k-th moment is related to the k-norm. Thus, if we only consider the absolute value, the skewness feature is nothing but a number that measures the difference between the distribution and its mean based on a real metric, i.e.

$$s = d_3(f,\mu) = \|f - \mu\|_3 = [E[|f - \mu|]^3]^{1/3}. \tag{3.16}$$

The skewness that provides more information can be used as a feature for analyzing shot transitions. Figure 3.4 shows how variance and skewness are affected during a dissolve transition.

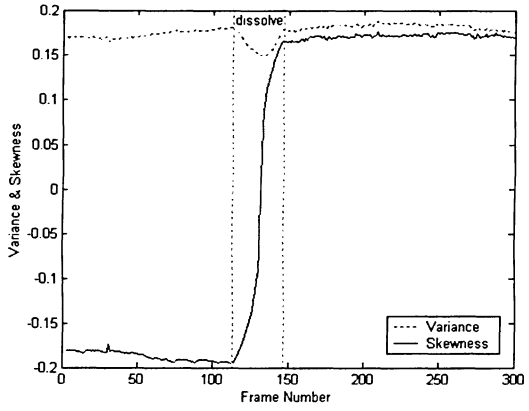


Figure 3.4: Variance and skewness curves during a typical dissolve.

It can be seen that intensity variance forms a parabola while skewness forms a cubical curve. Both features can capture the dissolve. But, numerically speaking, skewness provides higher discrimination ability at the shot boundary since its value changes from -0.19 to 0.16 while variance only changes from 0.17 to 0.14.

Another example is shown in Figure 3.5. This transition contains some extreme factors such as fast camera motion and similar scenes between two shots. Variance and skewness features extracted from the video (see Figure 3.5) are plotted in Figure 3.6. In this case the parabola pattern of variance shown in Figure 3.6 is not obvious due to motion and noise. But skewness is still a good feature to identify the dissolve. When a dissolve joins two similar scenes, at the beginning of the dissolve, the intensity of one shot decreases, but at the same time, it is compensated by similar intensities from the other shot. Such situation can cause serious problems for variance-based approaches. But by exploiting higher order (such as the skewness curve) feature, it is still possible to capture such dissolve transitions.

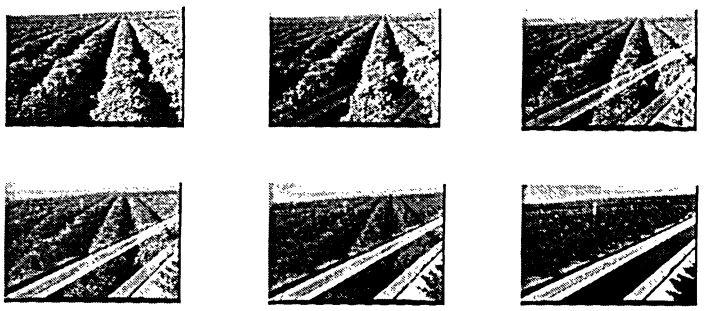


Figure 3.5: An example of dissolve transition (frames chosen at 995, 1000, 1005, 1010, 1015 and 1025).

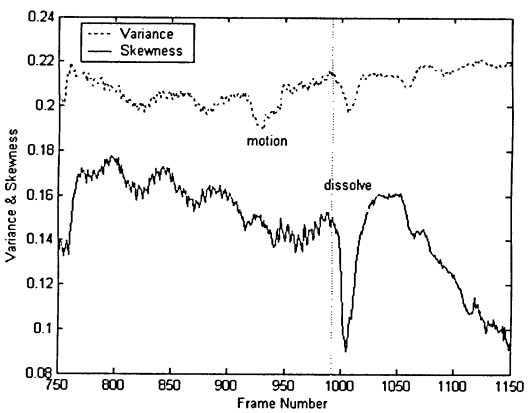


Figure 3.6: Variance and skewness curves during a dissolve.

In Figure 3.6, the cubical curve becomes a parabola-like curve when the cubical coefficient is close to zero.

3.4.2 Dissolve Detection

For dissolve detection, a new method based on a combined analysis of mean-variance-skewness is proposed in this chapter. The dissolve detector takes the output of cut detector

as input. Therefore, we assume there are no abrupt transitions within each input video segment. A temporal sliding window of size $2w + 1$ with $w = 16$ and centered at current frame is chosen to adaptively detect the features. Dissolve detection is achieved by three parts consisting of a variance based detection, a mean-skewness detection and a validation phase that is used to remove false positives.

During variance-based detection, a median filter with a length of three is applied to the variance curve to remove noise. And then a dissolve transition at time n is detected if all the following conditions are satisfied

1. Variance function $\sigma(n)$ has the minimum value in the sliding window; that is

$$\sigma(n) \leq \sigma(k), \forall\, k \in [n - w, n + w]. \tag{3.17}$$

2. Let $\sigma_{mean}[i, j]$ denote the mean value of $\sigma(k)$ between the interval $[i, j]$, i.e.

$$\sigma_{mean}[i, j] \equiv mean\{\forall\, \sigma(k), k \in [i, j]\}. \tag{3.18}$$

   In order to match the downwards-parabolic shape of variance curve within the sliding window, the left half of the curve should be decreasing while the right half should be increasing. From this, we have the following two conditions

$$\sigma_{mean}[n - w, n - w/2] > \sigma_{mean}[n - w/2, n], \tag{3.19}$$

$$\sigma_{mean}[n, n + w/2] < \sigma_{mean}[n + w/2, n + w]. \tag{3.20}$$

3. To make sure the curve has deep "valley" and strong "shoulders", the following two conditions should be satisfied

$$\sigma_{mean}[n - w - w, n - w] - \sigma(n) > \tau_4, \tag{3.21}$$

$$\sigma_{mean}[n + w, n + w + w] - \sigma(n) > \tau_4, \tag{3.22}$$

   where τ_4 is a given threshold.

4. A curve fitting with a degree of two is used to further match the parabola pattern in the sliding window. The performance is evaluated by the estimated quadratic coefficient $B_2{}'$ (see (3.13)) and the fitting error. We have

$$B_2{}' > \tau_5, \tag{3.23}$$

$$\sum_{i=n-w}^{n+w} |\sigma(i) - \sigma'(i)| < \tau_6, \tag{3.24}$$

where τ_5 and τ_6 are given thresholds, and $\sigma'(i)$, $i \in [n-w, n+w]$, is the value of the estimated curve evaluated at point i.

For variance-based detection, the above four conditions are used to match the parabolic shape. A dissolve is detected if all conditions are satisfied.

The mean-skewness detection combines mean and skewness features for shot detection. First, a median filter with a length of three is applied to the skewness curve to remove noise. And then, the first order difference of skewness curve is calculated and its absolute value is used as input. Dissolve detection now becomes measuring the input and finding the large positive spikes. A sliding window with a length of $2w+1$ with $w = 16$ is used to adaptively calculate the local properties. Mean curve is used to validate the detected transitions. Let $s(k)$, $\forall k \in [1, N]$, denote the skewness curve after being median-filtered. The first order difference $S(k)$, $\forall k \in [2, N]$, is given by

$$S(k) = |s(k) - s(k-1)|, \forall k \in [2, N]. \tag{3.25}$$

The frame at time n is declared as a dissolve boundary if all the following conditions are satisfied:

1. $S(n)$ has the maximum value in the sliding window; that is

$$S(n) \geq S(k), \quad \forall k \in [n-w, n+w]. \tag{3.26}$$

2. Denote $S_{median}(n)$ as the median value of the sliding window centered at current frame n; that is

$$S_{median}(n) \equiv median\{\forall S(k), k \in [n-w, n+w]\}. \tag{3.27}$$

The difference between the median value and $S(n)$ should be greater than a given threshold τ_7; that is

$$S(n) - S_{median}(n) > \tau_7. \tag{3.28}$$

3. A regression line is used to fit the mean curve $\mu(k)$, $\forall k \in [n-w, n+w]$, inside the sliding window. This condition requires that the fitting error should be less than a given threshold τ_8. From this, we have

$$\sum_{i=n-w}^{n+w} |\mu(i) - \mu^{'}(i)| < \tau_8, \tag{3.29}$$

where $\mu^{'}$, $\forall i \in [n-w, n+w]$ is the value of the estimated curve evaluated at point i.

A dissolve transition is detected if all three conditions are satisfied.

3.4.3 False Positive Elimination

In the false positive elimination phase, the shot lists obtained from variance-based method and mean-skewness method are merged into one list for further analysis. If the distance between two consecutive dissolves is less than the length of the sliding window $2w + 1$, duplicate entries are defined. In that case, we merge the overlapped dissolves into one dissolve. For each dissolve in the shot list, the histograms from frame $(n + w)$ and $(n - w)$ are compared to validate the results. If their difference is less than a given threshold , the dissolve is considered a false positive.

$$\sum_{k=1}^{K} |H(f_{n+w}, k) - H(f_{n-w}, k)| < \tau_9. \tag{3.30}$$

The elimination criterion is based on the assumption that the visual characteristics from two shots are expected to be different.

3.5 Experimental Results

Extensive experiments are performed to test the proposed shot detectors. Two TV shows, "Friends" and "Sex and the city" were selected, and documentary video data were collected

from Carnegie Mellon University's The Informedia Project at "The Open Video Project" [75]. Performance is evaluated using *precision* and *recall*. Precision and recall are defined as

$$precision = \frac{N_{correct}}{N_{correct} + N_{false}}, \tag{3.31}$$

$$recall = \frac{N_{correct}}{N_{correct} + N_{missed}}, \tag{3.32}$$

where $N_{correct}$ is the number of shot boundaries that are correctly detected, N_{false} is the number of false detected shot boundaries, and N_{missed} is the number of missed shot boundaries.

Experimental results are presented in Table 3.1. Precision and recall for hard cuts were obtained as 93.4% and 97.4% respectively. For dissolve detection precision and recall were 73.7% and 82.4%. In TV show "Friends", the three false alarms are caused by object motion, and fade in/out effects. In the show "Sex and the city", fast motion blur is used to connect two scenes. In fact, all three false alarms for hard cuts were caused by such special editing effects. Even though they were counted as errors in our tests, we could argue that they are actually shot boundaries. The documentary video data contain many water scenes and camera motions from close-up to establishing shot. Editing effects such as zoom-ins, zoom-outs, and camera panning are also used extensively. As it can be seen from the Table 3.1, the general performance of the documentary is not as good as TV shows, especially for cut detection. Part of the reason is that some transitions join similar outdoor scenes.

Table 3.1: Detection results for hard cuts (H) and dissolves (D).

Test Data	Total		Missed		False	
	(H)	(D)	(H)	(D)	(H)	(D)
Friends	73	4	0	1	3	0
Sex and the City	53	3	0	1	3	4
Documentary	64	44	5	7	7	11
Total	190	51	5	9	13	15

In one video clip from the documentary data, among the six dissolve transitions, only

one dissolve can be detected if using only variance feature. But all six dissolves can be successfully identified if skewness feature is added. To compare the results with other works, we refer to [69] [19] [74]. For dissolve detection, a precision of 75.1% and recall of 82.2% is reported in [19], and Lienhar [74] obtained a precision of 82.4% and recall of 75% by using neural networks. Hanjalic [69] reached a precision of 79% and recall of 83% with a smaller test set containing only 23 dissolves. Best results for dissolve detection still use intensity variance feature [69] [19]. Even though video data collected in the above works were carefully selected to contain as many effects as possible, the performance evaluations from different researchers are still based on different materials. However, by introducing mean-variance-skewness and the combined analysis of these new features, we present new patterns and criteria for analyzing dissolve transitions. The experimental results show that the proposed algorithms are effective for shot boundary detection. Also, the methods are computationally efficient.

3.6 A Web-enabled Integrated System

A system tool is developed to integrate the proposed shot detection algorithms. The graphic user interface of the system is shown in Figure 3.7. The system provides frame-level playback. Both manual shot detection and automatic shot detection are supported. After shots are automatically detected, users can edit the shot list, for example, to merge or split shots. In the proposed system, Synchronized Multimedia Integration Language (SMIL) standard is chosen as multimedia content descriptor. SMIL is a web multimedia format developed by the World Web Consortium (W3C) and released in 1998. SMIL provides a cross-industry support for synchronized multimedia integration [76]. It is built on Extensible Markup Language (XML) and allows users to write and publish interactive multimedia online. SMIL syntax and semantics can also be incorporated into other XML-based languages for multimedia timing and synchronization. A simple example of hybrid document combing HTML and SMIL is shown below.

```
<html xmlns:t="urn:schemas-microsoft-com:time">
<head>
<?import namespace="t" implementation="#default#time2">
</head>
<body>
<input id="button1" type="button" value="preview" fill="freeze" />
<t:video style="width:100; height:80px;"
src="./aquarium1.mpeg" clipBegin="00:00:00.000"
clipEnd="00:00:08.068" begin="button1.Click"
type="mpeg" />
...
</body>
</html>
```

After shot boundaries are detected, the shot list and other cinematic attributes are managed by a SMIL-based web document. The table-of-contents web-enabled indexing file generated by the tool is shown in Figure 3.8. Web users can browse and preview the video segments, and jump to a specified location from frame-level. During the implementation, we found that SMIL standard is an effective media description for video structuring and indexing, and its close connection to web makes it very convenient to build and present structured web-enabled multimedia content. Also, keywords and conceptual attributes can be embedded in the SMIL-based indexing file that could be used by existing text-based search engines to realize video web search request.

3.7 Summary

We have presented new feature extraction techniques and algorithms using statistical analysis for shot boundary detection. Cut detection is achieved by choosing the opponent color space that is robust to illumination changes. Dissolve detection is based on a combined analysis of mean-variance-skewness. By introducing these new features and criteria, the proposed dissolve detector has provided a new way to identify and analyze dissolve transitions. Experimental results show that the proposed algorithms can effectively detect both abrupt transitions and dissolve transitions, and are computationally efficient. We also presented a

system tool to structure and organize the detected shots. Shots and video information are managed and indexed by integrating SMIL web multimedia standard. That makes the system interoperable with existing web-based techniques. The generated indexing file provides functionalities like web-based user interaction, browsing and previewing of video content.

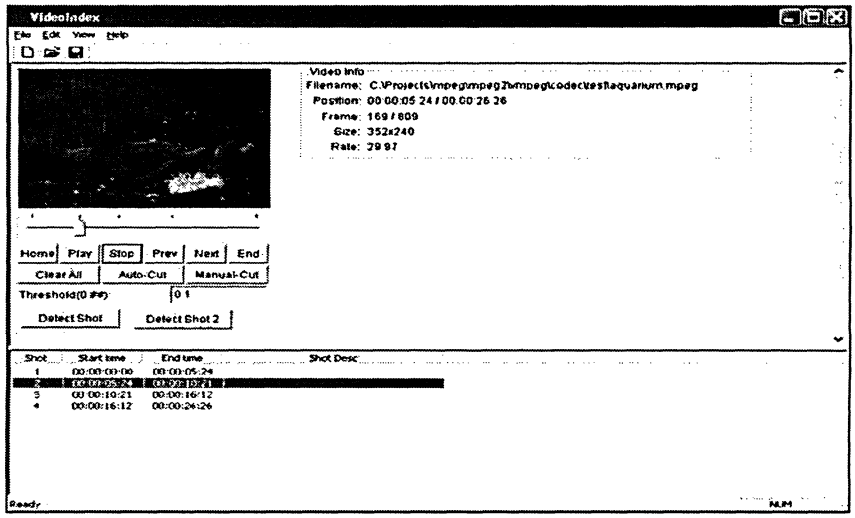


Figure 3.7: System graphic user interface.

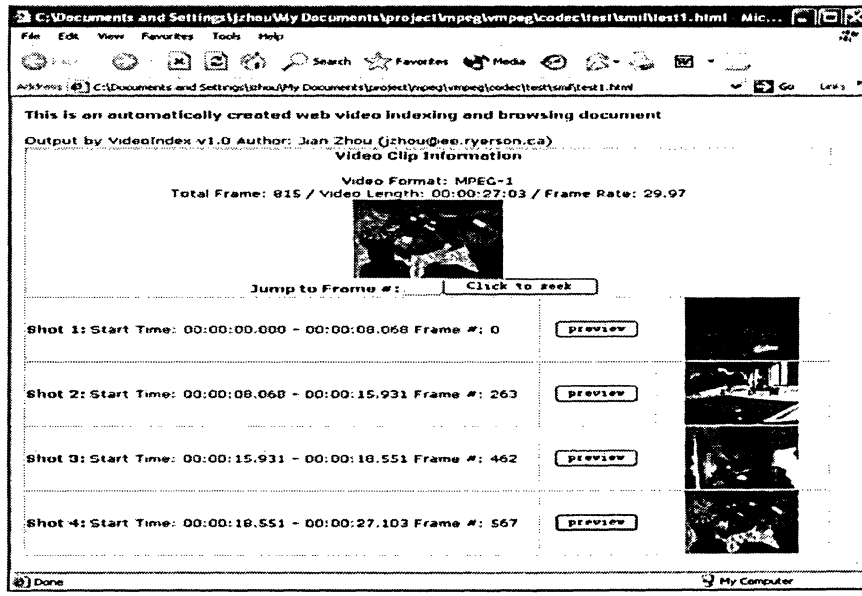


Figure 3.8: Generated HTML+SMIL indexing file.

Chapter 4

Video Parsing and Indexing Using Independent Component Analysis

In the previous chapter, we present the algorithms using basic statistical measure for video parsing and indexing. In this chapter, we present a new statistical analysis method and a feature extraction technique using higher order statistics for video parsing and indexing. Independent Component Analysis (ICA) is used for high order statistical analysis. The new ICA-based method can be regarded as batch-mode processing or an off-line method since all video frames are processed during feature extraction. By projecting video frames from illumination-invariant raw feature space into low dimensional ICA subspace, each video frame is represented by a two-dimensional compact feature vector. An iterative clustering algorithm based on adaptive thresholding is developed to detect cuts and gradual transitions simultaneously in ICA subspace. A video indexing scheme based on the clustered video frames is also developed. Experimental results successfully validate the new method and show its effectiveness for video parsing and video indexing. The comparison between the off-line method and the online method is discussed later in this chapter.

4.1 Introduction

As mentioned the previous chapter, video parsing is to temporally segment a video into its constituent shots and thus recovering the elementary units of a video. The research of video parsing focuses on the detection of two types of transitions: abrupt transition (cut) and

gradual transition (fade/dissolve).

Many automatic techniques have been developed to detect video boundaries. The detailed reviews are discussed in section 1.5 and section 3.1. Most of the existing techniques can achieve relatively good performance on hard cut detection. However, gradual transitions, especially dissolves are generally more difficult to detect. One of the main reasons is that it is difficult to define and capture the visual discontinuities. Therefore, new features, such as edge changes and intensity variance, have been introduced to detect dissolves. The most commonly used feature for dissolve detection is intensity variance. The intensity variance curve forms a downwards-parabolic shape during a dissolve and it has been used in many dissolve detectors [69] [18] [19]. In our previous work, we introduced a new feature based on skewness, and dissolves were detected by a combined analysis of mean-variance-skewness [77]. Most of these existing techniques require careful selection of thresholds to achieve good performance. Such parameter tuning is undesired, especially for the video data from different genres.

In this chapter, we present data-driven feature extraction for shot detection based ICA model. The same feature is used for both cut detection and gradual transition detection. Since the features learned from ICA can automatically adapt to data, the configurable parameters are expected to be more robust to different data, compared with those for manually selected features. In the new method, illumination-invariant chromaticity histogram from each video frame is created to form raw features. By performing ICA, two independent components (ICs) are generated and chosen as features. In the low dimensional ICA subspace, a dynamic clustering algorithm based on adaptive thresholding is developed to detect shot boundaries. A key-frame selection scheme based on the clustered video frames is also developed. Experimental results successfully show that the new method can effectively detect both abrupt transitions and gradual transitions.

4.2 Video Parsing and Indexing Using Independent Component Analysis

The new method has the following major steps: (i) Raw feature generation from illumination-invariant chromaticity histograms; (ii) ICA feature extraction; (iii) Dynamic clustering for shot detection. (iv) Video indexing based on the clustered video frames. Each step is described in the following subsections.

4.2.1 Illumination-invariant Chromaticity Histogram

Illumination changes and object/camera motion are the key factors that affect the performance of shot detection. Since histograms do not carry spatial information, they are expected to be robust to object and camera motion. However, histograms are generally sensitive to lighting changes. Therefore, in the new method, the normalized chromaticity histograms are chosen as raw features. Based on 3D RGB color space, the 2D illumination-invariant normalized chromaticity (r, g) is defined as [78],

$$r = R/(R+G+B), \quad g = G/(R+G+B). \tag{4.1}$$

Histograms with 256 bins are generated as features in the normalized chromaticity color space for each of the video frames. During implementation, only r component is used for simplicity. Thus, the dimension of raw feature vector is $n = 256$.

4.2.2 Feature Extraction Using ICA

ICA has been used for applications such as blind source separation, compression and denoising. In [47], the ICA model is used to extract basis functions from natural images. Such basis functions could be used as features since two different classes of images tend to have different basis functions. In the new method, the ICA model is applied in feature domain. Each video frame (raw feature vector) is processed as one observation that can be considered as a linear combination of hidden basis functions. Since the time course is only associated with the ICs, we select the most two significant ICs as the new features instead of the basis

functions. The temporal characteristics of ICs are explored by a clustering algorithm to detect shot boundaries.

