AUTOMATIC EXTRACTION OF HIGHLIGHTS FROM A BASEBALL VIDEO USING HMM AND MPEG-7 DESCRIPTORS

Abdullah Naseer Ahmed Saudagar

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

May 2011

APPROVED:

Kamesh Namuduri, Major Professor
Parthasarathy Guturu, Committee Member
Murali Varanasi, Committee Member and Chair of the Department of Electrical Engineering
Costas Tsatsoulis, Dean of College of Engineering
James D. Meernik, Acting Dean of the Toulouse Graduate School

Saudagar, Abdullah Naseer Ahmed. Automatic Extraction of Highlights from a Baseball Video Using HMM and MPEG-7 Descriptors. Master of Science (Electrical Engineering), May 2011, 38 pp., 2 tables, 5 illustrations, bibliography, 35 titles.

In today's fast paced world, as the number of stations of television programming offered is increasing rapidly, time accessible to watch them remains same or decreasing. Sports videos are typically lengthy and they appeal to a massive crowd. Though sports video is lengthy, most of the viewer's desire to watch specific segments of the video which are fascinating, like a home-run in a baseball or goal in soccer i.e., users prefer to watch highlights to save time. When associated to the entire span of the video, these segments form only a minor share. Hence these videos need to be summarized for effective presentation and data management.

This thesis explores the ability to extract highlights automatically using MPEG-7 features and hidden Markov model (HMM), so that viewing time can be reduced. Video is first segmented into scene shots, in which the detection of the shot is the fundamental task. After the video is segmented into shots, extraction of key frames allows a suitable representation of the whole shot. Feature extraction is crucial processing step in the classification, video indexing and retrieval system. Frame features such as color, motion, texture, edges are extracted from the key frames. A baseball highlight contains certain types of scene shots and these shots follow a particular transition pattern. The shots are classified as close-up, out-field, base and audience. I first try to identify the type of the shot using low level features extracted from the key frames of each shot. For the identification of the highlight I use the hidden Markov model using the transition pattern of the shots in time domain. Experimental results suggest that with reasonable accuracy highlights can be extracted from the video.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my major advisor Dr. Kamesh Namuduri for his guidance, constant mentoring and support. I would like to thank my advisory committee members Dr.Murali Varanasi, Dr. Parathasarathy Guturu for their motivation to finish my thesis successfully. I owe my gratitude to the faculty and staff of the electrical engineering department for their moral support. Lastly, I am very grateful to my family and friends, especially my parents for their encouragement and support.

TABLE OF CONTENTS

Page

LIST OF TABLES AND ILLUSTRATIONS

Page

Tables

Figures

CHAPTER 1

INTRODUCTION AND PREVIEW

1.1 Introduction

Vast amount of digital video is generated in our daily lives. Effective classification and retrieval of the desired information from huge collections of digital video is one of the most crucial and challenging problems. A lot of successful paradigms have emerged for video parsing, indexing, summarization, classification and retrieval. Although fruitful results have been achieved, more challenging problems need to be addressed and overcome in this field of research.

This increased generation and distribution rate of audiovisual content created a new problem: management of content. Unlike the tools for creation and distribution, tools to manage multimedia content are not matured enough. There are no feasible ways to automatically analyze, classify and browse the content.

Another issue that is currently in active research is the streaming of video over various networks. In the past few years, the number of streaming video sources has raised significantly. These cover almost any domain, ranging from personal web cameras, sports event transmission, news programs, surveillance and movie on demand streaming applications.

Sports video distribution over various networks should contribute to quick adoption and widespread usage of multimedia services worldwide, because sports video appeals to large audience. Processing of sports video, for example detection of important events and creation of summaries, makes it possible to deliver sports video over narrow band networks, such as Internet and wireless, since the valuable semantics generally occupy only a small portion of the whole

content. The value of sports video, however, drops significantly after a relatively short period of time.

Manual annotation of sports video by skilled librarians can be very time consuming. Hence it is more desirable to have automatic systems for multimedia content annotation and summarization. Certain features need to be extracted from the multimedia data which can be used for the data management, access, search and retrieval.

Automatic video extraction emerges as a solution to the problem of managing video content. Automatic video summarization can be defined as identifying relevant parts of a video and presenting it in an easily browsable form, with minimal user intervention.

1.2 Motivation

Watching sports is fun and especially when it comes to baseball it has lot of action. It is really interesting to watch a live match, but when it comes to browsing the action that had already taken place in a video it is really time consuming and painstaking effort. The value of the sports video falls and the volume of the video increases significantly with time. The amount of effort spent to extract all the important events in the video is relatively huge. Another important issue in recent times is the streaming of the live video on different bandwidth networks, because live video is bandwidth intensive. So, the solution is to send all the important events in the game making sure that the viewer's get all the action in the game. To catch all that action of important events in the video an automatic extraction model is required. This is the motivation for investigating the methods that can automate extraction of highlights from baseball video.

1.3 Thesis Overview and Contribution

This thesis discusses the automatic extraction of highlights from a baseball game video. In order to extract the highlights automatically, the system has to identify different views and align that pattern of views to flag that as a highlight. Here I first divided the given video in to different shots that does not have meaning on their own but gain meaning when joined together. From all the shots, I extracted the key frames that are designed to represent all or part of a shot. Key frames are individual images in the shots. Thus the amount of computation can be reduced if feature extraction is performed on the key frames instead of the entire shot. I extracted the different low level features from key frames such as color, motion, texture and edges that are defined by the MPEG-7 descriptors and using these features I classified the shots into different views like close-up, base, in-field, out-field views. Highlights in the game generally exhibit a sequence of views; I used this behavior and develop a hidden Markov model (HMM). Having done with the preprocessing of feature extraction and view classification, I applied the knowledge of HMM to extract the highlight sequence.

1.4 Thesis Outline

Before describing the work of the thesis, chapter 2 presents the literature survey and provides the necessary background.

Chapter 3 gives the detailed explanation of how video is structured in section 3.2, various types of shot transitions in section 3.3. Sections 3.5 and 3.6 explain the extraction of key frames and feature extraction from the key frames. Then the next section 3.7 explains the classification of shots using the available features.

In chapter 4 HMM for highlight detection gives the mathematical model behind the

highlight extraction. In sections 4.1, 4.2 and 4.3 of this chapter, I first gave a brief overview of the mathematical model, the problems faced in the model and the solutions to the problems. The last three sections show how this mathematical model helps in the extraction of the highlights.

Chapter 5 summarizes the experimental results obtained for shot detection, view classification and the overall performance of the system. Flow chart gives the overview of the system.

Chapter 6 concludes the presentation, in which it summarizes the major results and insights in the study. Suggestions for future extensions to the system including future study of the HMM and modifications for improving the system performance and accuracy are provided in this chapter.

1.5 Thesis Overview Diagram

The following figure illustrates all the stages of the highlight extraction process.