For the i-th video frame, let $\boldsymbol{h}_i$ denote the raw feature vector created from the normalized chromaticity histogram. Using $\boldsymbol{h}_i$ as a column, the observed n-dimensional ($n = 256$) signal is constructed in matrix form as

$$\boldsymbol{O} = [\boldsymbol{h}_1 \, \boldsymbol{h}_2 \cdots \boldsymbol{h}_p], \tag{4.2}$$

where p is the total length of the video sequence. Each frame is represented as a column vector of $\boldsymbol{O}$. ICA learning method is performed to generate the filtering matrix W and the independent sources. We reduce the dimension and only keep the two most significant projecting directions ($M = 2$). The two-dimensional output ICs is given by the product of matrices W and $\boldsymbol{O}$. Thus, the data is projected onto an ICA subspace spanned by two basis functions. Each IC gives the coordinates for one projection direction. A video frame is represented by a point in the ICA subspace. The frames within one shot tend to form a compact cluster.

4.2.3 Dynamic Clustering for Video Shot Detection

Based on video frame distribution in the ICA subspace, a dynamic clustering algorithm is developed to classify video frames into shots and thus to detect the shot boundaries. Euclidean distance is used as dissimilarity measure between two points, $\boldsymbol{o}_i$ and $\boldsymbol{o}_j$ in ICA subspace, where i and j are time index.

$$d(\boldsymbol{o}_i, \boldsymbol{o}_j) = \|\boldsymbol{o}_i - \boldsymbol{o}_j\|_2. \tag{4.3}$$

Given the $(i+1)$-th sample $\boldsymbol{o}_{i+1}$, the sample mean vector $\boldsymbol{\mu}$ can be iteratively updated as

$$\boldsymbol{\mu}_{i+1} = \boldsymbol{\mu}_i + \frac{(\boldsymbol{o}_{i+1} - \boldsymbol{\mu}_i)}{i+1}. \tag{4.4}$$

Denote the sample variance vector by $\boldsymbol{\sigma}^2 = [\sigma^2_{(1)} \, \sigma^2_{(2)} \cdots \sigma^2_{(M)}]$ where the m-th element $\sigma^2_{(m)}$, $1 \leq m \leq M$, is the sample variance in the m-th dimension of the feature vector in ICA subspace. The m-th element of vector $\boldsymbol{\sigma}^2$ at time $(i+1)$ is iteratively updated as

$$\sigma^2_{i+1,(m)} = (\frac{i-1}{i})\sigma^2_{i,(m)} + (i+1)(\mu_{i+1,(m)} - \mu_{i,(m)})^2. \tag{4.5}$$

Due to camera motion and noise, intra-shot variations may cause the cluster center to gradually float away. In order to reduce the contributions from old samples, we introduce a decay factor α with $0 < \alpha \leq 1$. Denote the weighting vector by $\boldsymbol{w} = [\alpha^{i-1}\, \alpha^{i-2} \ldots \alpha^1\, 1]^T$. For any given vector sequence $X = [\boldsymbol{o}_1\, \boldsymbol{o}_2 \cdots \boldsymbol{o}_i]^T$, the weighted sample mean of these i samples can be calculated as

$$\boldsymbol{\mu} = \frac{\boldsymbol{w}^T \cdot X}{\|\boldsymbol{w}\|_1}. \tag{4.6}$$

If the sample size i is large, the weighted sample mean can be iteratively estimated as

$$\boldsymbol{\mu}_{i+1} = \alpha\boldsymbol{\mu}_i + (1-\alpha)\boldsymbol{o}_{i+1}. \tag{4.7}$$

In practice, (4.7) is used instead of (4.4) to calculate the cluster center. We still use (4.5) to approximate the sample variance, since we are not interested in estimating a true and unbiased weighted variance.

During clustering process, for a new sample $\boldsymbol{o}_{i+1}$, we calculate the distance between this new sample and the cluster center. The adaptive threshold τ_α is defined as

$$\tau_\alpha = \beta\|\boldsymbol{\sigma}^2\|_1, \tag{4.8}$$

where β is a predefined parameter to control how big the intra-shot variations are allowed. The new sample is classified into the current cluster if the following condition holds

$$d(\boldsymbol{\mu}_i, \boldsymbol{o}_{i+1}) < \tau_\alpha. \tag{4.9}$$

Then (4.7) and (4.5) are used to update the sample mean and sample variance. Otherwise, if the distance is larger than τ_α, we create a new cluster initialized with the sample $\boldsymbol{o}_{i+1}$. The time index of $\boldsymbol{o}_{i+1}$ is saved as a shot boundary. Since the condition adapts to the density of the points in ICA subspace, this mechanism essentially introduces an adaptive thresholding.

Two techniques are developed to improve performance. The first technique is outlier removal. If the distance between one sample and the current cluster center is larger than

τ_α, we check whether or not the next sample satisfies the condition (4.9). If the next sample can be classified into the current cluster, the previous sample is considered as an outlier and discarded. The other technique is to improve the performance for detecting gradual transitions. Once the recent samples are found to "move away" from the current cluster, a new cluster is formed. But this new cluster might be within the transition period when dealing with gradual transitions. A special property for those points within a transition period is that they are sparsely distributed in ICA subspace as shown in Figure 4.1. To capture this property, we use a temporal window of size $K(K = 30)$. Let J denote the average variation of sample variance within the temporal window. We define a measure of cluster compactness as

$$J = (\sum_{k=1}^{K-1} \|\boldsymbol{\sigma}_{k+1}^2 - \boldsymbol{\sigma}_k^2\|_1)/(K-1). \tag{4.10}$$

The above criterion is used to distinguish gradual transitions from cuts. If J is larger than a predefined threshold, a gradual transition is declared. Otherwise, the boundary is detected as a cut. It is worth mentioning that this evaluation is checked once at the beginning only when a new cluster is formed.

The clustering algorithm is summarized as follows:

- **Initialization**: Get the first $P(P = 5)$ samples and calculate the sample mean and sample variance directly. In extreme cases such as "freeze" frames, a minimum value τ_b is used to initialize the variance if the calculated sample variance is less than τ_b.

- **Iterative clustering**:

  1. Get a new sample and check condition (4.9).
  2. Update mean and variance by (4.7) and (4.5) if condition (4.9) is satisfied. Otherwise, check the outlier removal rule.
  3. Repeat step 1 and 2 until a sample can not be classified into the current cluster.
  4. Create a new cluster, and use (4.10) to check the boundary type, and set the new cluster as the current cluster.

4.2.4 Cluster-based Video Indexing and Summarization

After the clusters are obtained, the frames closest to the cluster centers are chosen as the key-frames. For each cluster (shot), one or several frames can be selected as key-frames. The number of key-frames per shot is determined by the level of intra-shot activities. To evaluate the compactness of the cluster, we introduce the *Fisher's discriminant ratio* (FDR) [55] as a measure. The FDR is originally designed for cluster separability. Let C_1 denote the first class with the sample mean vector $\boldsymbol{\mu}_1$ and sample variance vector $\boldsymbol{\sigma}_1^2$, and C_2 denote the second class with the mean vector $\boldsymbol{\mu}_2$ and sample variance vector $\boldsymbol{\sigma}_2^2$. As a measure of class compactness, we propose a modified FDR which is defined as

$$FDR(C_1, C_2) = \frac{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_1^2}{\|\boldsymbol{\sigma}_1^2 + \boldsymbol{\sigma}_2^2\|_1}. \tag{4.11}$$

For video indexing, in order to determine whether we need to choose multiple key-frames from one cluster, we iteratively divide the cluster into sub-clusters until some certain termination conditions are satisfied. The proposed FDR is used to measure the compactness and separability of the sub-clusters. The cluster-based video key-frame selection scheme is described as follows:

- **Initialization**: Choose a cluster as the current input; check the termination conditions, if any condition is satisfied, then exit.
- **Recursively dividing**:
  1. Temporally equally divide the current cluster into two sub-clusters.
  2. Calculate the FDR of the two clusters.
  3. Check the termination conditions.
  4. If the termination conditions are not satisfied, set the first sub-cluster as the current input and goto step 1.
  5. If the termination conditions are not satisfied, set the second sub-cluster as the current input and goto step 1.

6. If any termination condition is satisfied, choose the frame that is closet to the cluster center as the key-frame.

- **Termination conditions**:
  - If the number of frames in the current cluster is less than 60 frames (about 2 seconds for NTSC standard), then stop splitting.
  - If the FDR of the two sub-clusters is less than a given threshold, then stop splitting.

The value of the modified FDR is large when the variances of the clusters are small and the means are not close. That implies that we intend to select more key-frames if the current shot's separability is high. Essentially, we perform a binary splitting and divide each cluster (shot) into a tree-like structure. When the termination conditions are satisfied, we generate the key-frames at the leaf nodes. Therefore, the number of key-frames is equal to the number of leaf nodes of the binary tree.

4.3 Experimental Results

In the experiments, we have collected TV shows and documentary video sequences as the test data. The test video data is carefully selected to include as many effects as possible. The documentary video sequences contain many editing effects such as zoom-ins, zoom-outs, and camera panning. The experimental results are shown in Table 4.1. Precision and recall for cuts were obtained as 95% and 97.4% respectively. Precision and recall for gradual transitions were 85.7% and 89.3%. The false positives for TV shows were caused by fast camera motion. The water scenes in documentary video created some false positive for gradual transition detection. One gradual transition in documentary video joins two similar scenes. That was missed in our method. Even though we have chosen the difficult test data, the proposed method still had good performance. The results show that the algorithms are effective for both cut detection and gradual transition detection

The patterns for dissolves and hard cuts are shown in Figure 4.1 and 4.2. The distribution of all video frames for one video clip is shown in Figure 4.3. A key-frame selection based on the video indexing and summarization scheme described in section 4.2.4 is also validated. Figure 4.4 shows an example that four key-frames are selected for one video shot.

Table 4.1: Detection results for hard cuts (H) and gradual transitions (G).

Test Data	Total		Missed		False	
	(H)	(G)	(H)	(G)	(H)	(G)
TV Show	53	3	0	1	2	1
Documentary	64	44	3	4	4	6
Total	117	47	3	5	6	7

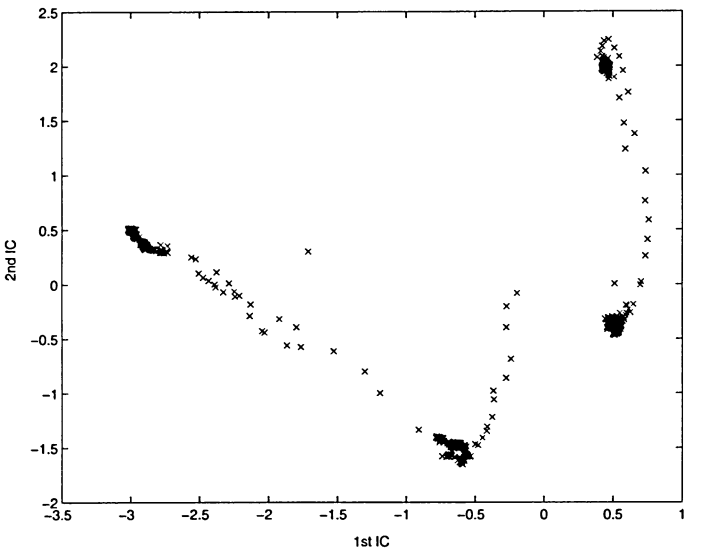


Figure 4.1: Cluster patterns formed during dissolves in the ICA subspace.

4.4 Comparison Between Online and Offline Methods

It is worth pointing out that the video parsing algorithm proposed in this chapter is different from the one proposed in previous chapter. The video parsing algorithm presented in

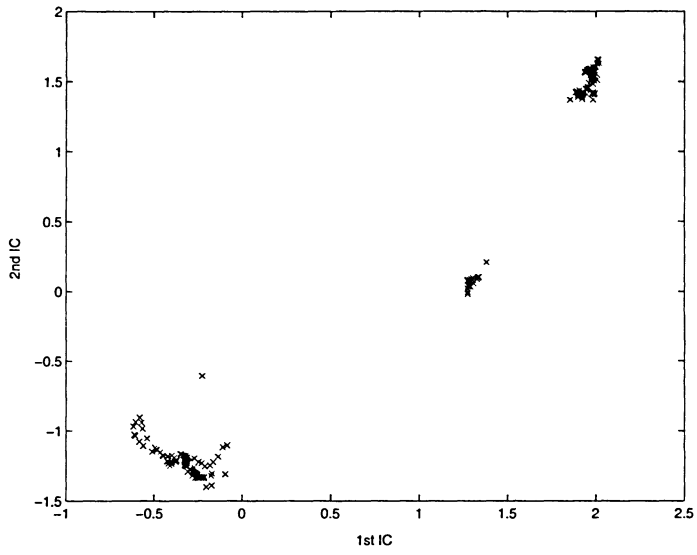


Figure 4.2: Cluster patterns for cuts in the ICA subspace.

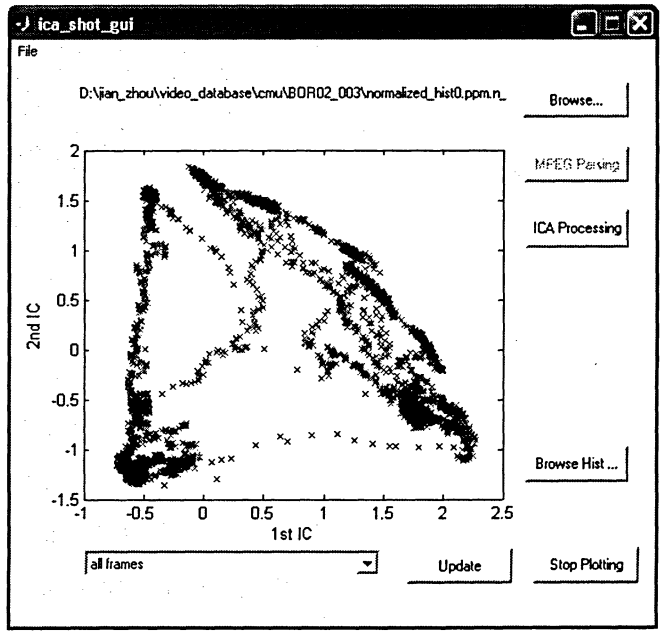


Figure 4.3: A video clip and its complete distribution in ICA subspace.

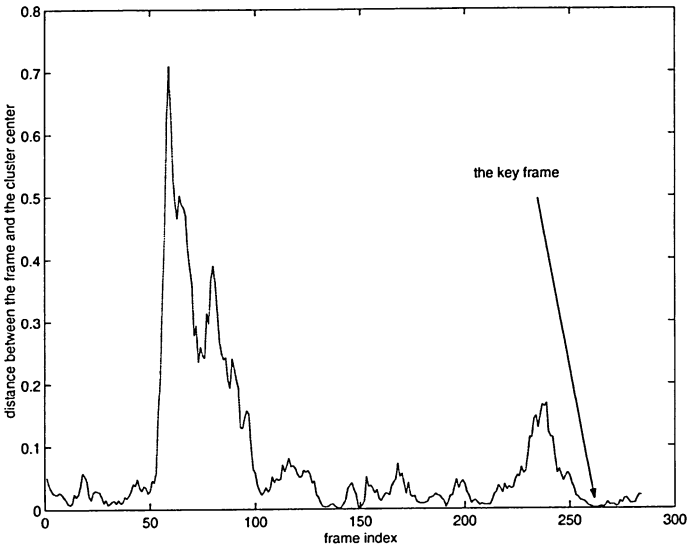


Figure 4.4: The frame closest (minimum distance) to the cluster center is selected as the key-frame.

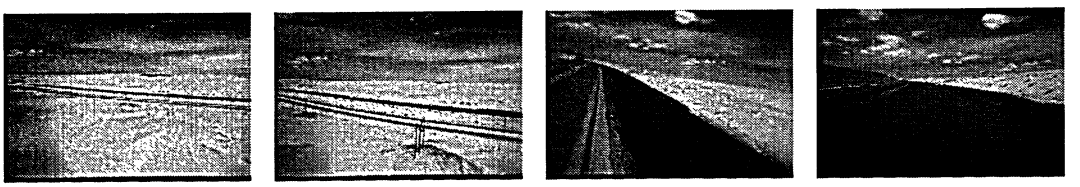


Figure 4.5: Multiple key-frame selection for one video shot.

Chapter 3 is based on basic statistical measures, and it is very fast to compute the features. Also, the method does not require all video frames to make decisions. However, for ICA based algorithm proposed in this chapter, all video frames are required to calculate the independent components. After the bases are obtained, each video frame is projected into the ICA subspace. Thus, the ICA based method can be considered as a batch mode processing. And furthermore, the extraction of features using ICA is not as fast as the basic statistical measures like mean, variance, and skewness. However, the fast online shot detection developed in Chapter 4 needs two passes to detect the boundaries since hard cuts and dissolve detectors use two different algorithms. Also, the mean-variance-skewness combined analysis

is designed for dissolve boundaries, other gradual transitions such as wipes and fades can not be identified. The ICA based method, on the other hand, can process the video frames in one pass, and it does not need to distinguish all the boundary types since a video shot is just described by an identified cluster. Therefore, gradual transitions such as wipes and fades can also be detected in ICA-based method. Also, the ICA based method can be used for video indexing and summarization since it is reasonable and easy to select key-frames from the detected clusters.

Another advantage of ICA based method is that the same feature space could be used for further analysis since the dynamics and the relationships at frame-level and shot-level are well preserved in ICA subspace.

4.5 Summary

In this chapter, a new statistical analysis method and a feature extraction technique using higher order statistics for video parsing and indexing are presented. Raw features are formed by normalized chromaticity histograms that are illumination-invariant. By performing ICA, two ICs are generated. Unlike typical image feature extraction using ICA, which uses basis functions as features, we choose ICs as features and explore their temporal characteristics. By projecting the high dimensional raw features into low dimensional ICA subspace, video shots are represented as separable compact clusters. A dynamic clustering algorithm using adaptive thresholding is developed to detect both cuts and gradual transitions at one pass. The simulations show that the method achieved good performance for detecting both abrupt transitions and gradual transitions. A video indexing scheme based on the clustered video frames is also developed, and the results show that selected key-frames are consistent with human perception.

Chapter 5

Video Dissimilarity Models

To locate a video clip in a large collection is very important for retrieval applications, especially for digital rights management. In this chapter, the statistical analysis method and the new feature extraction technique proposed in previous chapters are applied to video data to extract the shot-level features. And then a new technique for video dissimilarity measure is developed. This new algorithm is based on dynamic programming that fully uses the temporal dimension to measure the similarity between two video sequences. A normalized chromaticity histogram is used as a feature which is illumination-invariant. Dynamic programming is applied to shot-level to find the optimal nonlinear mapping between video sequences. Two new normalized distance measures are presented for video sequence matching. One measure is based on the normalization of the optimal path found by dynamic programming. The other measure combines both the visual features and the temporal information. Experimental results show that the shot-level approach is robust to frame rate conversion, color correction, and compressions. The proposed distance measures are suitable for variable-length comparisons.

5.1 Introduction

Content analysis of video is to extract meaningful information such that efficient classification, indexing, retrieval, and filtering are possible. One crucial step for such tasks is to define a similarity/dissimilarity measure between two video sequences. The common tech-

niques rely on key-frames since classical methods developed in content based image retrieval can be applied to these still-frames. In [29], a fast video signature based on randomized algorithms is proposed to approximate the video similarity defined as the percentage of clusters of similar frames shared between two video sequences. In [79], block-based minimum variances are used to create video hash values. However, temporal information is ignored in both of the above methods. A template-frequency model which makes uses of the temporal dimension is proposed in [80]. Another similarity measure between shots is developed by using dominant color histograms and spatial structure histograms [81]. In [82], the similarity between the query image and the video is defined as the distance between the query point and the linearly interpolated feature line. However, it is observed that video similarity measure is essentially a multiple-to-multiple matching process. For example, the query is not necessarily one key-frame or one shot. A query containing multiple frames or even multiple shots is also possible. One of the few research works that consider such a scenario is presented in [29]. However, the sequences are not treated as ordered sets since the frames are randomly sampled from video sequences. Therefore, the algorithm does not distinguish two sequences such as "AABBCC" and "CCBBAA". Also, if the video database contains many similar video sequences, the method proposed in [29] might not have enough discrimination abilities. Examples include sports video such as soccer video and football video. Note that key-frame based methods are not suitable for such query tasks since most of the scenes in those videos are very similar. Some other applications, such as digital rights management, also require quick identification of nearly the same content. Therefore, it is often necessary to incorporate order and temporal information. The desirable distance should be a proximity measure between two ordered sets.

In this chapter, we present a shot-level video similarity measure based on dynamic programming. Note the temporal information such as shot durations is not affected by frame rate conversion or illumination changes. The proposed method can be used to locate and identify a video sequence in large collections. Unlike the technique in [83] [84], where a frame-level dynamic programming is used to deal with frame misalignment, our new method

uses shot-level dynamic programming, where shot sequences are created in an illumination-invariant color space by clustering video frames in ICA subspace. In addition, two new normalized distances are introduced to calculate the dissimilarity. Optimal path is found by dynamic programming. The presented new method is robust to histogram processing, and frame rate conversion. The new distance measures are insensitive to the lengths of videos.

5.2 Video Dissimilarity Model

There is a growing concern about digital video piracy since the digital video content can be easily copied, edited, and redistributed with almost the same quality. Finding a specific video among large collections is very important for digital rights management applications. For example, a movie clip may be edited and converted to another file. Many specific attributes, such as frame rate, compression format, aspect ratio, color correction scheme, might have changed. During video editing, some inappropriate shots could be deleted and commercial breaks could be inserted. However, from human perception, we still regard them as the same content. Thus, in order to identify a specific video, an efficient video similarity method is required to identify the same content. Most existing similarity models are not suitable for such tasks since they either ignore the temporal dimension, or simplify the query model. In the presented method, new video similarity models based on dynamic programming are developed. We integrate both visual features and shot durations into dynamic programming framework, allowing variable-length comparison and partial matching.

5.2.1 Shot Detection

The first step is to segment a video into a shot sequences using a method in our previous work [85], where illumination-invariant chromaticity histograms are used as raw features and an ICA based method is used to convert the 256-dimensional raw feature subspace into a two dimensional feature space, in which a dynamic clustering algorithm is employed to cluster video frames into shots. Detailed information about ICA based shot detection is described in Chapter 4.

5.2.2 Shot-level Feature Extraction

The normalized chromaticity histogram is selected as a shot-level visual feature. The illumination invariant normalized chromaticity $(red, green)$ [78] is defined as: $red = R/(R+G+B)$, $green = G/(R+G+B)$. Histograms with 256 bins are generated in the normalized chromaticity color space for each frame of the video. In this work, only r component is used. Each shot is represented by a feature vector which is the mean vector of all video frames within the same shot. A shot sequence is then a vector sequence, $\{\boldsymbol{r}(i),\ i = 1, \ldots, N_R\}$, where $\boldsymbol{r}(i)$ represents the i-th shot and N_R is the total number of shots. Shot lengths (measured in time) are also calculated during feature extraction.

5.2.3 Dissimilarity Model

Let $R = \{\boldsymbol{r}(1)\,\boldsymbol{r}(2)\,\ldots\,\boldsymbol{r}(N_R)\}$ be a reference shot sequence of length N_R and $T = \{\boldsymbol{t}(1)\ \boldsymbol{t}(2)\,\ldots\,\boldsymbol{t}(N_T)\}$ be a test shot sequence of length N_T. In general, the number of shots in R is not equal to the number of shots in T, i.e. $N_R \neq N_T$. Denote the two alignment functions (shot index functions) by $p(\cdot)$ $(1 \leq p(i) \leq N_R)$ and $q(\cdot)$ $(1 \leq q(i) \leq N_T)$ for R and T respectively. The pair of alignment functions forms an ordered set which is defined as a path. The overall cost D is defined as

$$D = \sum_{i=1}^{N_p} d(\boldsymbol{r}(p(i)), \boldsymbol{t}(q(i))), \tag{5.1}$$

where N_p is the total length of the path and $d(\cdot)$ is the distance measure which needs to be carefully designed to measure the dissimilarity between two feature vectors. The optimization goal is to find the alignment functions $p(\cdot)$ and $q(\cdot)$ that minimize the overall cost D in (5.1).

To design a suitable distance measure $d(\cdot)$, the feature vectors need to be properly scaled or normalized such that all features contribute equally. Cosine measure is the cosine of the angle between two vectors. This measure captures a scale-invariant similarity. The distance function $d(\cdot)$ based on cosine measure is defined as

$$d(\boldsymbol{r}(i), \boldsymbol{t}(j)) = 1 - \frac{\boldsymbol{r}(i)^T \cdot \boldsymbol{t}(j)}{\|\boldsymbol{r}(i)\|_2 \cdot \|\boldsymbol{t}(j)\|_2}. \tag{5.2}$$

The dynamic programming can be employed for shot sequence comparison since the cost is additive. According to Bellman's optimality principle, we have the following recursive equation

$$D_{min}(p(i), q(i)) = \min_{p(i-1), q(i-1)} [D_{min}(p(i-1), q(i-1)) + d(p(i), q(i)|p(i-1), q(i-1))]. \quad (5.3)$$

Constraints including global constraints, local constraints, and end point constraints are given as

1. $p(1) = 1$, $q(1) = 1$;
2. $p(N_p) = N_R$, , $q(N_p) = N_T$;
3. $0 \leq p(i) - p(i-1) \leq 1$; $0 \leq q(i) - q(i-1) \leq 1$, $\forall i \geq 1$;
4. $p(i) - p(i-1) + q(i) - q(i-1) \geq 1$;

The constraints defined above guarantee the alignment paths are monotonically non-decreasing.

5.2.4 Normalized Distance Measure

The overall cost D can be used to measure the distance or dissimilarity between two video sequences. A desirable property for such a measurement is that the cost D should not depend on the lengths of the sequences. Therefore, a proper normalization of the total cost is necessary. For string matching, the problem has been addressed in [86] using *normalized edit distance*. However, it is computationally expensive. In practice, D/N_p can be used to calculate the distance measure with a certain amount of normalization. Our first new simplified normalization measure D_1 is defined as

$$D_1 = D_0/N_p, \quad (5.4)$$

where D_0 denotes the original total cost, i.e., $D_0 = D_{min}$, with D_{min} defined in (5.3).

For video sequence comparison, normalization of the total cost by the length of the path is essentially related to the number of shots since the length of the path is bounded between

$\max(N_R, N_T)$ and $(N_R + N_T)$. Note that one video sequence with more shots does not necessarily imply it is longer than the other. However, in terms of video similarity measure, people are often interested in how long the two video sequences "overlap" instead of how many shots (or key-frames) are similar. In another word, if we have two pairs of dissimilar shots, it is reasonable to penalize the longer sequences more, compared with the other pair with relatively shorter durations. Therefore, the *normalized edit distance* proposed in [86] cannot be directly applied here since it is only penalizes the lengths of sequences without considering the duration of each symbol. We present the second new distance measure to integrate both visual features and shot durations for video sequence comparison as follows

$$D_2 = \frac{\sum_{i=1}^{N_p}[d(\boldsymbol{r}(p(i)), \boldsymbol{t}(q(i))) \cdot |L_R(p(i)) - L_T(q(i))|]}{\sum_{i=1}^{N_p}(L_R(p(i)) + L_T(q(i)))}, \tag{5.5}$$

where $L_R(n)$ is the duration for n-th shot in R and $L_T(n)$ is the duration for n-th shot in T. It is easy to show that D_2 has an upper bound as follows

$$D_2 \leq \frac{\sum_{i=1}^{N_p} \max(L_R(p(i)), L_T(q(i)))}{\sum_{i=1}^{N_p}(L_R(p(i)) + L_T(q(i)))}. \tag{5.6}$$

This new distance measure D_2 combines both visual feature and time information. For applications that do not require strong temporal information, the distance measure D_1 can be used. While all the distances defined above can be used to measure the distance between two video sequences, the original total cost D_0 highly depends on the length of the path. For D_1, the values are within the range of $[0, 1]$ since the cost is normalized by the length of the path. However, to compare large video sequences, even if the two sequences are very dissimilar, the value of D_1 may still be very small because of the large length of the path. That makes it difficult to evaluate the variable-length comparisons or choose a suitable global threshold to identify videos. On the other hand, the value of distance measure D_2 is numerically stable and at the same time has good discrimination ability, as will be shown by the numerical results.

5.3 Experimental Results

To show the effectiveness of the proposed algorithm, a one-hour movie (drama) is arbitrarily captured as a test example. The same movie is obtained from two different sources and then encoded. Thus, we have two test data sets for the same movie. One video originally comes from the TV source, and then being MPEG-2 encoded in NTSC format with a frame rate of 29.97, frame size of 352x240, aspect ratio 4:3, and bit rate 3249kbps VBR. The video is manually divided into several smaller clips (A'-G') as reference videos. The other video originally comes from film source. We encode the video using MPEG-1 with a frame rate of 24, bit rate 1411 kbps, aspect ratio 16:9, and plus a simple color correction. This video is also manually divided into smaller clips A-G. Therefore, totally 14 video clips are used in our tests.

The same frame/scene from different sources are shown in Figure 5.1. As it can be seen that the pictures are different. The intensity histograms of the two pictures are shown in Figure 5.2. It can be observed that the lighting conditions and the editing effects have made the histogram from the TV source left-shifted, compared to the one from the film source. Because of different lighting conditions, the dissimilarity measure between these two histograms is very large. The normalized histograms we select, however, is illumination invariant. The normalized histograms for the two frames are shown in Figure 5.3. As it can be seen that the lighting changes are partially compensated since the curves are centered and normalized.

By applying shot detection algorithm on each clip, fourteen shot sequences are created. Dynamic programming is used to find the optimal alignment path. Three distance measures D_0, D_1, and D_2 are calculated and the results are shown in Table 5.1, 5.2, and 5.3 respectively. Though the values on the diagonal are relatively small for all three distances, Table 5.1 shows that D_0 does relate to the number of shots. For example, all values in the fourth row and the fourth column in Table 5.1 are relatively large. That is because both shot sequences D and D' have more shots than the others. In practice, for example, if we get a distance measure 1.30, we cannot decide if the same content has been identified since that number

might mean "similar" between long sequence comparisons but "dissimilar" between short sequence comparisons. The improved distance D_1 normalizes the cost by path. As we mentioned earlier, this measure solves the length problem, but is not consistent with human perception because temporal information is not considered. Hence, D_1 can be used when temporal information is not important for some applications. Table 5.3 shows the results for the proposed distance D_2. As it can be seen that the values are numerically stable (see (5.6)) and provide a consistent normalization. In practical applications, a global threshold could be easily selected to identify the video.

Table 5.1: Dissimilarity measure by the original total cost (D_0).

	A'	B'	C'	D'	E'	F'	G'
A	**0.87**	2.54	2.28	5.60	1.69	2.85	2.22
B	3.10	**1.18**	2.21	5.87	2.31	4.22	3.02
C	2.31	2.63	**0.42**	4.72	1.91	3.62	1.56
D	6.93	6.34	5.18	**1.30**	4.87	10.2	3.63
E	2.13	3.17	2.33	5.13	**0.17**	3.44	2.20
F	3.00	3.69	3.96	10.6	3.33	**0.13**	3.15
G	2.62	3.13	2.05	4.04	1.90	3.08	**0.49**

Table 5.2: Dissimilarity measure by the total cost normalized by path (D_1).

	A'	B'	C'	D'	E'	F'	G'
A	**0.22**	0.36	0.46	0.40	0.42	0.71	0.44
B	0.62	**0.17**	0.37	0.59	0.33	0.60	0.43
C	0.46	0.44	**0.07**	0.34	0.32	0.60	0.22
D	0.53	0.63	0.37	**0.09**	0.44	0.73	0.26
E	0.53	0.45	0.39	0.47	**0.04**	0.86	0.44
F	0.75	0.53	0.66	0.76	0.83	**0.04**	0.63
G	0.65	0.45	0.34	0.29	0.38	0.62	**0.10**

In the second experiment, we further specifically show that the proposed dissimilarity

Table 5.3: Dissimilarity measure by the total cost incorporated with shot duration (D_2).

	A'	B'	C'	D'	E'	F'	G'
A	**0.01**	0.10	0.37	0.35	0.23	0.08	0.41
B	0.43	**0.04**	0.30	0.15	0.20	0.22	0.26
C	0.36	0.23	**0.02**	0.25	0.22	0.09	0.54
D	0.33	0.09	0.25	**0.04**	0.25	0.11	0.50
E	0.20	0.15	0.21	0.22	**0.01**	0.07	0.32
F	0.07	0.27	0.07	0.10	0.06	**0.01**	0.09
G	0.31	0.26	0.33	0.54	0.34	0.09	**0.01**

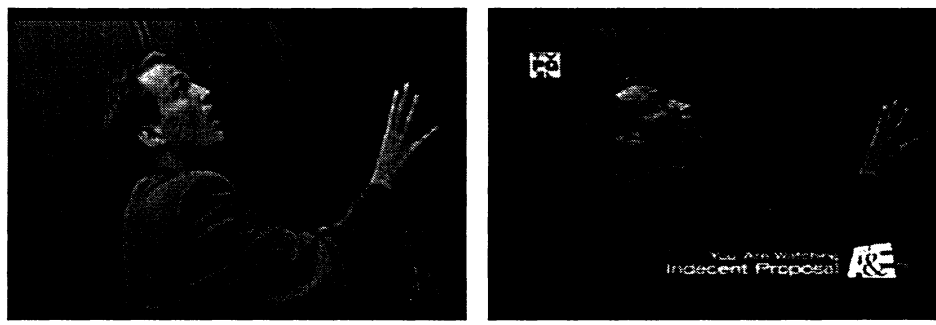
measure especially the distance D_2 is effective to correctly evaluate the variable-length comparisons when other distances fail. We select a video segment A' from the reference video as the query, and four video segments A_s, A_l, B_s, and B_l from the test video as our test dataset. Note that a test video and a reference video differ in frame-rate, aspect ratio, and lighting conditions. The video clips are not of the same length and each contains different numbers of shots. Semantically, the test clip A_s is a subset of A', and A' is a subset of A_l, while B_s and B_l have no overlapping with A'. We use the null set symbol ϕ to denote this no overlapping relationship. A_s and B_s are smaller video clips, compared with A_l and B_l. The relationships between the test dataset and the query video A' are listed in the second column in Table 5.4. The dissimilarity measures between A' and each of the clips in the test dataset are computed and listed in Table 5.5. Intuitively, A' should have relatively small distances with A_s and A_l, but large distances with B_s and B_l. However, as it can be seen in Table 5.5, the results show that D_0 and D_1 cannot reflect the true semantic relationship, since the measures are affected by the length and the number of shots. But D_2 is still able to identify that A_s and A_l are more similar to A', compared with others.

Table 5.4: Data used for the second experiment.

	number of shots	relationship with A'
A'	19	
A_s	14	$A_s \subset A'$
A_l	43	$A' \subset A_l$
B_s	10	$A' \cap B_s = \phi$
B_l	30	$A' \cap B_l = \phi$

Table 5.5: Similarity measure results for the second experiment.

	D_0	D_1	D_2
A' *vs.* A_s	2.112	0.111	0.067
A' *vs.* A_l	11.094	0.258	0.179
A' *vs.* B_s	5.413	0.257	0.398
A' *vs.* B_l	12.747	0.425	0.342



(a) (b)

Figure 5.1: The same frame in different sources (a) film source (b) TV source.

5.4 Summary

In this chapter, we present a novel technique to identify video clips based on the statistical analysis and the new feature extraction technique proposed in previous chapters. The

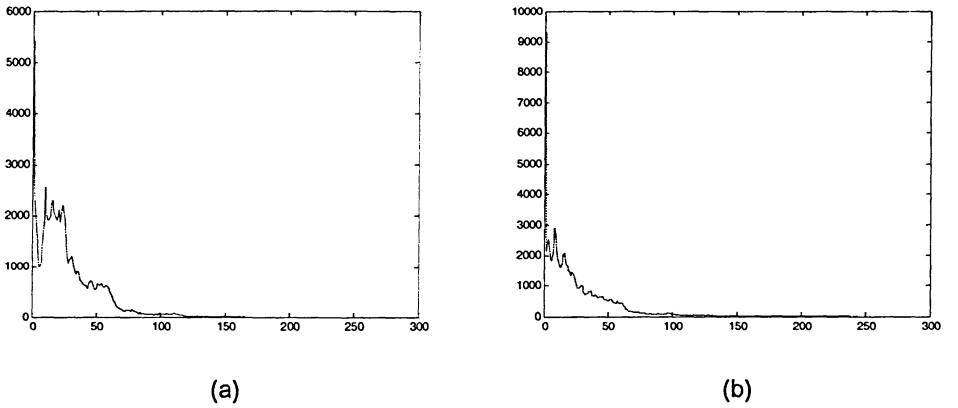


Figure 5.2: The histograms of the same frame (see Figure 5.1) (a) film source (b) TV source.

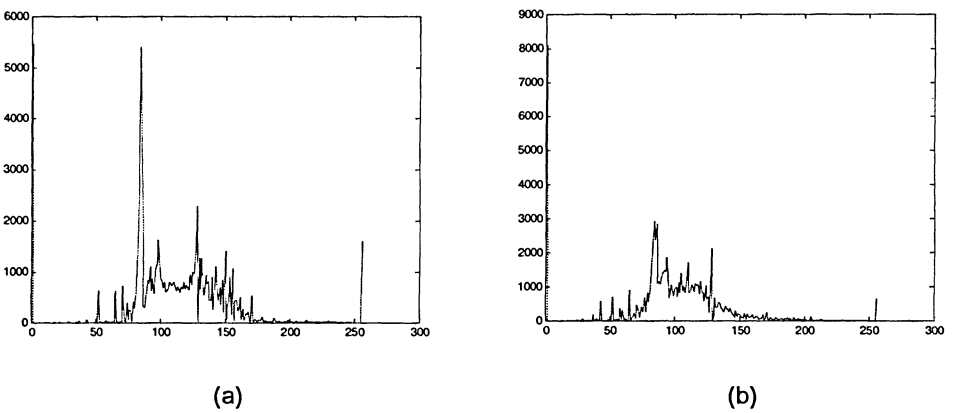


Figure 5.3: The normalized histograms of the same frame (see Figure 5.1) (a) film source (b) TV source.

algorithm operating on shot-level fully makes use of the temporal information. A video similarity model which combines both visual features and shot durations is presented. The nonlinear optimal mapping between the reference video and the test video is achieved by using dynamic programming. Experimental results show that the method is robust to frame rate conversion, histogram level editing, and compression format. In the presented framework, we develop new distance measures for video sequence comparison. The proposed video distances are numerically stable and consistent with human perception. Other potential ap-

plications and our future work include content based video retrieval among collections with high similarities and commercial breaks detection.

Chapter 6

Video Object Segmentation and Tracking

Automatic video object segmentation and tracking is a challenging problem. Statistical-based solutions using probabilistic fuzzy c-means and Gibbs Random Fields are investigated in this chapter for video object segmentation and tracking. The spatial segmentation is based on the probabilistic fuzzy c-means clustering and Gibbs sampling. The obtained segmented mask is then refined by taking into account of motion information. Motion vectors are calculated using block matching method based on phase correlation. The motion features and their spatial relationships are used to associate the segmented regions to form video objects. Temporal tracking is achieved by projecting the blocks in current frame to the next frame. The motion-compensated prediction is carried out directly over the membership matrix which is used as the initialization of probabilistic fuzzy c-means clustering for the next frame. Experimental results show that the proposed method can automatically extract and track the video object in a cluttered background. The major advantages of the proposed method are its ability to deal with deformable objects and being fully automatic.

6.1 Introduction

Analyzing spatial-temporal patterns is a fundamental research in digital video. One important characteristic of video is its temporal dimension. Traditional video coding standards, such as MPEG-1/MPEG-2, exploit the similarities between neighboring frames and reduce

the temporal redundancy by using block-based motion estimation methods. However, a human viewer does not view the video as a collection of rectangular blocks. Recently, partitioning video sequences into semantic video objects has been an active research area. The MPEG-4 [87] video standard introduces a framework for video object based coding. A video object may have arbitrary shape and may exist for an arbitrary length of time. The concept of video object not only allows more flexible options for video coding, it also supports high-level interpretation and manipulation of video contents. Applications to object-based video representation include video surveillance for security, video editing, animation, video conference, content-based video indexing and retrieval.

Automatic video object segmentation and tracking is difficult in that most sub-problems such as spatial segmentation, motion segmentation, occlusion, video object formation, appearance/disappearance of video objects and tracking of deformable objects are all non-trivial. Thus, a simplified formulation is often used among existing techniques. For example, the background is assumed to be static, or the system is semi-automatic such that the video object boundaries are already coarsely initialized by users. Many segmentation and tracking techniques have been proposed in literatures. Classical methods are mainly based on motion estimation and motion segmentation. In [2], image sequence is decomposed into layers by estimating and clustering affine parameters. Borshukov, *et al.* [88] improved this method by replacing adaptive K-means with a merging algorithm and implementing the block-based affine modeling in a multistage. In [33], a multi-resolution iterative refinement algorithm based on Kalman filtering was proposed. More recently, many researchers [34] [35] built their trackers on particle filtering framework since, in theory, particle filters can deal with non-linear and non-Gaussian estimations. Other methods based on mean-shift algorithm [89], spatial-temporal information [90], edge maps [91] were also developed to segment and track video objects.

Due to limitations of motion estimation, methods based on motion segmentation may not give accurate object boundaries. Active contours (i.e. snakes) have been widely used to track non-rigid objects. However, most motion-based techniques generally require user initializa-

tion and need additional models to process occlusion and de-occlusion. Spatial-temporal segmentation and tracking techniques consider both spatial and temporal information. Such techniques typically have a spatial segmentation step and a merging step based on motion features.

The new method presented in this chapter can be categorized into spatial-temporal in the sense we utilize the spatial features and temporal information in different stages. The new method aims at extracting and tracking deformable video objects and is fully automatic. The segmentation algorithm is based on probabilistic fuzzy c-means clustering with integration of Gibbs random fields that is employed to compute the local conditional probability as neighborhood constraints. During image segmentation, spatially connected pixels tend to belong to the same segment. However, this constraint is usually not well utilized in classical c-means or fuzzy c-means clustering techniques. In the new method, we bring Gibbs Random Fields into probabilistic fuzzy c-means framework to compute the local conditional probabilities as spatial neighborhood constraints. For motion segmentation, the block matching method using phase correlation is used to compute the temporal features. The segmented regions are analyzed and labeled to form video objects. Motion-compensated predictions are also applied to track and estimate the interested regions for the next frame. Experimental results show that the proposed method can automatically extract and track the video object in cluttered background.

6.2 Video Object Segmentation and Tracking

The new video segmentation and tracking method presented in this chapter includes the following steps: (i) spatial segmentation; (ii) motion segmentation; (iii) data association; (iv) temporal tracking. Motion segmentation and spatial segmentation are processed in different steps. Their results are analyzed and combined in data association step to define and label the video objects. The temporal tracking is introduced as the motion-compensated predictions of regions. Details are described in the following subsections.

6.2.1 Spatial Image Segmentation

As an important visual cue, the color features from the perceptually uniform CIE (Commission Internationale d'Eclairage) L*u*v color space are extracted in pixel domain. The L*u*v color space can linearize the perceptibility of color difference. Thus, the difference measured in Euclidean distance is consistent with the perceptual color difference viewed by human. For each pixel, a three-dimensional color feature vector is computed. Denote $\boldsymbol{y}_i = [y_i^{(L)}\, y_i^{(u)}\, y_i^{(v)}]^T$ as the color feature vector for the i-th pixel, where $y_i^{(L)}$, $y_i^{(u)}$, and $y_i^{(v)}$ are L, U, and V components of pixel i in L*u*v color space, respectively. The pixels in each frame are quantized into N (the number of clusters) colors according to the rule proposed in [92]. During spatial segmentation process, only color features are used. The spatial color segmentation process is based on probabilistic fuzzy c-means framework and Gibbs sampling.

Probabilistic Fuzzy C-means Clustering

Fuzzy c-means clustering techniques are generalized in [93]. In standard fuzzy c-means clustering, denote $\boldsymbol{y}_k$ by the color feature vector for the k-th pixel, and given the image of N pixels, i.e., $\boldsymbol{Y} = [\boldsymbol{y}_1\, \boldsymbol{y}_2 \cdots \boldsymbol{y}_N] \subset \Re^n$, the algorithm aims at finding a fuzzy partition $\boldsymbol{U}$ of the N elements based on the following objective function [93]

$$J_{FCM}(\boldsymbol{U}, \boldsymbol{V}) = \sum_{k=1}^{N} \sum_{i=1}^{c} u_{ik}^m (d_{ik})^2, \tag{6.1}$$

where

$$d_{ik}^2 = \|\boldsymbol{y}_k - \boldsymbol{v}_i\|_\Sigma^2 = (\boldsymbol{y}_k - \boldsymbol{v}_i)^T \Sigma (\boldsymbol{y}_k - \boldsymbol{v}_i), \tag{6.2}$$

in which c is the number of classes, $m \in [1, \infty)$ is the weighting exponent which controls the amount of fuzziness, u_{ik} is the degree of membership of $\boldsymbol{y}_k$ to the class i, the three dimensional column vector $\boldsymbol{v}_i$ represent the center of cluster i, $\boldsymbol{V} = [\boldsymbol{v}_1\, \boldsymbol{v}_2 \cdots \boldsymbol{v}_c]$ is the matrix of cluster centers, Σ is a positive-definite weight matrix, and d_{ik} is the distance measure between sample $\boldsymbol{y}_k$ and cluster center $\boldsymbol{v}_i$. The partition matrix $\boldsymbol{U}$ is also called membership matrix.

A probabilistic fuzzy c-means clustering is introduced in [94]. The fusion of probabilistic and fuzzy information can be represented as

$$u_{ik}^{*} = u_{ik} \cdot p_{ik}, \tag{6.3}$$

where $k = 1, 2, \ldots, N$, and p_{ik} is the probability of data point k belonging to cluster i. Based on the above modification, the objective function becomes

$$J_{PFCM}(\boldsymbol{U}, \boldsymbol{V}) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik}^{*})^{m} \cdot (d_{ik})^{2}. \tag{6.4}$$

The cluster centers and the membership matrix can be iteratively updated as

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} (u_{ik}^{*})^{m} \cdot \boldsymbol{y}_k}{\sum_{k=1}^{N} (u_{ik}^{*})^{m}}, \tag{6.5}$$

$$u_{ik} = \frac{\sum_{i=1}^{c} u_{ik} \cdot p_{ik}}{\sum_{s=1}^{c} (d_{ik}/d_{sk})^{2/(m-1)}}. \tag{6.6}$$

In [94], a method based on indicator and *ordinary kriging* is proposed to calculate p_{ik}. Different from [94], we propose to bring Gibbs sampling into fuzzy c-means framework. Details are presented in the following section.

Proposed Algorithm Integrating Probabilistic Fuzzy C-means Clustering and Gibbs Sampling

Standard fuzzy c-means has been a popular technique for image segmentation [95]. However, the relationship between pixels in spatial domain is not well utilized. Adding the spatial constraints directly in the objective function could be one possible solution. The other direction is to relax the constraints $\sum_{i=1}^{c} u_{ik} = 1$ and include spatial probabilistic information. The probabilistic fuzzy c-means framework introduces another way to associate the probabilities with membership matrix. That makes it possible to incorporate additional constraints and prior knowledge into the learning process.

Markov random fields (MRF) is the two-dimensional extension of Markov models. An image in spatial domain can be modeled by Markov random fields since it captures the context-dependent relationship of pixels within a neighborhood area. According to [96], an

equivalent relationship is built upon between Markov random fields and Gibbs distributions. Thus the computationally more tractable Gibbs random fields can be used to compute the conditional probabilities.

To spatially segment the video frames, we proposed a new method which integrate the Gibbs Random Fields into probabilistic fuzzy c-means framework. The Gibbs sampler is used to compute the local conditional probabilities as local neighborhood constraints. Those probabilities are directly associated with membership matrix and updated at each iteration. In Gibbs sampling, the local conditional probability in spatial domain is of the form [97]

$$p(z(\boldsymbol{x}_i) \mid z(\boldsymbol{x}_j), \forall \boldsymbol{x}_j \neq \boldsymbol{x}_i) = Q_{\boldsymbol{x}_i}^{-1} \cdot \exp\{-\frac{1}{T} \sum_{C \mid \boldsymbol{x}_i \in C} V_C(z(\boldsymbol{x}) | \boldsymbol{x} \in C)\}, \tag{6.7}$$

where

$$Q_{\boldsymbol{x}_i} = \sum_{z(\boldsymbol{x}_i) \in \Gamma} \exp\{-\frac{1}{T} \sum_{C \mid \boldsymbol{x}_i \in C} V_C(z(\boldsymbol{x}) \mid \boldsymbol{x} \in C)\}, \tag{6.8}$$

in which $z(\boldsymbol{x}) \in \Gamma = \{0, 1, \ldots, L-1\}$ is a discrete-valued random field evaluated at location $\boldsymbol{x}$, C is a *clique* which consists of a single pixel or a set of pixels, $Q_{\boldsymbol{x}_i}$ is a normalizing constant such that probabilities sum up to 1, T is a parameter and also known as *temperature*, $V_C(\cdot)$ are functions of the states of the pixels in the cliques set. The exponent function during the implementation is chosen as

$$-\frac{1}{T} z_{ij} \cdot (\nu_1 + \nu_2 \cdot (z_{i-1,j} + z_{i+1,j}) + \nu_2 \cdot (z_{i,j-1} + z_{i,j+1})), \tag{6.9}$$

where $z_{i,j}$ is the class label at location (i, j), and the ν_1, ν_2 are constants that depends on the local configuration of on the cliques [97].

The new segmentation algorithm based on Gibbs sampler and fuzzy c-means are summarized as follows:

1. Set values for the number of clusters C, the weighing exponent m, the termination criterion ϵ., and the maximum iteration steps. The number of clusters is determined by the relative smoothness of the whole image according to the guideline in [92].

2. Initialize the membership matrix $\boldsymbol{U}$.

3. Remove the fuzzyness in the membership matrix by choosing the maximum element at each column, construct the image in pixel domain, and compute the local conditional probability p_{ik} using (6.7) which is based on Gibbs sampler.

4. Evaluate the current cluster centers according to (6.5).

5. Update the membership matrix according to (6.6).

6. Compare the current membership matrix and the one obtained in previous loop, if $\|\boldsymbol{U}^{t+1} - \boldsymbol{U}^{t}\| < \epsilon$ or maximum iteration steps are reached, then stop; otherwise, return to step 3.

7. Remove the fuzzyness in the membership matrix by choosing the maximum element in each column. The result is the segmentation mask.

Several parameters during implementation are chosen as follows: the termination condition $\epsilon = 1e - 5$; the temperature is chosen as 1.

Since the proposed spatial segmentation mainly depends on the colors, the learning process allows regions with any arbitrary shape to be detected as long as the interested neighboring regions are distinguishable by their colors, and thus makes it possible to segment and track de-formable objects.

After the spatial segmentation mask is obtained, each region is given a unique label. A median filter is then applied to fill isolated small holes

6.2.2 Motion Segmentation

In motion segmentation, the block matching method based on phase correlation [97] is used to compute the motion feature vectors. Note that as initial segmentations, motion segmentation and spatial segmentation are separately processed. To extract motion feature during motion segmentation, each video frame is partitioned into 16-by-16 non-overlapping blocks. Fourier transform is used to calculate the spectrum. The motion vectors can be found by locating the peaks in the phase-correlation function, since a translational shift in spatial domain results

in a phase change in spectrum domain. For each block, a two dimensional motion vector $v = [v_x, v_y]$, is obtained, which represent the translational shift along the horizontal and vertical directions. Block matching using phase correlation has some desirable properties. First, it is relatively insensitive to illumination changes since the shifts in the mean value do not affect Fourier phase. Second, it is computationally efficient, compared to other motion estimation techniques such as pixel-level optical flow estimation methods. Note that this method models the motions as two-dimensional shift between two image blocks. Therefore, complex motions such as rotational motions cannot be captured.

In addition, the texture feature is utilized during motion segmentation. The reliability of motion features depends on the variations within a block. For example, if there is no texture within an area, i.e., the color in that area is almost uniform, good matches can always be found even there are no motions in the blocks. Those large motion vectors should be considered as noises. To this end, we introduced a criterion which evaluates the texture within a block to validate or reject the motion features. During implementation, we choose the variance of the block to evaluate the amount of texture within a block. If the matched block contains little texture, the motion vector for this block is rejected (motion vectors are assigned zero). Otherwise, it is accepted.

After blocks with motions are identified, a post-filtering based on the image dilation and erosion is applied to absorb nearby neighboring blocks. Then the output is the motion segmentation mask.

Note that static background is not assumed. Otherwise, a simple technique such as the image difference can be used to directly identify the region of interest (ROI). In our work, motions are estimated over all the blocks. As long as the background motion is relatively smaller than that of ROI, the motion segmentation described above may still be able to identify the interested regions.

6.2.3 Data Association

The motion vectors obtained by block matching method are assigned to each region. Regions which show consistency in motions and are spatially connected are identified and grouped together to form semantic objects. This data association step is important since it bridges the gap between the low-level features and high-level semantics. There are two tasks in this step. The first task is to build semantic video objects from low-level segmented regions. The second task is to label the video objects and keep track of their labels over the time.

The video objects formation is achieved by combing the spatial and motion segmentation results. The motion vector for the block is assigned to each pixel within the block, and thus the motion feature is extended from block level to pixel level. Then, in each spatial segmented region, the summation of absolute values of motion vectors is calculated. A predefined threshold is used to select those candidate regions that show certain amount of motions. Other regions are removed and assumed as static regions. Then, a dilation operator is applied on motion segmentation masks. The output, combined with the candidate regions, is used to calculate the final output. The overlapped areas are identified as semantic video objects.

After video objects are detected, they are put in correspondence over time. We implemented a solution that uses positions (positions of the centroids of video objects) and normalized histogram to compute the dissimilarity between video objects. In the video object pool in previous frame, the one which has the minimum distance with the current video object is associated with each other and labeled as the same object.

6.2.4 Temporal Tracking

We consider temporal tracking as the motion-compensated predictions of the interested regions.

Temporal information is directly applied in membership matrix instead of pixel domain. The reason is that the membership matrix itself already contains all the information about how the image is partitioned into regions since the spatial segmentation is essentially the

removal of fuzzyness in the membership matrix. The advantage of performing tracking on membership matrix is that the matrix estimated by motion-compensated from current frame can be directly used as the initialization for the next frame. This strategy can reduce the learning time since the initial membership matrix for the next frame is already a close approximation of true membership matrix which will be learnt. The similar idea has been proposed in [98]. However, the motion features used in [98] is pixel-level optical flow motion estimation which is computationally demanding.

The motion vector for a block can be considered as the average motion of all pixels in that block. Therefore, we use the block motion vector to approximate the pixel motions. Denote $\boldsymbol{v}(x_1, x_2) = [v_x\, v_y]$ by the motion feature vector of the pixel at location (x_1, x_2) and denote $S^{(n+1)}$ by the estimated spatial segmentation result for the $(n+1)$ frame. $S^{(n+1)}$ can be obtained by

$$S^{(n+1)}(x_1, x_2) = S^{(n)}(x_1 - v_x, x_2 - v_y). \tag{6.10}$$

Then, the initialization of membership matrix for the next frame is given by

$$u_{ik} = 1 \quad if\, S^{(n+1)} = i, \tag{6.11}$$

$$u_{ik} = 0 \quad otherwise. \tag{6.12}$$

6.3 Experimental Results

The image sequence "Hall Monitor" is used to test the proposed video object segmentation and tracking method. The test data contains 298 frames and each frame is of size 353X240. The spatial and motion segmentation results are represented as masks which are superimposed to the original images for display purpose. Figure 6.1 and Figure 6.2 show the motion segmentation results for frame 31. It can be seen that motion feature alone can only produce very coarse boundaries. Figure 6.3 (a) and (b) show the spatial segmentation results based on probabilistic fuzzy c-means and Gibbs random fields. An example of spatial segmentation is shown in Figure 6.3 (b). As it can be seen that some background areas are mistakenly segmented into potential region of interests since their colors are very similar.

Such errors can be eliminated during data association step. Data association combines both spatial segmentation results and motion segmentation results. We first identify the spatial regions which show a certain amount of motions. An example is shown in Figure 6.3 (c). Then, motion segmentation masks are combined with the identified region to create the final segmented results. The temporal tracking through motion-compensated prediction of membership matrix is verified in the experiments. Figure 6.4 illustrates the tracking results for frame 23, 32, 36, and 42. As it can be seen that the tracking performs very well on the major target but the boundaries are not very accurate. The error is mainly caused by the estimation noise from motion segmentation.

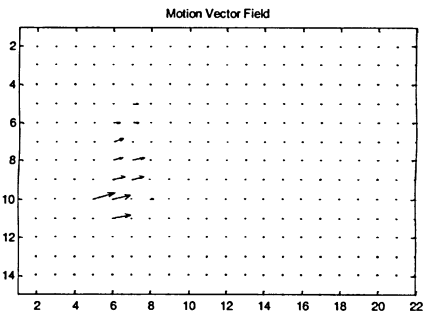


Figure 6.1: Motion vector field computed by block matching method using phase correlation.

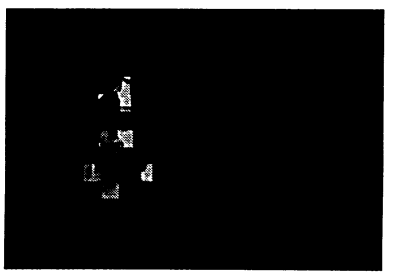


Figure 6.2: Motion Segmentation result.

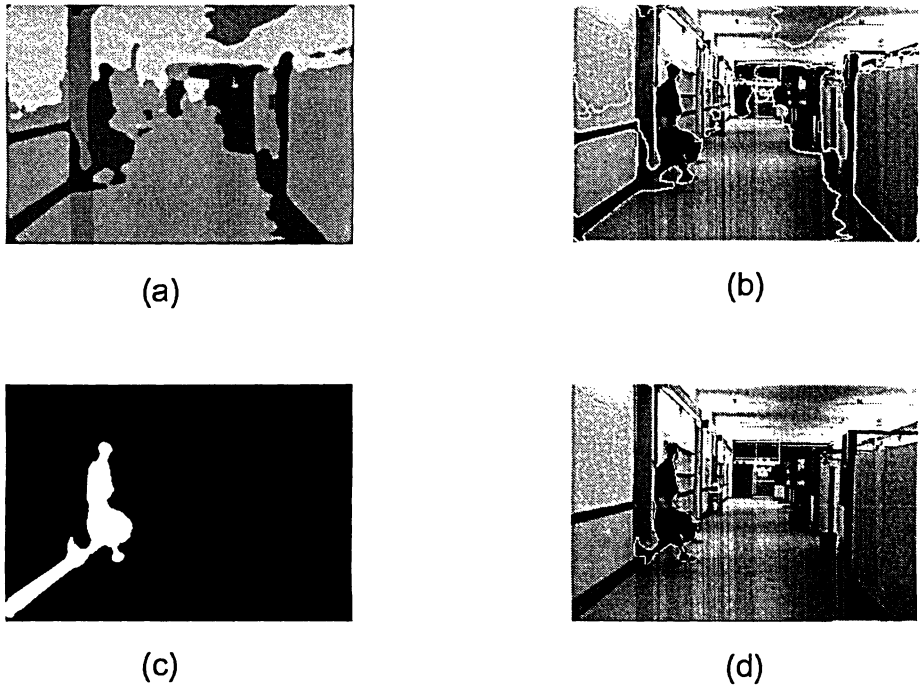


Figure 6.3: Spatial and motion segmentation result: (a) regions obtained from spatial segmentation. (b) spatial segmentation result. (c) spatial region that contains motions. (d) final results after data association.

6.4 Summary

In this chapter, we investigate the statistical modeling for localized feature extraction and analysis. We present a new fully automatic video segmentation and tracking method that combines probabilistic fuzzy c-means and Gibbs random fields. Color, motion and texture features are utilized together. In the spatial segmentation process, Gibbs sampling is integrated into probabilistic fuzzy c-means framework to compute the local conditional probabilities as spatial constraints. Motion segmentation is based on block matching method using phase correlation. In data association, motion segmentation masks and spatial segmentation masks are combined together to create video objects. The temporal tracking is performed for the motion-compensated prediction of membership matrix. The proposed

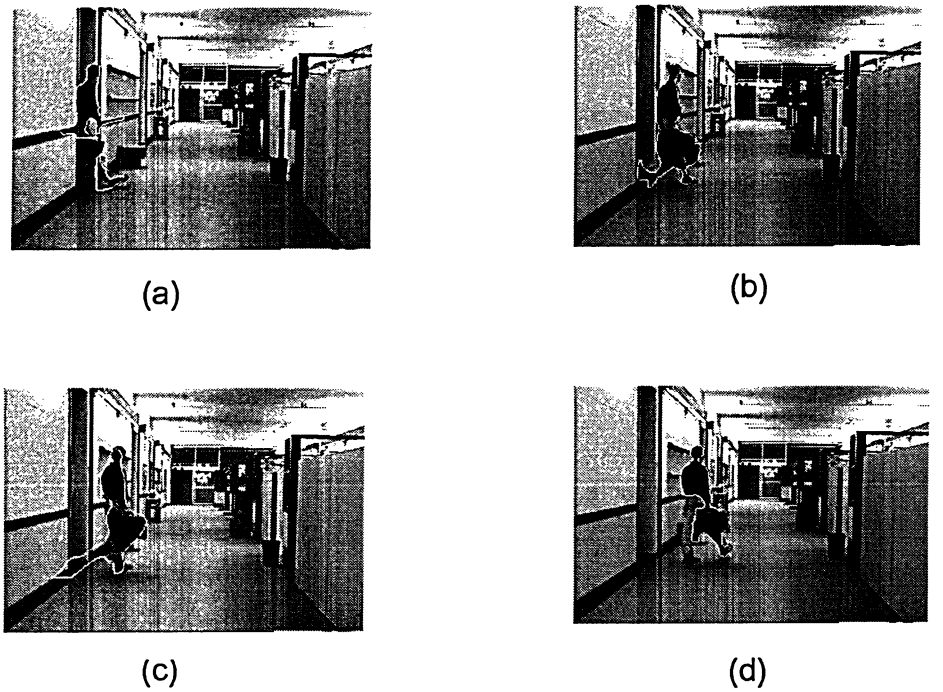


Figure 6.4: Tracking results at (a) frame 23 (b) frame 32 (c) frame 36 (d) frame 42.

method brings the probabilistic fuzzy c-means clustering into video object extraction and tracking, and integrated Gibbs random fields into the framework. The experimental results show that the proposed method can detect and track de-formable objects and being fully automatic. We note that complex motions such as rotational motions cannot be captured due to the limitation of the motion model. Future work will include such situations and the occlusion problem under this framework.

Chapter 7

Semantic Event Detection and Recognition in Videos

In this chapter, a new spatiotemporal statistical framework based on the Hidden Markov Model (HMM) and the Independent Component Analysis (ICA) mixture model is developed for content analysis of video. The observations of HMM are modeled as the mixture of non-Gaussians, and each non-Gaussian is associated with a standard ICA. The re-estimation formulas for model parameter learning are developed. The proposed new framework is application independent and can be applied to sequential data analysis. We apply the new framework to the video data for event detection and recognition. The video frames are first transformed into ICA subspace, and their coordinates in the subspace are considered as observations of the Markov process. In the ICA subspace, the ICA mixture model is used to estimate the observation distributions and to capture the spatial characteristics, and HMM is applied to explore the temporal characteristics of video frames. Note that ICA is used twice in our video detection algorithm. Firstly, we choose ICA as a feature extraction technique. Secondly, since ICA is well-known for its ability to estimate the non-Gaussian source densities, we choose the ICA mixture model as a density estimation method for the parametric representations of the distributions. We apply this statistical framework for event detection and recognition. The model parameters are trained by the training sequences. The likelihood is used to recognize and identify the video events. As a case study, golf video sequences are used to test the effectiveness of the proposed algorithm. The experimental

results show that the presented method can effectively detect and recognize the recurrent patterns in video data.

7.1 Introduction

Various techniques have been developed to analyze the content of the video data. Early works mainly focus on video transition detections [3], [15], [69]. Transitions or shot boundaries are detected such that post production of video can be recovered. After video shots are identified, one or several video frames are selected as key-frames to represent the video shots for indexing and retrieval purpose. For detailed literature review, please refer to Chapter 1 and 3. Besides the frame-level global features, the localized features and the object-based representation of video have also been investigated by researchers. In [99] and [100], the trajectories of objects based on object segmentation and tracking are used for video indexing. The object-based representations utilize both spatial and temporal information. However, such object-based analysis is often limited to the intra-shot analysis, and it only provides a very small time scale access for video.

For the past few years there has been an increased interest in semantic event detection and recognition from video. A semantic video event can be described by the low-level features. To bridge the semantic gap, new tools and models need to be developed. Several directions have been studied recently. One direction is to represent the high-dimensional video data in a compact representation, and thus make it possible to index, analyze and retrieve the elements efficiently. In [24], principal component analysis (PCA) is used to reduce the dimension of features of video frames, and two applications were demonstrated. One is the high-level scene analysis, and the other is the sports video classification. On the other hand, other researchers are utilizing new models to analyze the semantics from video. In [36], HMM modeling is used to detect play and break event for soccer video. Dominant color ratio and the magnitude of the motion vectors are used as features. The observations are modeled as a mixture of Gaussians with two mixtures per state. The sequences are segmented by computing the maximum likelihood to classify the video segments. In [101], HMM and audio

features are used to classify TV programs into commercial, basketball, football, news and weather video. In [39], MPEG-7 audio features and entropic prior HMM models are used to recognize common audio events such as applause and cheering.

Most existing HMM-based video analysis systems mainly focus on video classification tasks. In [36] and [38], hierarchical HMM structures are used to model the events. However, the tree-like hierarchical HMM is very complex by nature, and such structures may not be practical due to the computation burden. Also, the feature space and the pre-defined events are domain dependent and may not be generalized to other domains.

In this chapter, we develop a generic framework to detect and recognize semantic events. The task is to recognize and identify the known semantic events from video data. Note that finding a good feature space and representing the video data in an efficient way is crucial for recognition systems. In Chapter 4, we propose a compact feature space to represent video data based on ICA. The new representation makes it easier to analyze the dynamics and characteristics contained in video data. In the feature space, we develop a new statistical modeling by combining ICA mixture model and HMM modeling. Note that ICA techniques are used twice in our method. At the feature extraction step, ICA is used as a preprocessing filter to decompose the video signals. During the modeling step, ICA mixture model are integrated into the HMM framework to capture the spatial and temporal characteristics. We use HMM framework to grasp the temporal structures of video data since HMM is well-known for its capability to capture the temporal statistics of a stochastic process and it has already been widely and successfully used in the pattern recognition community. Simulation results show that these structures are good enough to model the semantic events. In our HMM modeling, each semantic event is described by one HMM model, and its parameters are learnt through the training sequences. The maximum likelihood criterion is used to evaluate how well an unknown video segment matches the model. Sequences with larger likelihoods are considered to be more likely to contain the pre-defined semantic events. As a case study, golf video sequences are used to test the effectiveness of the proposed algorithm. Differently from football, soccer, and tennis video, golf video has not been well analyzed in

the literature. The content analysis and the event detection in golf domain could provide potentials applications for home video and entertainment.

7.2 A Novel Framework of ICA Mixture Hidden Markov Model

7.2.1 Problem Formulation

For a video sequence with T video frames, we assume a feature vector of length L can be extracted from each frame. Let $\boldsymbol{o}_t\,(1 \leq t \leq T)$ be the feature vector for the t-th video frame. Each feature vector can be considered as one observation in the L-dimensional feature space. An event is defined as a video segment which has certain semantic meanings. In this chapter, we are only interested in the supervised learning techniques. Thus, the training sequences that define the events are known in advance. Let $\boldsymbol{E}_d\,(1 \leq d \leq D)$ denote a possible event where D is the total number of all possible events for a given video set. We assume a semantic event $\boldsymbol{E}_d$ can be described by an observation sequence, i.e.,

$$\boldsymbol{E}_d : \{\boldsymbol{o}_{z_d}, \boldsymbol{o}_{z_d+1}, \ldots, \boldsymbol{o}_{z_d+Z_d-1}\} \tag{7.1}$$

where $\boldsymbol{o}_{z_d}$ is the first frame for the event $\boldsymbol{E}_d$, and Z_d is the number of frames of the observation sequence. For a given video sequence, the objective of event detection and recognition is to first identify the event boundaries and then classify each video segments into one of the D possible known events. In the following sections, mathematical models are developed to represent the observation sequence. In our method, an event is described by model parameters. Equation (7.1) shows that our semantic events are defined on frame-level. To simplify the detection for event boundaries, we make the assumption that the beginning and ending frames of an event are located at shot boundaries. The assumption allows the shot-level information to be utilized for event boundary detections.

From the above problem formulation, it can be seen that the event detection and recognition consists of three major parts. The first part is to identify the event boundaries by using shot-level information for the given video sequence. The second part is the model

identification, i.e., semantic events are represented as model parameters using the training sequences. The third part is to compute the likelihood for each model given an unknown sequence. The event whose model parameters give the maximum likelihood is considered the identified event.

7.2.2 The ICA Mixture Hidden Markov Model

HMM framework [57] is chosen to model the temporal characteristics and the spatial distributions of video data in the feature space. Our motivations come from HMM's great success to model the temporal structure in sequential data, especially in the speech recognition field [57]. For video content analysis, the video sequence can also be considered as the sequential data. Many categories of video data, such as news and sports, often contain many temporal structures and recurrent patterns. That makes HMM a perfect tool to analyze the video content. In this chapter, we are interested in analyzing the video data that have a certain amount of recurrent patterns.

Modeling the Event Using HMM

As reviewed in Chapter 2, a classical discrete HMM model assumes the observations can be chosen from finite symbols defined as $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\}$ where M is the number of symbols. We denote the given observation sequence as $\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T$, where T is the length of the sequence and $\boldsymbol{o}_t, 1 \leq t \leq T$, is the L-dimensional feature vector for the t-th video frame. Let $q = q_1, q_2, \cdots, q_T$ be the hidden state sequence. A discrete HMM model with N states is determined by the parameters $\lambda = (A, B, \pi)$, where $A = \{a_{ij}\}, 1 \leq i, j \leq N$ is the state transition probability matrix, $a_{ij} = P(q_{t+1} = j \mid q_t = i)$ is the probability of state j at time $t+1$ given the state is i at time t, $B = \{b_j(k)\}$ is the observation symbol probability distribution for the discrete model, $b_j(k) = P(\boldsymbol{o}_t = \boldsymbol{v}_k \mid q_t = j), 1 \leq j \leq N, 1 \leq k \leq M$ is the probability of observing $\boldsymbol{v}_k$ given the current state is j at time t, and $\pi = \{\pi_i\}, 1 \leq i \leq N$ is the initial state distribution where $\pi_i = P(q_1 = i)$. Note that the model described above is a discrete model. For continuous observations, the symbol probability distribution $B = \{b_j(k)\}$ is replaced by $B = \{b_j(\boldsymbol{o})\}, 1 \leq j \leq N$, where $b_j(\boldsymbol{o})$ is the probability density function of the

observations at state j.

Under the HMM framework, different techniques can be used to model a semantic event. An event can be presented by the model structure using the supervised learning. Different event is represented by different model parameters. A semantic event can also be described by a hidden state. However, the states are "hidden" in nature, and generally there is no way to validate the "correct" state sequence. The choice of an optimality criterion is a strong function of the intended use for the uncovered state sequence [57]. Therefore, in our method, we use one HMM model to describe one event. The advantages of this method are that it is relatively robust to the small variations in training data, and it is generally not computationally demanding. For the discrete HMM, an HMM model is uniquely identified by its parameters $\lambda = \{A, B, \pi\}$, where $B = \{b_j(k)\}$ are symbol probabilities at each state. Because the distributions of video frames in our feature space are continuous, therefore it is advantageous to choose continuous densities to model the observations. In this chapter, we choose the continuous observation densities to avoid the errors introduced by quantization in discrete HMM. Under the continuous HMM framework, the parameters to describe the observations become $B = \{b_j(\boldsymbol{o})\}$, where $b_j(\boldsymbol{o})$ is the probability density function of the observations at state j. The probability density functions are generally represented in parametric forms. Thus, the whole parameter set for continuous HMM modeling to describe a set of semantic events $\boldsymbol{E}_d$ can be written as

$$\boldsymbol{E}_d : \lambda_d = (A_d, \Theta\{B_d\}, \pi_d), \quad 1 \leq d \leq D, \tag{7.2}$$

where D is the number of events, $\lambda_d = (A_d, \Theta\{B_d\}, \pi_d)$ are the HMM model parameters for the event $\boldsymbol{E}_d$. A_d is the transition matrix for the d-th event. π_d is the initial state distributions for the d-th event. B_d is the probability density function, and $\Theta\{B_d\}$ is the parameters set which uniquely determines the continuous observation densities in all states.

The identification of the model is to find the model parameters λ_d^* that gives the maximum likelihood,

$$LH_{\lambda_d}(\boldsymbol{O}) \equiv P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T \mid \lambda_d) = P(\boldsymbol{O} \mid \lambda_d), \quad 1 \leq d \leq D, \tag{7.3}$$

$$\lambda_d^* = \arg\max_{\lambda_d} LH_{\lambda_d}(\boldsymbol{O}), \quad 1 \leq d \leq D. \tag{7.4}$$

To apply HMM modeling techniques for video event detection, our interests include: 1) identification of model given known observations. 2) detection of an event given the model. The former is to adjust the model parameters to maximize the likelihood of the observations. The latter is to evaluate how likely the sequence is produced by the model.

Given the observation sequence $\boldsymbol{O}$, the description of the semantic event is essentially to build the model, and thus to estimate the model parameters. The Baum-Welch method can be used to estimate the model parameters λ such that the joint probability $P(\boldsymbol{O} \mid \lambda)$ is maximized. After convergence, an event described by $\boldsymbol{O}$ is represented by HMM model parameters. For the second problem, given the unknown observation sequence $\boldsymbol{O}'$, and a model $\lambda = (A, B, \pi)$, the likelihood $P(\boldsymbol{O}' \mid \lambda)$ determines how well the unknown observation sequence matches the given model. A larger likelihood implies the sequence is more likely to have the event described by the model parameters. The likelihood can be computed using *the Forward-Backward Procedure* [57]. For supervised learning, besides the HMM model parameters λ, the number of classes (i.e., the number of events) and the training sequences for each class still need to be determined.

Continuous Observation Densities Using Gaussian Mixture Model

Continuous observation densities can be used in HMM to avoid the quantization errors introduced by vector quantization in discrete HMM [57]. The continuous observation model has been formulated in [102], and the continuous densities based on the Gaussian mixture model have been formulated in [103]. Thus, for the Gaussian mixture observation model, the observation densities are of the form

$$b_j(\boldsymbol{o}) = \sum_{k=1}^{K} P(C_{jk}) \cdot p(\boldsymbol{o} \mid C_{jk}), \quad 1 \leq j \leq N, \tag{7.5}$$

where

$$p(\boldsymbol{o} \mid C_{jk}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_{jk}|^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{o} - \boldsymbol{\mu}_{jk})^T \boldsymbol{\Sigma}_{jk}^{-1}(\boldsymbol{o} - \boldsymbol{\mu}_{jk})). \tag{7.6}$$

The multivariate variable $\boldsymbol{o}$ can be considered as the observation vector being modeled, K is the number of mixtures, C_{jk} is the k-th mixture component in state j, $P(C_{jk})$ is the

probability of choosing the k-th mixture component in state j, $p(\boldsymbol{o} \mid C_{jk})$ is a Gaussian density with mean vector $\boldsymbol{\mu}_{jk}$ and covariance matrix $\boldsymbol{\Sigma}_{jk}$ for the k-th mixture component in state j. The structure of continuous HMM based on Gaussian mixture model is shown in Figure 7.1.

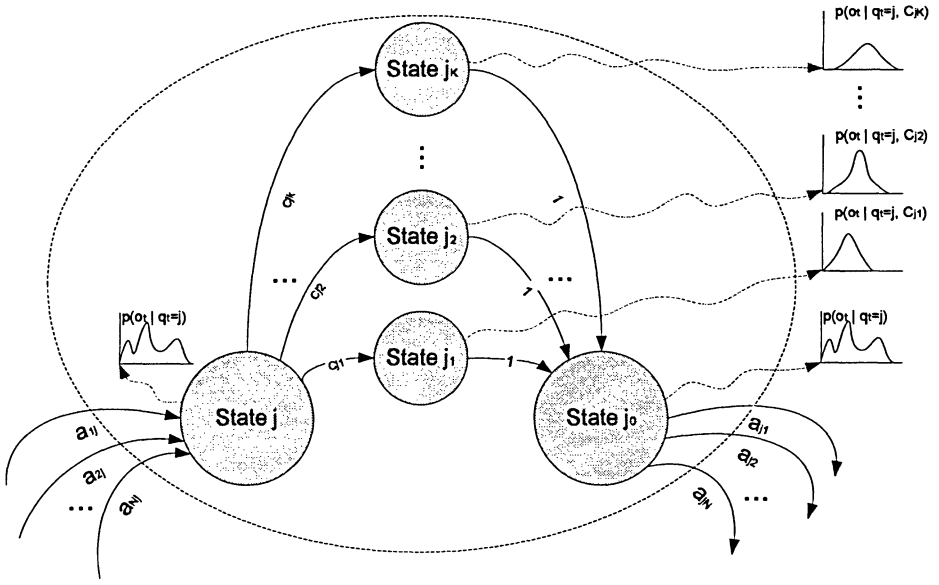


Figure 7.1: Equivalent K state re-configuration for state j with K Gaussian mixtures.

Continuous Observation Densities using non-Gaussian Mixture Model

Even though the Gaussian mixture model can approximate arbitrarily closely any continuous density function for a sufficient number of mixtures, the results may highly depend on how the number of mixtures is chosen and how the parameters are estimated. In this chapter, we introduce a new HMM observation model using non-Gaussian mixtures. Each non-Gaussian mixture is associated with a standard ICA. Thus we called this new continuous observation model as the non-Gaussian mixture observation model or the ICA mixture observation model for HMM. The reason for using a mixture of non-Gaussians is because the distribution of video frames in the feature space generally shows non-Gaussian charac-

teristics, and such higher order statistics can be captured by ICA blindly. We choose *ICA mixture model* instead of *ICA* because the observed video data can be categorized into mutually exclusive classes, similarly like Gaussian mixture models. Such characteristics are often true in video data since the separate stories are often interlaced in the video sequences. The ICA mixture model is first to divide the observed data into mutually exclusive classes, and then model each class as a linear combination of independent, non-Gaussian sources. This allows modeling classes with non-Gaussian structures. The observation densities described by ICA mixture can model a broader range of probability density functions, and can be considered as a complement of Gaussian mixture modeling. When using ICA mixture to capture non-Gaussian structures and classify the data, better results were reported in [104], compared with Gaussian mixture.

To choose ICA mixture as HMM observation model, we are not really interested in the properties of the recovered sources or their physical meanings. Our major concern is to estimate the observation densities based on ICA mixture learning algorithms and the source models we selected. The goal is to develop a parametric form to represent the observation densities and then derive HMM learning algorithms for HMM models.

Let q_t be the hidden state at time t, the proposed non-Gaussian mixture observation model that brings ICA mixture model into HMM framework to capture the non-Gaussian structures can be represented as follows

$$b_{q_t}(\boldsymbol{o}) = \sum_{k=1}^{K} p(\boldsymbol{o} \mid C_{q_t k}, \theta_{q_t k}) \cdot P(C_{q_t k}), \tag{7.7}$$

where $\boldsymbol{o}$ is the vector being modeled. $P(C_{q_t k})$ is the class probability to the k-th class. $p(\boldsymbol{o} \mid C_{q_t k}, \theta_{q_t k})$ is a non-Gaussian probability density function that describes the statistics of the observations for the k-th class at time t, given the state at time t is q_t, where $\theta_{q_t k}$ represents the parameters of the densities in state q_t. Note that (7.7) is very similar to (7.5) since both are essentially mixture models. The only difference between (7.7) and (7.5) is that the mixture component $p(\boldsymbol{o} \mid C_{q_t k}, \theta_{q_t k})$ in (7.7) is a non-Gaussian density function instead of a Gaussian density. The challenges and difficulties reside in the inferring the non-Gaussian

density analytically. However, the non-Gaussianities can be captured by ICA by seeking statistically independent sources. Thus, each non-Gaussian mixture component density in (7.7) is further modeled as a standard ICA. Therefore, the continuous observation model using non-Gaussian mixture model is essentially a ICA mixture model.

The non-Gaussian distributions can be further decomposed into functions through the standard ICA as follows.

In classical ICA without considering the mixture modeling, the observation sequence $\boldsymbol{O} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T$ is modeled as an L-dimensional random variable $\boldsymbol{o}_t$ which is further modeled as a linear combination of L statistically independent sources $\boldsymbol{s}_t$ plus the bias $\boldsymbol{\mu}$, i.e.:

$$\boldsymbol{o}_t = \boldsymbol{M}\boldsymbol{s}_t + \boldsymbol{\mu}, \quad t = 1, \ldots, T. \tag{7.8}$$

where $\boldsymbol{M}$ $(L \times L)$ is known as the *mixing matrix* in other ICA literatures. As mentioned in Chapter 2, to avoid the ambiguity of the terms, in this thesis we always refer to $\boldsymbol{M}$ as *the basis matrix* to distinguish the word "mixture" in the mixture model. The ICA task is to find the filter matrix $\boldsymbol{W} \approx \boldsymbol{M}^{-1}$ using only the observed signals $\boldsymbol{O}$. Since the observed signals are a linear transformation of the sources, their multivariate probability density functions satisfy the following relationship:

$$p(\boldsymbol{o}) = \frac{p(\boldsymbol{s})}{|\boldsymbol{J}|} \tag{7.9}$$

where $|\cdot|$ denotes the absolute value and $\boldsymbol{J}$ is the Jacobian of the transformation determined by the basis matrix. Therefore, the log likelihood can be written as:

$$\log p(\boldsymbol{o}) = \log p(\boldsymbol{s}) - \log(\det|\boldsymbol{M}|). \tag{7.10}$$

Equation (7.8)-(7.10) describe the case when all the observations are assumed to be generated from one class (i.e. $K = 1$). The observation model introduced in (7.7) requires a mixture modeling since the observations are assumed to be generated from multiple classes. The standard ICA can be generalized into ICA mixture model that allows modeling of classes with non-Gaussian structures. The ICA mixture model, originally proposed by Lee and Lewicki [104], assumes that the observed data $\boldsymbol{O} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T$ are drawn independently

and generated by a mixture density model. The likelihood of the data is given by the joint density:

$$p(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T \mid \Theta) = \prod_{t=1}^{T} p(\boldsymbol{o}_t \mid \Theta), \tag{7.11}$$

The mixture density is:

$$p(\boldsymbol{o} \mid \Theta) = \sum_{k=1}^{K} p(\boldsymbol{o} \mid C_k, \theta_k) p(C_k) \tag{7.12}$$

where $\Theta = (\theta_1, \ldots, \theta_k)$ are the unknown parameters for each $p(\boldsymbol{O}_t \mid C_k, \theta)$. C_k denotes the class k and K is the number of classes. Then the data in each class are described by:

$$\boldsymbol{o}_t = \boldsymbol{M}_k \boldsymbol{s}_{k,t} + \boldsymbol{\mu}_k, \quad \boldsymbol{o}_t \in C_k, \tag{7.13}$$

where M_k is a square basis matrix for the k-th class, and $\boldsymbol{\mu}_k$ is the bias vector for class k. The task is first to classify the unlabeled data points into one of the K classes, and then to determine the parameters for each classes. The parameters include $(\boldsymbol{M}_k, \boldsymbol{\mu}_k)$ for the k-th class, and the class probability $p(C_k \mid \boldsymbol{o}_t, \theta)$ for each data point. Within each class, the data points are modeled by the standard ICA. Therefore, considering (7.10), we rewrite the total likelihood of the data based on the ICA mixture model as:

$$\log p(\boldsymbol{o} \mid \Theta) = \sum_{t=1}^{T} \log\Big(\sum_{k=1}^{K} p(C_k)(\log p(\boldsymbol{s}_{k,t}) - \log(\det|\boldsymbol{M}_k|))\Big). \tag{7.14}$$

The ICA mixture model described above can be integrated into HMM framework to model the observations. The motivation comes from the characteristics of video analysis. When doing video analysis, a video segment generally consists of several shots, and each shot can be considered as a class in the observation space. Classical Gaussian mixture models can be used to model the observations. However, because of intra-shot activities and camera motions, the distributions for each class rarely shows a Gaussian shape.

7.3 Algorithms for ICA Mixture Hidden Markov Model

7.3.1 Model Parameters

In the previous section, we integrate the ICA mixture model into HMM framework to model the observation densities as the mixture of non-Gaussians. We call this new proposed frame-

work as *ICA Mixture Hidden Markov Model* or ICAMHMM. The ICAMHMM framework has the following model parameters:

$$\lambda = (A, C, \mu, M, \pi), \tag{7.15}$$

where $A = \{a_{ij}\}, 1 \leq i, j \leq N$ is the transition matrix. $C = \{P(C_{jk})\}$, $1 \leq j \leq N$, $1 \leq k \leq K$ is the mixture matrix. $\mu = \{\boldsymbol{\mu}_{jk}\}$, $1 \leq j \leq N, 1 \leq k \leq K$ is the mean coefficients for the mixture densities, where $\boldsymbol{\mu}_{jk}$ is the L-dimensional mean vector for the k-th mixture component at state j. $M = \{\boldsymbol{M}_{jk}\}$, $1 \leq j \leq N, 1 \leq k \leq K$ is the ICA basis coefficients, where $\boldsymbol{M}_{jk}$ is the $L \times L$ basis matrix for the k-th ICA source at state j. $\pi = \{\pi_i\}, 1 \leq i \leq N$, is the initial state distribution. Note that C, μ, and M are essentially the parameters for the modeling of the observation distributions in parametric form.

The identification of the model is to infer the parameters based on training data. In [103], an interactive procedure to update parameters based on Gaussian mixtures has been formulated and derived under the HMM framework. In the previous section, we formulate a new observation model which is based on non-Gaussian mixtures, and each non-Gaussian mixture component is associated with a standard ICA. In this section, the re-estimation formulas for model parameter learning of the ICAMHMM framework are derived.

In order to develop the updating rules for the model parameters λ in the ICAMHMM framework, we first review some definitions that are required for our derivations. Recall the forward variable $\alpha_t(i)$ and the backward variable $\beta_t(i)$ reviewed in Chapter 2

$$\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = i \mid \lambda), \tag{7.16}$$

$$\beta_t(i) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T, \mid q_t = i, \lambda). \tag{7.17}$$

Using the forward variable and the backward variable defined above, two probabilities of the joint event can be defined [57]:

$$\xi_t(i,j) \equiv P(q_t = i, q_{t+1} = j \mid \boldsymbol{O}, \lambda) = \frac{\alpha_t(i)\, a_{ij}\, b_j(\boldsymbol{o}_{t+1})\, \beta_{t+1}(j)}{P(\boldsymbol{O} \mid \lambda)}, \tag{7.18}$$

$$\gamma_t(i) \equiv P(q_t = i \mid \boldsymbol{O}, \lambda) = \frac{\alpha_t(i)\, \beta_t(i)}{P(\boldsymbol{O} \mid \lambda)}, \tag{7.19}$$

where (7.18) defines the probability of the joint event: a path passes through state i at time t *and* through state j at time $t+1$, given the available sequence of observations $\boldsymbol{O}$ *and* the parameters of the model λ. The (7.19) defines the probability of being in state i at time t, given the observation sequence $\boldsymbol{O}$, and the model λ.

In our non-Gaussian mixture observation model, we generalize the intermediate variable $\gamma_t(j)$ to $\gamma_t(j,k)$. The variable $\gamma_t(j,k)$ is defined as

$$\gamma_t(j,k) = \gamma_t(j) \cdot \frac{P(C_{jk}) \cdot p(\boldsymbol{o}_t \mid C_{jk}, \theta_{jk})}{\sum_{m=1}^{K} P(C_{jm}) \cdot p(\boldsymbol{o}_t \mid C_{jm}, \theta_{jm})}, \tag{7.20}$$

where $p(\boldsymbol{o}_t \mid C_{jk}, \theta_{jk})$ is the non-Gaussian observation probability density function $p(\boldsymbol{o} \mid C_{jk}, \theta_{jk})$ evaluated at $\boldsymbol{o}_t$. The term $\gamma_t(j,k)$ can be interpreted as the probability of being in state j at time t with the k-th mixture component accounting for $\boldsymbol{o}_t$.

7.3.2 Re-estimation Algorithms

Proof of Convergence

To derive ICAMHMM updating rules, we introduce the likelihood function (as defined in (7.3)) and partition the likelihood function over the state space. First, we denote the joint likelihood of observations and the state sequence as

$$LH_\lambda(\boldsymbol{O}, q) \equiv P(\boldsymbol{O}, q \mid \lambda). \tag{7.21}$$

Thus, the partition of the likelihood function over the state space is represented as a sum over the set, ℓ, of all possible state sequences q

$$LH_\lambda(\boldsymbol{O}) = \sum_{q \in \ell} P(\boldsymbol{O}, q \mid \lambda) = \sum_{q \in \ell} LH_\lambda(\boldsymbol{O}, q). \tag{7.22}$$

Note that the posterior likelihood defined in (7.22) is over all possible sequences of states. The objective is to maximize $LH_\lambda(\boldsymbol{O})$ over all parameters λ. For a particular state sequence $q = q_1, q_2, \ldots, q_T$, the probability $P(q \mid \lambda)$ is:

$$P(q \mid \lambda) = \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1} q_t}. \tag{7.23}$$

By substituting (7.23) into (7.22), the likelihood function can be rewritten as:

$$LH_\lambda(\boldsymbol{O}) = \sum_{q \in \ell} P(\boldsymbol{O}, q \mid \lambda) = \sum_{q \in \ell} P(\boldsymbol{O} \mid q, \lambda) \cdot P(q \mid \lambda) = \sum_{q \in \ell} \pi_{q_1} \cdot \prod_{t=1}^{T-1} a_{q_t q_{t+1}} \cdot \prod_{t=1}^{T} b_{q_t}(\boldsymbol{o}_t). \tag{7.24}$$

Note that $b_{q_t}(\boldsymbol{o}_t)$ is the observation probability density function $b_{q_t}(\boldsymbol{o})$ (see Equation 7.7) evaluated at $\boldsymbol{o}_t$, given the state at time t is q_t. Recall (7.7), we model $b_{q_t}(\boldsymbol{o})$ as the ICA mixture models

$$b_{q_t}(\boldsymbol{o}_t) = \sum_{k=1}^{K} p(\boldsymbol{o}_t \mid C_{q_t k}, \theta_{q_t k}) \cdot P(C_{q_t k}), \tag{7.25}$$

where $p(\boldsymbol{o}_t \mid C_{q_t k_t}, \theta_{q_t k_t})$ is the non-Gaussian probability density function $p(\boldsymbol{o} \mid C_{q_t k_t}, \theta_{q_t k_t})$ evaluated at $\boldsymbol{o}_t$. Based on the above representation, the likelihood function can be further partitioned by choosing a particular classification sequence, $\boldsymbol{K} = k_1, k_2, \ldots, k_T$, of mixture densities, where the values of k_t $(1 \leq t \leq T)$ can be chosen from $\{1, 2, \ldots, K\}$. The mixture sequence $\boldsymbol{K}$ determines which class each observation belongs to. We denote the set of all possible mixture sequences as $\hbar$. By definition we have the following partitions for the likelihood function:

$$LH_\lambda(\boldsymbol{O}) = \sum_{q \in \ell} LH_\lambda(\boldsymbol{O}, q) = \sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} LH_\lambda(\boldsymbol{O}, q, \boldsymbol{K}), \tag{7.26}$$

where $LH_\lambda(\boldsymbol{O}, q, \boldsymbol{K})$ is the joint likelihood of $\boldsymbol{O}, q, \boldsymbol{K}$ for some particular mixture sequence $\boldsymbol{K} \in \hbar$

$$LH_\lambda(\boldsymbol{O}, q, \boldsymbol{K}) = \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t q_{t+1}} \prod_{t=1}^{T} p(\boldsymbol{o}_t \mid C_{q_t k_t}, \theta_{q_t k_t}) \cdot P(C_{q_t k_t}). \tag{7.27}$$

Thus, a complete representation of the likelihood function is represented as

$$LH_\lambda(\boldsymbol{O}) = \sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t q_{t+1}} \prod_{t=1}^{T} p(\boldsymbol{o}_t \mid C_{q_t k_t}, \theta_{q_t k_t}) \cdot P(C_{q_t k_t}). \tag{7.28}$$

To apply the maximum likelihood estimation, we define the auxiliary function (the Q-function) with similar form as [102] [103]:

$$Q(\lambda, \lambda^*) = \sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} LH_\lambda(\boldsymbol{O}, q, \boldsymbol{K}) \cdot \log LH_{\lambda^*}(\boldsymbol{O}, q, \boldsymbol{K}), \tag{7.29}$$

where λ is the current parameter set, and λ^* is the new parameter set. It has been proved that maximization of Q-function of the above form leads to increased likelihood [61] [103], i.e., $Q(\lambda, \lambda^*) > Q(\lambda, \lambda)$ implies that $LH_{\lambda^*}(\boldsymbol{O}) > LH_{\lambda}(\boldsymbol{O})$. Recall that the likelihood $LH_{\lambda}(\boldsymbol{O})$ is first partitioned in the state sequence space, and then further partitioned in the mixture sequence space. Similarly, by definition the Q-function can also be partitioned as

$$Q(\lambda, \lambda^*) = \sum_{j=1}^{N} Q_j(\lambda, \lambda^*) = \sum_{j=1}^{N} \sum_{k=1}^{K} Q_{jk}(\lambda, \lambda^*), \tag{7.30}$$

where

$$Q_j(\lambda, \lambda^*) = \sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} LH_{\lambda}(\boldsymbol{O}, q, \boldsymbol{K}) \cdot \log LH_{\lambda^*}(\boldsymbol{O}, q_t = j, \boldsymbol{K}), \tag{7.31}$$

and

$$Q_{jk}(\lambda, \lambda^*) = \sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} LH_{\lambda}(\boldsymbol{O}, q, \boldsymbol{K}) \cdot \log LH_{\lambda^*}(\boldsymbol{O}, q_t = j, k_t = k). \tag{7.32}$$

Based on (7.30)-(7.32), we successfully partition the Q-function, and construct the function (7.32). The inner summation of (7.32) has the identical form to the one used by Liporace in [102]. In [102], Liporace has already proved that the Q-function of that form has a unique global maximum as a function of λ^*, and this maximum is at a critical point. Assuming the mixture densities in (7.25) are non-Gaussian *and* log-concave or elliptically symmetric, and based on Liporace's theorems, we can conclude our Q-function that takes summations over N and K also has a unique maximum at a critical point. Thus, the maximization of the Q function leads to increased likelihood, and the likelihood function converges to a relative maximum.

Derivation of Re-estimation Formulas

Since we have proved the convergence of a relative maximum, we now can apply the standard Lagrange optimization technique to derive the re-estimation formulas for model parameter learning of the ICAMHMM framework.

To derive the re-estimation formulas, we can calculate the Q-function and split the results

into three terms as

$$
\begin{aligned}
Q(\lambda, \lambda^*) \;=\; & \sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} P(\boldsymbol{O}, q, \boldsymbol{K} \mid \lambda) \cdot \log \pi^*_{q_1} + \\
& \sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} P(\boldsymbol{O}, q, \boldsymbol{K} \mid \lambda) \cdot (\sum_{t=1}^{T-1} \log a^*_{q_t q_{t+1}}) + \\
& \sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} P(\boldsymbol{O}, q, \boldsymbol{K} \mid \lambda) \cdot \{\sum_{t=1}^{T} \log[p(\boldsymbol{o}_t \mid C^*_{q_t k_t}, \theta^*_{q_t k_t}] \cdot P(C^*_{q_t k_t})]\}.
\end{aligned}
\tag{7.33}
$$

Since the parameters π^*_i, a^*_{ij}, and $P(C^*_{jk})$ are independently separated in the sum, we can optimize each term individually. The first term in (7.33) becomes

$$
\sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} P(\boldsymbol{O}, q, \boldsymbol{K} \mid \lambda) \cdot \log \pi^*_{q_1} = \sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i \mid \lambda) \cdot \log \pi^*_i. \tag{7.34}
$$

To optimize the right hand side of (7.34), we can add the Lagrange multiplier ψ using the constraint that $\sum_i \pi^*_i = 1$, and setting the partial derivative to zero. We get

$$
\frac{\partial}{\partial \pi^*_i} [\sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i \mid \lambda) \cdot \log \pi^*_i + \psi(\sum_{i=1}^{N} \pi^*_i - 1)] = 0. \tag{7.35}
$$

Calculate the derivative and sum to get ψ first, and then solve for each π^*_i, we get

$$
\pi^*_i = \frac{P(\boldsymbol{O}, q_1 = i \mid \lambda)}{P(\boldsymbol{O} \mid \lambda)}. \tag{7.36}
$$

The second term in (7.33) becomes

$$
\sum_{q \in \ell} \sum_{\boldsymbol{K} \in \hbar} P(\boldsymbol{O}, q, \boldsymbol{K} \mid \lambda) \cdot (\sum_{t=1}^{T-1} \log a^*_{q_t q_{t+1}}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j \mid \lambda) \log a^*_{ij}. \tag{7.37}
$$

In a similar way, we use the Lagrange multiplier with the constraint $\sum_{j=1}^{N} a^*_{ij} = 1$, and get

$$
a^*_{ij} = \frac{\sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j \mid \lambda)}{\sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i \mid \lambda)}. \tag{7.38}
$$

The third term in (7.33) can be further split into two terms, so we get

$$\sum_{q\in\ell}\sum_{\boldsymbol{K}\in\hbar} P(\boldsymbol{O},q,\boldsymbol{K}\mid\lambda)\cdot\{\sum_{t=1}^{T}\log[p(\boldsymbol{o}_t\mid C^*_{q_tk_t},\theta^*_{q_tk_t}]\cdot P(C^*_{q_tk_t})]\}$$
$$=\sum_{j=1}^{N}\sum_{k=1}^{K}\sum_{t=1}^{T}P(\boldsymbol{O},q_t=j,k_t=k\mid\lambda)\cdot\log P(C^*_{jk})+$$
$$\sum_{j=1}^{N}\sum_{k=1}^{K}\sum_{t=1}^{T}P(\boldsymbol{O},q_t=j,k_t=k\mid\lambda)\cdot\log p(\boldsymbol{o}_t\mid C^*_{jk},\theta^*_{jk}). \tag{7.39}$$

The first term on the right hand side of (7.39) can be optimized using Lagrange multiplier to get the $P(C^*_{jk})$

$$P(C_{jk}{}^*)=\frac{\sum_{t=1}^{T}P(q_t=j,k_t=k\mid\boldsymbol{O},\lambda)}{\sum_{t=1}^{T}\sum_{k=1}^{K}P(q_t=j,k_t=k\mid\boldsymbol{O},\lambda)}. \tag{7.40}$$

The second term on the right hand side of (7.39) is the representation based on non-Gaussian densities. As formulated earlier, we associate each non-Gaussian mixture component with a standard ICA. Thus, the non-Gaussian component can be decomposed as a linear combination of statistically independent sources. The inference of non-Gaussian component densities in parametric form is described as follows.

In the proposed ICAMHMM framework, the spatial statistics of the observations are exploited by the ICA mixture model. The observation model is to represent the observation density functions as the weighted sum of non-Gaussian distributions. The number of the mixture components can be considered as the number of classes. For each class, the non-Gaussian density is represented as a function of statistically independent sources. Thus, without considering the state space, the data within the k-th ($1\leq k\leq K$) class are described by

$$\boldsymbol{o}_t=\boldsymbol{M}_k\cdot\boldsymbol{s}_k+\boldsymbol{\mu}_k, \tag{7.41}$$

where $\boldsymbol{M}_k$ is the basis matrix for the k-th component, $\boldsymbol{s}_k$ contains the statistically independent sources, and $\boldsymbol{\mu}_k$ is the bias vector for class k.

Taking the state space into account and applying ICA mixture method, the observation density at time t, given state j, is computed as

$$\log p(\boldsymbol{o}_t \mid C_{jk}, \theta_{jk}) = \log p(\boldsymbol{s}_{jk}) - \log(\det |\boldsymbol{M}_{jk}|), \tag{7.42}$$

where $\theta_{jk} = \{\boldsymbol{M}_{jk}, \boldsymbol{\mu}_{jk}\}$. Note that $\boldsymbol{M}_{jk}$ implicitly models the sources $\boldsymbol{s}_{jk}$ (see equation 7.41). The density of $\boldsymbol{s}_{jk}$ can be approximated by super-Gaussian or sub-Gaussian densities depending on the source model. Note that for each given state, we model the observations using a non-Gaussian mixture model. Thus, the number of parameters to be estimated is rather high. In order to make the implementation simple and to reduce the number of free parameters, we assume all the states share the same coefficients, i.e., the same observation model. Thus, in the following derivations, we drop the state index j in (7.42).

The basis matrix $\boldsymbol{M}_k$ for the k-th class can be learnt by using the standard ICA algorithm. Many ICA estimation algorithms have been developed, as described and reviewed in Chapter 2. We choose the infomax estimation algorithm because of its efficiency. Also, by choosing the different sigmoidal nonlinearities in infomax, it is suitable for learning both super-Gaussian and sub-Gaussian ICA sources [104] [53]. The adaptation of the basis matrix is

$$\triangle \boldsymbol{M}_k \propto p(C_k \mid \boldsymbol{o}_t, \Theta) \cdot \frac{\partial}{\partial \boldsymbol{M}_k} \log p(\boldsymbol{o}_t \mid C_k, \theta_k), \tag{7.43}$$

where $\Theta = \{\theta_1, \ldots, \theta_K\}$ are the parameters for each component density. $p(C_k \mid \boldsymbol{o}_t, \Theta)$ can be computed as

$$p(C_k \mid \boldsymbol{o}_t, \Theta) = \frac{p(\boldsymbol{o}_t \mid \theta_k, C_k) \cdot p(C_k)}{\sum_{k=1}^{K} p(\boldsymbol{o}_t \mid \theta_k, C_k) \cdot p(C_k)}. \tag{7.44}$$

The mean vector for each class is approximated by

$$\boldsymbol{\mu}_k = \frac{\sum_{t=1}^{T} \boldsymbol{o}_t \cdot p(C_k \mid \boldsymbol{o}_t, \Theta)}{\sum_{t=1}^{T} p(C_k \mid \boldsymbol{o}_t, \Theta)}. \tag{7.45}$$

A Summary of the Algorithm

The ICA mixture model itself in observation space only gives us the spatial statistics. The temporal dynamics are not considered in the ICA mixture model itself since generally ICA

algorithm ignores the order of the signals. However, by integrating ICA mixture model into HMM framework, the temporal information is modeled by the transition matrix and the state sequence. During implementation, we calculate the observation model first, i.e., to compute the density parameters using the ICA mixture model without considering the temporal information, and then use the results as the initializations to compute the parameters for HMM framework. In the observation modeling step, the following parameters are obtained by ICA mixture model: 1. the basis matrix coefficients $M = \{\boldsymbol{M}_k\}$, $1 \leq k \leq K$. 2. the mean vector coefficients $\mu = \{\boldsymbol{\mu}_k\}$, $1 \leq k \leq K$. 3. the observation densities $b(\boldsymbol{o}_t)$. To integrate them into ICAMHMM framework, these coefficients are duplicated for each state and used as initial values for each state. In the temporal modeling step, Equation (7.36), (7.38), and (7.40) are implemented and calculated using the intermediate variables iteratively. To summarize the algorithms derived above, the re-estimation formulas for ICAMHMM are listed as follows:

$$\pi_i^* = \gamma_1(i) \tag{7.46}$$

$$P(C_{jk}^*) = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_t(j,k)} \tag{7.47}$$

$$\boldsymbol{\mu}_{jk}^* = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{7.48}$$

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \tag{7.49}$$

Note that when calculating the intermediate variables like $\alpha_t(i)$, $\beta_t(i)$, $\xi_t(i,j)$, $\gamma_t(i,j)$, the observation densities $b_j(\boldsymbol{o}_t)$ is computed as

$$b_j(\boldsymbol{o}_t) = \sum_{k=1}^{K} P(C_{jk}) \cdot b_{jk}(\boldsymbol{o}_t), \tag{7.50}$$

where

$$b_{jk}(\boldsymbol{o}_t) = \exp(\log p(\boldsymbol{s}_{jk}) - \log(\det|\boldsymbol{M}_{jk}|)). \tag{7.51}$$

Equation (7.51) can be interpreted as the k-th component density, given the state j. Thus, the weighted summation over k gives the overall observation density at state j. During the practical implementation, the *log* is used to avoid the precision issues.

The procedures for ICAMHMM framework are summarized as follows:

1. Initialize the parameters, such as the number of mixtures K, the number of hidden states N, and the ICA source model $p(\boldsymbol{s})$.

2. Apply ICA mixture to model the observations, and calculate the parameters for mixture component densities.

3. Apply the derived re-estimation formulas (7.46)-(7.51) to compute all the parameters for ICAMHMM.

The detailed description of ICAMHMM learning is listed in Algorithm 1.

7.3.3 Likelihood Evaluation

The re-estimation formulas derived in the previous subsection can be used to compute all the parameters needed to represent a HMM model with non-Gaussian mixture observation densities. Each model (model parameters) represents one event. Given any observation sequence $\boldsymbol{O}$, the likelihood of producing the observation sequence is given by $P(\boldsymbol{O} \mid \lambda)$. The calculation of the likelihood is very similar to the classical *Forward-Backward Procedure*. The major difference is that the observation densities are computed from the sources and basis matrices. The likelihood $P(\boldsymbol{O} \mid \lambda)$ is inductively solved as follows:

1)Initialization:

$$\alpha_1(i) = \pi_i \cdot \sum_{k=1}^{K} P(C_{ik}) \cdot \exp(\log p(\boldsymbol{s}_{ik}) - \log(\det|\boldsymbol{M}_{ik}|)), \quad 1 \leq i \leq N. \tag{7.52}$$

2)Induction:

$$\alpha_{t+1}(i) = [\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij}] \sum_{k=1}^{K} P(C_{jk}) \cdot \exp(\log p(\boldsymbol{s}_{jk}) - \log(\det|\boldsymbol{M}_{jk}|)), \tag{7.53}$$

$$1 \leq t \leq T-1, \quad 1 \leq i \leq N. \tag{7.54}$$

3)Termination:

$$P(\boldsymbol{O} \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{7.55}$$

Algorithm 1 ICAMHMM Learning

Select the number of mixtures K and the number of hidden states N
Select ICA source model $p(\boldsymbol{s})$
Initialize ICA mixture model parameters $\boldsymbol{M}_k, \boldsymbol{\mu}_k, k = 1, \ldots, K$
Input observation sequence $\boldsymbol{o}_t, t = 1, \ldots, T$
Repeat
  Repeat
    For k = 1 to K do
      Calculate $\boldsymbol{s}_k$ using (7.41)
      Calculate $p(\boldsymbol{s}_k)$
      Calculate the observation densities $b(\boldsymbol{o}_t)$ using (7.50)
      Adapt the basis matrix $\boldsymbol{M}_k$ using (7.43)
      Adapt the bias vector $\boldsymbol{\mu}_k$ using (7.45)
    End
  Until all the observation samples have been used
  Calculate the class probability for each class using 7.44
Until the adaptation has converged
Assign each observation to one of the K classes
Initialize the initial state distributions using the class distributions
Re-calculate the bias vector $\boldsymbol{\mu}_k$ for each class
For each class, duplicate $\boldsymbol{\mu}_k$ and $\boldsymbol{M}_k$ for each state as initializations
Repeat
  Calculate observation densities and component densities using (7.50) and (7.51)
  Calculate $\alpha_t(i), \beta_t(i), \xi_(i,j), \gamma_t(i), \gamma_t(j,k), 1 \leq i, j \leq N, 1 \leq k \leq K$
  Calculate the likelihood of the observations sequence (the summation over $\alpha_t(i)$)
  Adapt the transition matrix
Until the adaptation has converged or the maximum number of iterations is reached

7.4 Event Detection Based on ICA Mixture Hidden Markov Model

The proposed ICAMHMM framework is applied to video data for content analysis. We develop an event detection and recognition system which includes the following modules: 1) Feature extraction. 2) Shot boundary detection. 3) Observation density estimation and model training. 4) Event detection. First, we choose the illumination invariant histograms to generate the raw features, and then ICA is applied to process these features and project the data into two dimensional ICA subspace. In the ICA subspace, shot boundaries are identified by using a clustering algorithm. Next, we assume that each event consists of one or multiple shots. Several training shots are selected to train the ICAMHMM model, and the rest of the video data are evaluated by the trained models. The model that gives the maximum likelihood is considered as the identified event.

Note that the iteration procedure proposed in section 7.3.3 only gives one likelihood for $\boldsymbol{O}$, given the model parameters. For D events, we need to calculate D likelihood given all the model parameters. The one that give the maximum is considered as the detected event (See equation 7.3 and equation 7.4).

7.4.1 Feature Extraction

In this chapter, we use the frame-based global features to analyze the content. However, the proposed framework can be easily extended to incorporate local features, such as object based features, to analyze the events which are associated to the specific video objects. Illumination change is an important factor that affects the performance of content based analysis of video data. To reduce the lighting effects, we choose the normalized chromaticity histograms [78] as our color features. Recall the 2D illumination-invariant normalized chromaticity described in Chapter 4, the (r, g) is defined as,

$$r = R/(R+G+B), \quad g = G/(R+G+B). \tag{7.56}$$

Histograms with 256 bins are generated as features in the normalized chromaticity color space for each video frame. The dimension of the feature vector for each video frame is 256. In our algorithms, the feature extraction proposed in Chapter 4 is applied to extract the two independent components (ICs) from high-dimensional feature vector. The ICA task is to find filter matrix using only the observations. Each video frame is processed as one observation that can be considered as a linear combination of hidden basis functions. The ICA model assumes that the observations are a linear combination of statistically independent sources. The illumination invariant histograms are used as the input signal. We denote $\boldsymbol{x}_t, t = 1, \ldots, T$ as the input signal, and $\boldsymbol{o}_t, t = 1, \ldots, T$ as the output signals of ICA learning model, where T is the number of video frames. The ICA learning model is defined as

$$\boldsymbol{o_t} = W \cdot \boldsymbol{x_t} = W \cdot M \cdot \boldsymbol{s}_t, \tag{7.57}$$

where W is the filter matrix, $\boldsymbol{s}_t$ is the statistically independent sources. $\boldsymbol{o}_t$ is considered as recovered independent sources. The rows of the output signals are independent components (ICs). Since the time course is only associated with the ICs, we select the most two significant ICs as the new features instead of the basis functions. Thus, the observation feature space is reduced from high-dimensional to 2-dimensional.

7.4.2 Shot Boundary Detection

Based on video frame distribution in the ICA subspace, a dynamic clustering algorithm [85] is applied to classify video frames into shots and detect the shot boundaries. Each video frame is represented by a point in ICA subspace, and Euclidean distance is used as dissimilarity measure between two points. A dynamic clustering algorithm based on adaptive thresholding is employed to detect shot boundaries [85]. Detailed description of the shot boundary detection can be found in Chapter 4.

7.4.3 Model Training

The output signals $\boldsymbol{o}_t, t = 1, \ldots, T$ from ICA learning in previous feature extraction step are used as observations in ICAMHMM framework. During ICAMHMM model training, the

spatial characteristics of the signals are captured by ICA mixture model and the temporal characteristics of the observations are explored by HMM's modeling to find the most probable state sequence. The parameters re-estimation formulas are derived in the earlier sections are used. The model training described in this section can be considered as a supervised learning technique since the training sequences are manually annotated. For each event, a different model is trained, and the model parameters are used to represent the event.

7.4.4 Event Detection

In the proposed event detection system, we assume each shot can be categorized into one of the candidate events. The detection of event is essentially a sequence classification since each event is represented as model parameters. For a new sequence, we evaluate the log-likelihood given each model. The event whose model gives the maximum log-likelihood will be declared as the detected event for the test sequence.

7.5 Experimental Results

In order to test the effectiveness of the proposed algorithm, we choose golf video data as a case study. For the golf video, the recurrent patterns are generally very recognizable especially when the players hit the long and straight shots. The first scene is relatively static when the payer prepares for his hit. After he swings and hits the ball, the next scene often contains high motion activities when the camera follows the ball. Finally, the scenes always focus on the golf court to track the ball or the player, and those generally contain low activities.

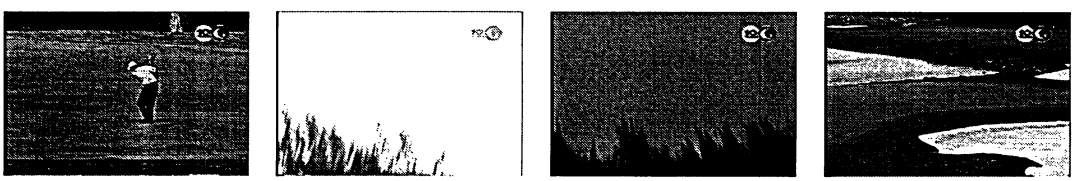


Figure 7.2: Video pattern for a tee shot with full swing.

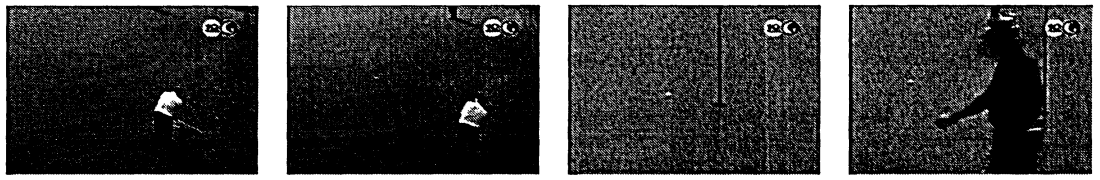


Figure 7.3: Video pattern for a fairway shot.

Figure 7.4: Irrelevant events such as talks and interviews.

In the experiment, one hour golf video is captured from TV. The video data is encoded in MPEG-1 format with a frame rate of 29.97 fps. The video contains different lighting conditions, multiple views in one window, and quick camera motions. Compared with the surveillance or traffic video with relatively static background, this golf video data is a challenge to analyze. Even though the video contains many dynamics, some recurrent patterns are still recognizable by human perception. For example, "full swing" scenes generally begin 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. Finally, the last scenes are usually some zoom-ins to locate the slowly moving ball. Other events include random camera moves, audiences, invited talks, and natural views. We define three events: "full-swing", "non-full-swing", and "irrelevant event". "Full-swing" event is defined as the golf swing that produces long and straight shots with full swings. "Non-full-swing" event is defined as the soft hit such as fairway shots and bunker shots, and generally the ball does not fly very high. The irrelevant event includes all other scenes such as the invited talk, the scene of the audience, etc. Three video sequences (Training sequence 1, 2, and 3) which contain different events are used as the training data. The goal is to detect which event an unlabeled video sequence might belong

to. The training results are shown in Table 7.1. The log-likelihood of the training sequences for each model/event are listed in the second column in Table 7.1. After the training, we use the rest of golf video data as the test sequences to verify the proposed event detection algorithms. All the test sequence are manually annotated and classified into one of the three events. The ground truth information is shown in Table 7.2. The event detection results are shown in Table 7.3. To evaluate the performance of the proposed event detection algorithm, we introduce a detection rate

$$D = \frac{N_{correct}}{N_{total}}, \tag{7.58}$$

where $N_{correct}$ is the number of correctly detected events, and N_{total} is the number of total events. The overall detection rate for the proposed framework is 70.79%.

The distributions of video frames in ICA subspace are plotted in Figure 7.5-7.10. It can be seen that different events have different patterns. For Figure 7.5 and 7.11, their patterns are very similar since they belong to the same event.

It is worth pointing out that the research of semantic event detection is currently still at an early stage. Thus, very few works have attempted to develop semantic event detection models. To compare the results with those of other works, we refer to [36] [101] for discussions. Xie [36] obtained an overall classification accuracy of 83.5% for "play" and "break" event detection in soccer domain, and Liu [101] achieved 84.7% with 5 hidden states and 128 symbols to classify TV programs into five categories using audio features. However, the features and the events defined in [36] are specifically for "play" and "break" events, and thus may not be generalized to other applications. The classification results obtained in [101] also depends on the characteristics of each category, and cannot be directly used for the generic semantic event detection and recognition.

7.6 Summary

In this chapter, a new statistical framework based on HMM and ICA mixture model is proposed to analyze the content of video. The observation densities of HMM are modeled by non-Gaussian mixtures and each non-Gaussian mixture density is learned by the standard

Table 7.1: Log-likelihood for the training sequences.

	Log-likelihood for the trained model
Training 1 sequence for event 1	2369.1
Training 2 sequence for event 2	-396.06
Training 3 sequence for event 3	7914.3

Table 7.2: Ground truth for the test sequences.

	The number of events
Event 1	54
Event 2	132
Event 3	16
Total events	202

Table 7.3: Event detection results.

	The number of events
Event 1	51
Event 2	132
Event 3	19
Total events	202

ICA. The new framework extends the classical continuous HMM and thus allows a larger range of densities to be modeled. The spatial statistics are explored by the ICA mixture model and the temporal characteristics are captured by HMM state transitions. We introduce a supervised learning for video event detection and recognition. The proposed framework is applied to golf video to detect three different semantic video events. Each event is described by one HMM model. During the evaluation period, the model that gives the maximum likelihood for the test sequence is considered as the detected event. The experimental results show that the presented method can effectively detect and recognize the recurrent patterns

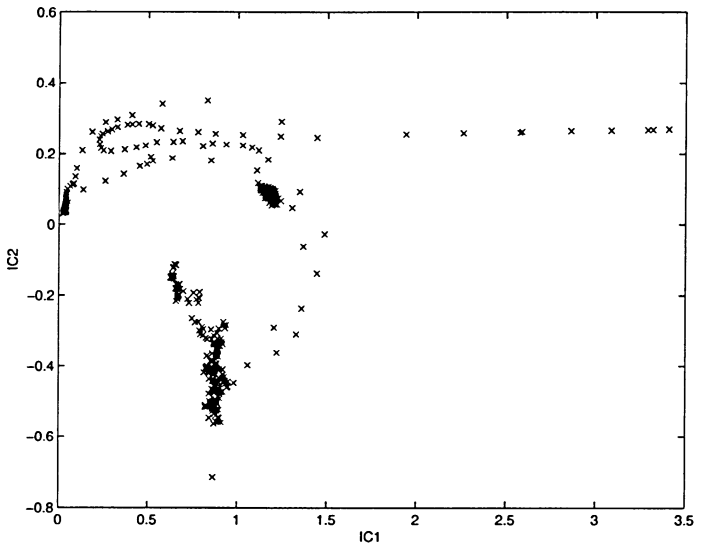


Figure 7.5: Training sequence 1 and its patterns for a full-swing shot.

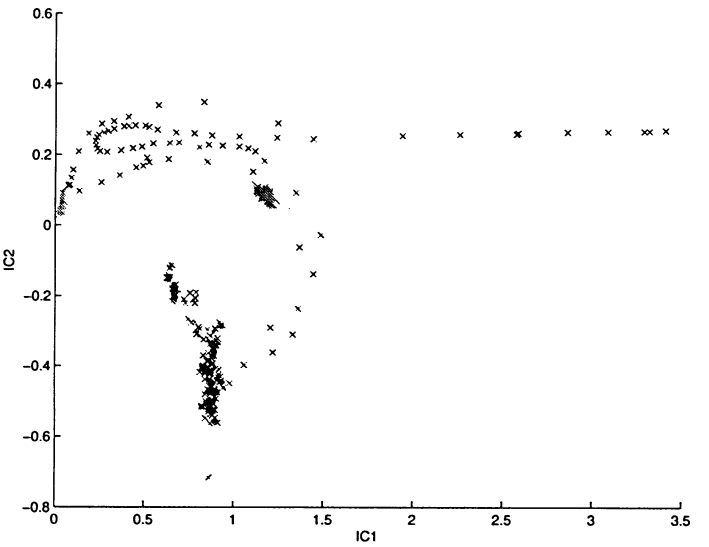


Figure 7.6: Four classes are learned for training sequence 1 (full-swing event).

in the video data.

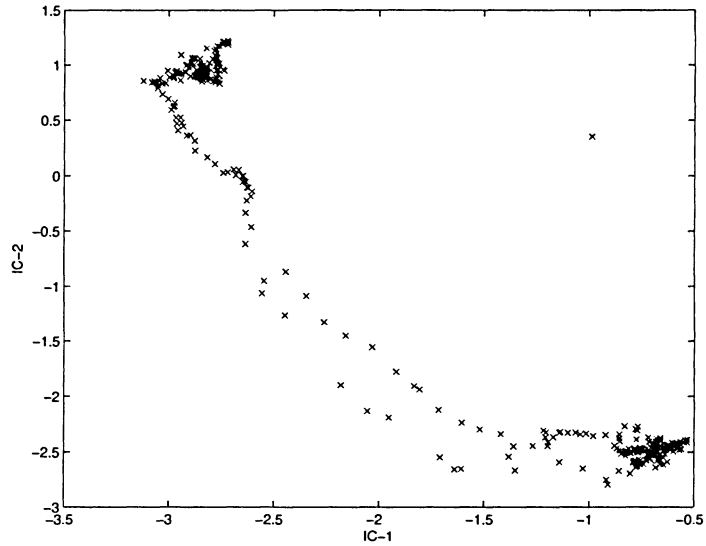


Figure 7.7: Training sequence 2 and its patterns for a non-full-swing shot.

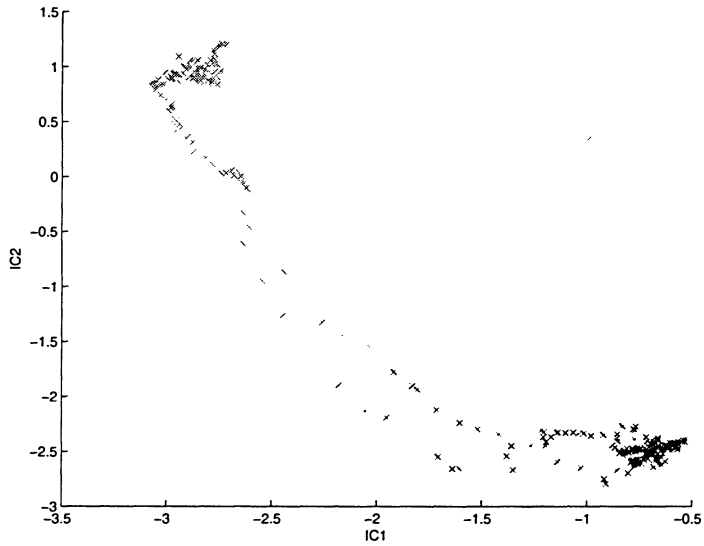


Figure 7.8: Two classes are learned for training sequence 2 (non-full-swing event).

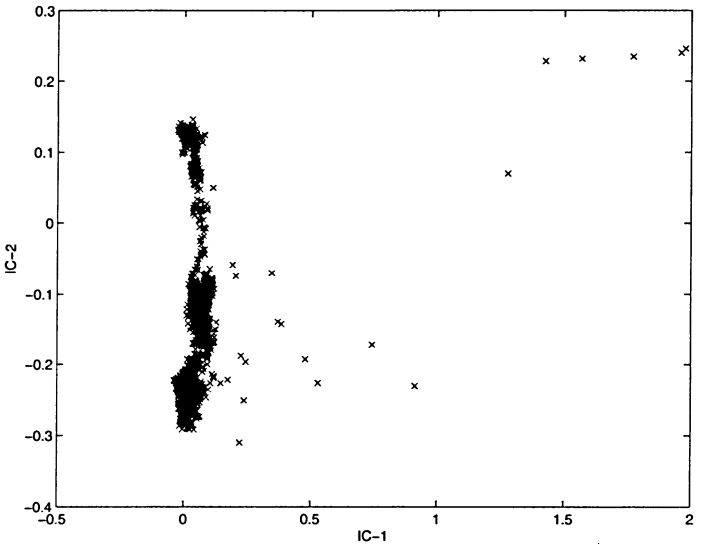


Figure 7.9: Training sequence 3 and its patterns for an irrelevant event.

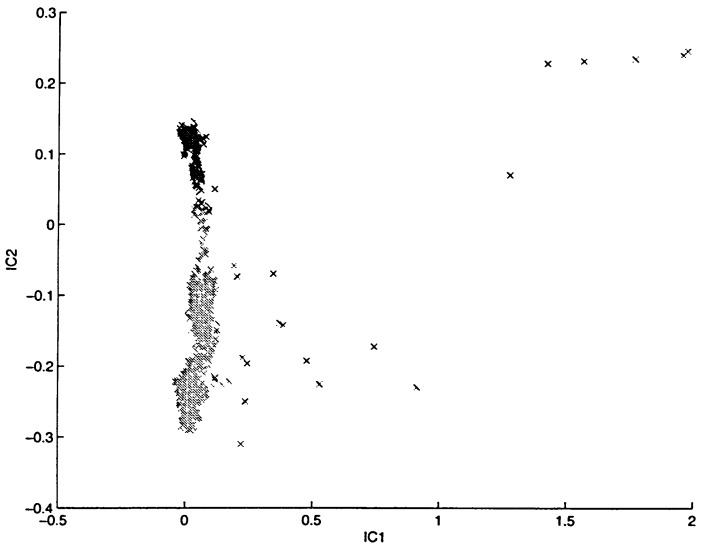


Figure 7.10: Two classes are learned for training sequence 3 (irrelevant event).

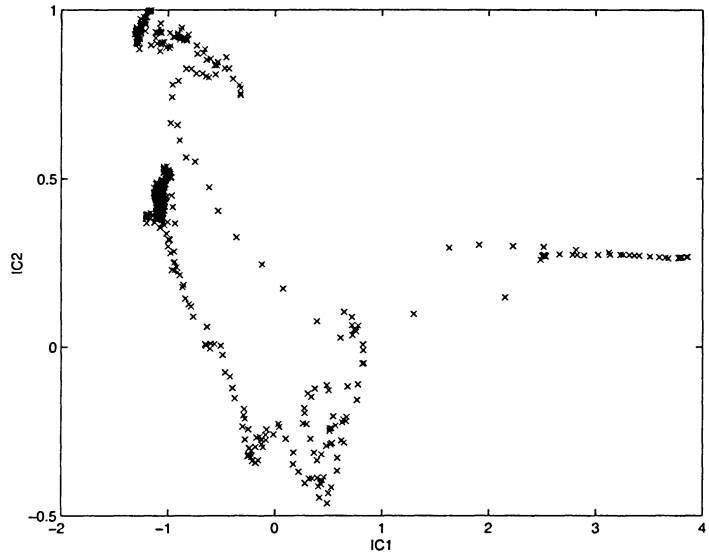


Figure 7.11: One example of the detected full-swing event.

Chapter 8

Conclusion and Future Work

Content analysis of video is still an active research area and many challenging problems still remain open. Our major concern is to find solutions and statistical modeling techniques for content analysis of video. In this thesis, we investigate some fundamental tasks for content analysis of video based on statistical modeling. New feature extraction techniques based on statistical analysis and a spatiotemporal framework combining HMM and ICA mixture model are developed. The proposed algorithms have been applied to several applications to analyze the video content.

Video parsing is an important step for video content analysis and many shot detection solutions have been developed by researchers. However, transition detection, especially the gradual transition detection still has room to improve since the discontinuities between the gradual transitions are still very difficult to model and detect. For this video parsing task, we apply the statistical analysis methods to video data and develop two solutions, namely, the online video parsing, and the off-line video parsing. The online video parsing uses the basic statistical measures and is fast to compute the features. In order to achieve better performance, we explicitly model the hard cuts and dissolves, and extend the classical variance-based methods to a mean-variance-skewness combined analysis. The advantages of the online video parsing are its speed and the simplicity of implementation. The method can be implemented as an online solution since essentially a decision of identifying a shot boundary can be made without knowing the whole video sequence. Experimental results

have shown its good performance measured in precision and recall. However, for the on-line solution, different methods are used to identify the abrupt transitions and the gradual transitions. Also, other gradual transitions such as wipes and fades are not considered even though they are not as common as dissolves. The off-line video parsing, on the other hand, processes and identifies both abrupt transitions and gradual transitions through one pass. The algorithm can be used to detect all types of transitions since it is based on dynamic clustering in ICA subspace. The disadvantage is that it requires all the video frames be available when extracting features. Thus, this method can be considered as an off-line solution. Another very common problem among the standard histogram-based methods is the effect of lighting changes. To avoid the problem, we transform the video frames into the illumination-invariant color spaces such that the features are insensitive to the different lighting conditions.

For video similarity models, we apply the proposed statistical analysis methods to extract shot-level features, and then develop the video dissimilarity measures using dynamic programming. Classical video dissimilarity models are mainly based on the distance-like functions in the feature space. The drawback is that the temporal information is not well utilized. The proposed dissimilarity models explore both spatial and temporal characteristics and fully make use of the temporal information.

Statistical analysis is also performed on video object segmentation and tracking. Classical solutions often use k-means for segmentation. We observe that the spatial constraints and the temporal information are not utilized during the k-means clustering. Therefore, we propose a new probabilistic fuzzy c-means framework to incorporate the Gibbs random fields as the spatial constraints. Motion vectors based on phase correlation are used to locate the active segmented regions. Block-based temporal tracking between two frames is directly performed on the membership matrix. Experimental results show that the proposed method can effectively extract the video objects.

Lastly, we develop a generic framework that combines HMM and ICA mixture model to explore the spatial and temporal characteristics of the signals. We extend the classical

continuous HMM modeling to allow the signals to be modeled as a mixture of non-Gaussian densities. An arbitrary non-Gaussian density generally is not tractable in parametric forms. To overcome this problem, we introduce the ICA mixture model as a density estimation tool. Each non-Gaussian mixture component is associated with a standard ICA, thus each non-Gaussian density can be parametrically represented as a basis matrix and the ICA sources. This new framework allows a broader range of distributions to be modeled, and is also application independent. We derive the re-estimation formulas to learn the model parameters, and then apply the method to the video data for semantic event detection and recognition. The statistical modeling methods and the feature extraction techniques proposed earlier for low-level content analysis are utilized together as the foundations for high level semantic analysis. Experimental results on golf video show that the new framework can effectively recognize the video events.

Even though the thesis covers major tasks for content analysis of video, there are still some possible research directions and potential applications which can be conducted in the future:

- For video indexing and summarization, the solutions we have proposed are mainly based on the frame-level global features. In Chapter 4, we propose a video object based segmentation and tracking algorithm. Object-based video indexing and summarization based on the localized object-level features could be the future work.

- For the proposed ICAMHMM framework, we only use golf video as a case study. We believe this framework can also be applied to other domains for content analysis. The object-level localized features can also be used for semantic analysis. For example, in surveillance video and traffic video, the background is generally static and the video objects are often the regions of interests. Thus, combing the proposed event detection framework and the localized features might create some interesting results.

- In the proposed ICAMHMM framework, the model topology structure is pre-defined. In the future, we could extend the framework to adaptively learn the model structure

from the data, though the complexity and the computation load might increase. It would be worth investigating whether the performance gain is significant if we relax the condition and let both the model structure and model parameters be learnt from the data at the same time.

- We have performed some preliminary implementations based on the proposed algorithms. However, because of the time limit, we do not have the chance to build a completely working video content analysis system. In the future, the development and implementation of a complete content based video management system would be very useful. Such a system could be used to verify the functionalities, validate the algorithms, and provide the first-hand user experience for video interaction and manipulation. The major modules may include automated vide parsing, indexing, summarization, video access and management, low-level video query and retrieval, and semantic video search and retrieval.

- For content analysis of video, user subjectivity and interaction modeling are also important. In the future, some research areas such as user access behavior modeling, query and interaction modeling, semi-automated or automated relevance feedback could be very significant and promising research directions.

- The algorithms proposed in this thesis could be used in some potential applications, such as video codec, content protection, consumer electronics, network devices, and network software. For example, a scene change (shot boundary) in the video content often results in an increase of the throughput in the streaming network devices, thus, a strategy based on the content analysis of video could be integrated into the network devices to manage the internal buffers. For digital TV, in the future, it might be possible to employ the content analysis techniques to increase the resolution of certain areas in the picture, or use the video object-based motion-adaptive solutions to reduce the cross-luma and cross-chroma artifacts.

List of Abbreviations

Abbreviation	Details
CBVR	Content Based Video Retrieval
DCT	Discrete Cosine Transform
EMD	Earth Mover's Distance
FDR	Fisher's Discriminant Ratio
HMM	Hidden Markov Model
ICA	Independent Component Analysis
ICAMHMM	ICA Mixture Hidden Markov Model
KL-divergence	Kullback-Leibler divergence
MPEG	Moving Picture Experts Group
MRF	Markov Random Field
PCA	Principal Component Analysis
SMIL	Synchronized Multimedia Integration Language
XML	Extensible Markup Language

Bibliography

[1] T. Sikora, "Mpeg digital video-coding standards," *IEEE Signal Processing Magazine*, vol. 14, no. 5, pp. 82–100, Sept. 1997.

[2] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," *Visual Database Systems*, vol. 2, pp. 113–127, 1992.

[3] H.J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, June 1993.

[4] H.J. Zhang, C.Y. Low, S.W. Smoliar, and J.H. Wu, "Video parsing, retrieval and browsing: An integrated and content-based solution," in *ACM Multimedia'95*, San Francisco, CA, Nov. 1995, pp. 15–23.

[5] S.-F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, "VideoQ: An automated content-based video search system using visual cues," in *ACM Multimedia'97*, Seattle, WA, Nov. 1997, pp. 313–324.

[6] H.J. Zhang, S.W. Smoliar, J.H. Wu, C.Y. Low, and A. Kankanhalli, "A video database system for digital libraries," in *Selected Papers from the Digital Libraries Workshop on Digital Libraries*, London, UK, 1995, pp. 253–264, Springer-Verlag.

[7] A. Del Bimbo, *Visual Information Retrieval*, Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[8] S. Santini and R. Jain, "The graphical specification of similarity queries," *Journal of Visual Languages and Computing*, vol. 7, no. 4, pp. 403–421, 1996.

[9] W. Zhou and C.-C. Jay Kuo, *Intelligent Systems for Video Analysis and Access over the Internet*, IMSC Press Multimedia Series. Prentice Hall, Upper Saddle River, NJ, 2003.

[10] M. Flickner, H. Sawhney, W. Niblack, and *et al.*, "Query by image and video content: the QBIC system," *IEEE Computer*, vol. 3, no. 9, pp. 23–32, 1995.

[11] A. Pentland, R.W. Picard, and S. Sclaroff, "PhotoBook: Content-based manipulation of image databases," *International Journal of Computer Vision (Historical Archive)*, vol. 18, no. 3, pp. 233–254, June 1996.

[12] J. Bach, C. Fuller, A. Gupta, and *et al.*, "The Virage image search engine: An open framework for image management," *Proc. SPIE Storage and Retrieval for Image and Video Databases*, vol. 2670, pp. 76–87, 1996.

[13] J. R. Smith and S. Chang, "VisualSEEK: a fully automated content-based image query system," in *ACM Multimedia'96*, Boston, MA, Nov. 1996, pp. 87–98.

[14] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. Huang, "Supporting similarity queries in MARS," in *ACM Multimedia'97*, Seattle, WA, Nov. 1997, pp. 403–413.

[15] B. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, pp. 533–544, Dec. 1995.

[16] L. Guan, S.-Y. Kung, and J. Larsen, *Multimedia Image and Video Processing*, Image Processing Series. CRC Press LLC, Boca Raton, FL, 2001.

[17] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *ACM Multimedia'95*, San Francisco, CA, Nov. 1995, pp. 189–200.

[18] A.M. Alattar, "Detecting and compressing dissolve regions in video sequences with a dvi multimedia image compression algorithm," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 1993, vol. 1, pp. 13–16.

[19] B.T. Truong, C. Dorai, and S. Venkatesh, "New enhancements to cut, fade, and dissolve detection processes in video segmentation," in *ACM Multimedia'00*, Los Angeles, CA, Nov. 2000, pp. 219–227.

[20] M.M. Yeung, B. Yeo, and B. Liu, "Time-constrainted clustering for segmentation of video into story units," in *Proc. Int. Conf. Pattern Recognition*, Vienna, Austria, Aug. 1996, pp. 375–380.

[21] M.M. Yeung and B. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771–785, Oct. 1997.

[22] Y. Rui, T.S. Huang, and S.F. Chang, "Image retrieval: Current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, no. 4, pp. 39–62, Apr. 1999.

[23] A.M. Ferman and A.M. Tekalp, "Efficient filtering and clustering methods for temporal video segmentation and visual summarization," *J. Vis. Commun. and Image Rep.*, vol. 9, no. 4, pp. 336–351, Dec. 1998.

[24] E. Sahouria and A. Zakhor, "Content analysis of video using principal components," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1290–1298, Dec. 1999.

[25] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 2001.

[26] J. Goldberger, S. Gordon, and H. Greenspan, "An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures," in *ICCV'03*, 2003, pp. 487–493.

[27] Y. Rubner, "Texture metrics," in *Ph.D. Thesis*, Stanford University, May 1999.

[28] Y. Rubner, C. Tomasi, and L.J. Guibas, "The earth mover's distance as a metric for image retrieval," in *STAN-CS-TN-98-86*, Stanford University, 1998.

[29] S.-S. Cheung and A. Zakhor, "Efficient video similarity measurement with video signature," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 59–74, Jan. 2003.

[30] S.-S. Cheung and A. Zakhor, "Fast similarity search and clustering of video sequences on the world-wide-web," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 524–537, June 2005.

[31] A. Kushki, P. Androutsos, K.N. Plataniotis, and A.N. Venetsanopoulos, "Retrieval of images from artistic repositories using a decision fusion framework," *IEEE Trans. on Image Processing*, vol. 13, no. 3, pp. 277–292, Mar. 2004.

[32] J.Y.A. Wang and E.H. Adelson, "Representing moving images with layers," *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 625–638, Sept. 1994.

[33] F.G. Meyer and P. Bouthemy, "Region-based tracking using affine motion models in long image sequences," *CVGIP: Image Understanding*, vol. 60, no. 2, pp. 119–140, 1994.

[34] Y. Wu and T.S. Huang, "A co-inference approach to robust visual tracking," in *ICCV'01*, 2001, vol. 2, pp. 26–33.

[35] D. Serby, E.K. Meier, and L. Van Gool, "Probabilistic object tracking using multiple features," in *ICPR'04*, 2004, vol. 2, pp. 184–187.

[36] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and Hidden Markov Models," *Pattern Recognition Letters*, vol. 25, no. 7, pp. 767–775, May 2004.

[37] L. Xie, S. Chang, A. Divakaran, and H. Sun, "Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models," in *IEEE Intl. Conf. Multimedia and Expo (ICME)*, July 2003.

[38] M.R. Naphade and T.S. Huang, "Extracting semantics from audiovisual content: the final frontier in multimedia retrieval," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 793–810, July 2002.

[39] 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 *Proc. of 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings (ICASSP '03)*, Hong Kong, China, Apr. 2003, pp. V – 632–5.

[40] X. Li and F.M. Porikli, "A hidden Markov model framework for traffic event detection using video features," in *Proc. of 2004 IEEE International Conference on Image Processing, 2004. Proceedings (ICIP '04)*, Singapore, Oct. 2004, vol. 5, pp. 2901–2904.

[41] M.H. DeGroot, *Probability and Statistics (Second Edition)*, Addison-Wesley Series in Statistics. Addison-Wesley Publishing Company, Menlo Park, California, 1989.

[42] P. Comon, "Independent component analysis, a new concept?," *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.

[43] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[44] R. Vigario, J. Sarela, V. Jousmiki, M. Hamalainen, and E. Oja, "Independent component approach to the analysis of EEG and MEG recordings," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 5, pp. 589–593, May 2000.

[45] J.V. Stone and J. Porill, "Regularisation using spatiotemporal independence and predictability," *Computational Neuroscience Report*, vol. 201, 1999.

[46] J. Hurri, A. Hyvärinen, J. Karhunen, and E. Oja, "Image feature extraction using independent component analysis," in *Proc. NORSIG'96, Espoo, Finland.*, 1996.

[47] A.J. Bell and T.J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[48] S.Z. Li, X. Lv, and H.J. Zhang, "View-subspace analysis of multi-view face patterns," in *RATFG-RTS '01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, Washington, DC, USA, 2001, pp. 125–132, IEEE Computer Society.

[49] A. Hyvärinen, "New approximations of differential entropy for independent component analysis and projection pursuit," in *Advances in Neural Information Processing Systems.* 1998, vol. 10, pp. 273–279, The MIT Press.

[50] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[51] A.J. Bell and T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[52] J.F. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Signal Processing Letters*, vol. 4, no. 4, pp. 112–114, April 1997.

[53] T.-W. Lee, M. Girolami, and T. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources," *Neural Computation*, vol. 11, no. 2, pp. 609–633, 1999.

[54] J. Cardoso, "Blind signal separation: statistical principles," in *Proceedings of the IEEE, Special issue on blind identification and estimation*, R.-W. Liu and L. Tong, Eds., 1998, vol. 9, pp. 2009–2025.

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

[56] M.Y. Hwang and X. Huang, "Shared-distribution hidden Markov models for speech recognition," *IEEE Trans. on Image Processing*, vol. 1, no. 4, pp. 414–420, Oct. 1993.

[57] L.R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[58] M.Y. Chen, A. Kundu, and S, "Variable duration hidden Markov model and morphological segmentation for handwritten word recognition," *IEEE Trans. on Image Processing*, vol. 4, no. 12, pp. 1675–1688, May 1995.

[59] J.-L. Chen, A. Kundu, and S.N. Srihari, "Unsupervised texture segmentation using multichannel decomposition and hidden Markov models," *IEEE Trans. on Image Processing*, vol. 4, no. 5, pp. 603–619, May 1995.

[60] C. Anton-Haro, J.A.R. Fonollosa, and J.R. Fonollosa, "Blind channel estimation and data detection using hidden Markov models," *IEEE Trans. on Signal Processing*, vol. 45, no. 1, pp. 241—247, Jan. 1997.

[61] L.E. Baum and G.R. Sell, "Growth functions for transformations on manifolds," *Pac. J. Math.*, vol. 27, no. 2, pp. 211–227, 1968.

[62] J.K. Baker, "The dragon system – an overview," *IEEE Transactions on Acoust. Speech Signal Processing*, vol. 23, no. 1, pp. 24–29, 1975.

[63] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[64] M.J. Russell and R.K. Moore, "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, USA, Mar. 1985, pp. 29–45.

[65] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden Markov model structure for information extraction," in *AAAI 99 Workshop on Machine Learning for Information Extraction*, 1999.

[66] Z. Ghahramani, "Learning dynamic Bayesian networks," *Lecture Notes in Computer Science*, vol. 1387, pp. 168–197, 1998.

[67] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *MULTIMEDIA '95: Proceedings of the third ACM international conference on Multimedia*, New York, NY, USA, 1995, pp. 189–200, ACM Press.

[68] B. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, pp. 533–544, Dec. 1995.

[69] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 90–104, Feb. 2002.

[70] R. Lienhart, "Reliable transition detection in videos: A survey and practitioners guide," *International Journal of Image and Graphics (IJIG)*, vol. 1, no. 3, pp. 469–486, Aug. 2001.

[71] B.L. Tseng, C.Y. Lin, and J.R. Smith, "Using MPEG-7 and MPEG-21 for personalizing video," *IEEE Multimedia*, vol. 11, no. 1, pp. 42–52, Mar. 2004.

[72] J. Boreczky and L. Rowe, "Comparison of video shot boundary detection techniques," in *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*, 1996, pp. 170–179.

[73] A. Poirson and B. Wandell, "Pattern-color separable pathways predict sensitivity to simple colored patterns," *Vision Research*, vol. 36, no. 4, pp. 515–526, 1996.

[74] R. Lienhart, "Reliable transition detection in videos: A survey and practitioner's guide," *International Journal of Image and Graphics*, vol. 1, no. 3, pp. 469–486, 2001.

[75] G. Geisler and G. Marchionini, "The open video project: research-oriented digital video repository," in *DL'00: Proceedings of the fifth ACM conference on Digital Libraries*, New York, NY, USA, 2000, pp. 258–259, ACM Press.

[76] J. Ayers *et al.*, *Synchronized Multimedia Integration Language (SMIL) 2.0*, World Wide Web Consortium Recommendation, [Online]: http://www.w3.org/TR/smil20/, Aug. 2003.

[77] J. Zhou and X.-P. Zhang, "A web-enabled video indexing system," in *ACM MIR'04*, New York, USA, Oct. 2004, pp. 307–314.

[78] M.S. Drew, J. Wei, and Z.-N. Li, "Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalize images," in *ICCV'98*, 1998, pp. 533–540.

[79] R. Lancini, F. Mapelli, and A. Mucedero, "Automatic identification of compressed video," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, May 2004, pp. 445–448.

[80] P. Muneesawang and L. Guan, "Automatic relevance feedback for video retrieval," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Apr. 2003, pp. 1–4.

[81] T. Lin, C.W. Ngo, H.J. Zhang, and Q.Y. Shi, "Integrating color and spatial features for content-based video retrieval," in *Proc. Int. Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 592–595.

[82] L. Zhao, W. Qi, S.Z. Li, S.Q. Yang, and H.J. Zhang, "Content-based retrieval of video shot using the improved nearest feature line method," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, May 2001, pp. 1625–1628.

[83] H. Cheng, "Temporal registration of video sequences," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Hong Kong, China, Apr. 2003, pp. 489–492.

[84] Y. Tan, S. Kulkarni, and P. Ramadge, "A framework for measuring video similarity and its application to video query by example," in *Proc. Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999, pp. 106–110.

[85] J. Zhou and X.-P. Zhang, "Video shot boundary detection using independent component analysis," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Philadelphia, PA, Mar. 2005, pp. 541–544.

[86] A. Marzal and E. Vidal, "Computation of normalized edit distance and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 926–932, Sept. 1993.

[87] I.E.G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*, John Wiley & Sons, New York, 2003.

[88] G.D. Borshukov, G. Bozdagi, Y. Altunbasak, and A.M. Tekalp, "Motion segmentation by multistage affine classifications," *IEEE Trans. on Image Processing*, vol. 6, no. 11, pp. 1591–1594, Nov. 1997.

[89] B. Han, D. Comanicui, Y. Zhu, and L. Davis, "Incremental density approximation and kernel-based bayesian filtering for object tracking," in *CVPR04*, 2004, pp. 638–644.

[90] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatio-temporal segmentation based on region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 1591–1594, Sept. 1998.

[91] C. Kim and J.N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 122–129, Feb. 2002.

[92] Y. Deng, C. Kenney, M.S. Moore, and B.S. Manjunath, "Peer group filtering and perceptual color image quantization," in *Proc. of IEEE Intl. Symposium on Circuits and Systems*, 1999, pp. 21–24.

[93] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[94] T.D. Pham, "Image segmentation using probabilistic fuzzy c-means clustering," in *Proc. Int. Conf. Image Processing*, 2001, pp. 722–725.

[95] Z. Chi, H. Yan, and T. Pham, *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*, World Scientific, Singapore, 1996.

[96] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[97] A.M. Tekalp, *Digital video processing*, Prentice-Hall, Upper Saddle River, NJ, USA, 1995.

[98] R. Castagno, "Video segmentation based on multiple features for interactive and automatic multimedia applications," in *Ph.D. Thesis*, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1998.

[99] N. Dimitrova and F. Golshani, "Motion recovery for video content classification," *ACM Transactions on Information Systems*, vol. 13, no. 4, pp. 408–439, Oct. 1995.

[100] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhang, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 602–615, Sept. 1998.

[101] Z. Liu, J. Huang, and Y. Wang, "Classification of TV programs based on audio information using Hidden Markov Model," in *Proc. of 1998 IEEE Second Workshop on Multimedia Signal Processing (MMSP'98)*, Redonda Beach, CA, Dec. 1998, pp. 27–31.

[102] L.A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Inform. Theory*, vol. 28, no. 5, pp. 729–734, Sept. 1982.

[103] B.-H. Juang, S. Levinson S., and M. Sondhi, "Maximum likelihood estimation for multivariate Mixture observations of Markov Chains," *IEEE Trans. Inform. Theory*, vol. 32, no. 2, pp. 307–309, Mar. 1986.

[104] T.-W. Lee and M.S. Lewicki, "Unsupervised image classification, segmentation, and enhancement using ICA mixture models," *IEEE Trans. on Image Processing*, vol. 11, no. 3, pp. 270–279, Mar. 2002.

VITA

NAME:	Jian Zhou
PLACE OF BIRTH:	Heilongjiang, China
YEAR OF BIRTH:	1974
POST-SECONDARY EDUCATION AND DEGREES:	Nankai University Tianjin, China 1993-1997, B.Eng.
HONORS AND AWARDS:	Nankai Scholarship 1995-1996
RELATED WORK EXPERIENCE:	Video DSP Engineer Pixelworks Inc. 2005-, Toronto, Canada
	Software Engineer & Team Lead Vialta Inc. 2000-2002, Toronto, Canada
	Software Engineer Jitong Communications Co., Ltd. 1999-2000, Shenyang, China
	Software Engineer Shenyang Institute of Computing Technology Chinese of Academy Sciences 1997-1999, Shenyang, China

PUBLICATIONS

1. Jian Zhou and X.-P. Zhang, "A Web-enabled Video Indexing System," in *Proceedings of ACM MIR'04*, New York, USA, Oct. 2004.

2. Jian Zhou and X.-P. Zhang, "Video shot boundary detection using independent component analysis," in *Proceedings of IEEE ICASSP'05*, Philadelphia, PA, USA, Mar. 2005.

3. Jian Zhou and X.-P. Zhang, "Video Object Segmentation and Tracking using Probabilistic Fuzzy C-means," in *Proceedings of IEEE MLSP'05*, Mystic, CT, USA, Sept. 2005.

4. Jian Zhou and X.-P. Zhang, "Automatic identification of digital video based on shot-level sequence matching," in *Proceedings of ACM MM'05*, Singapore, Nov. 2005.

5. Jian Zhou and X.-P. Zhang, "A Hidden Markov Model framework for Content Analysis of Video," in preparation.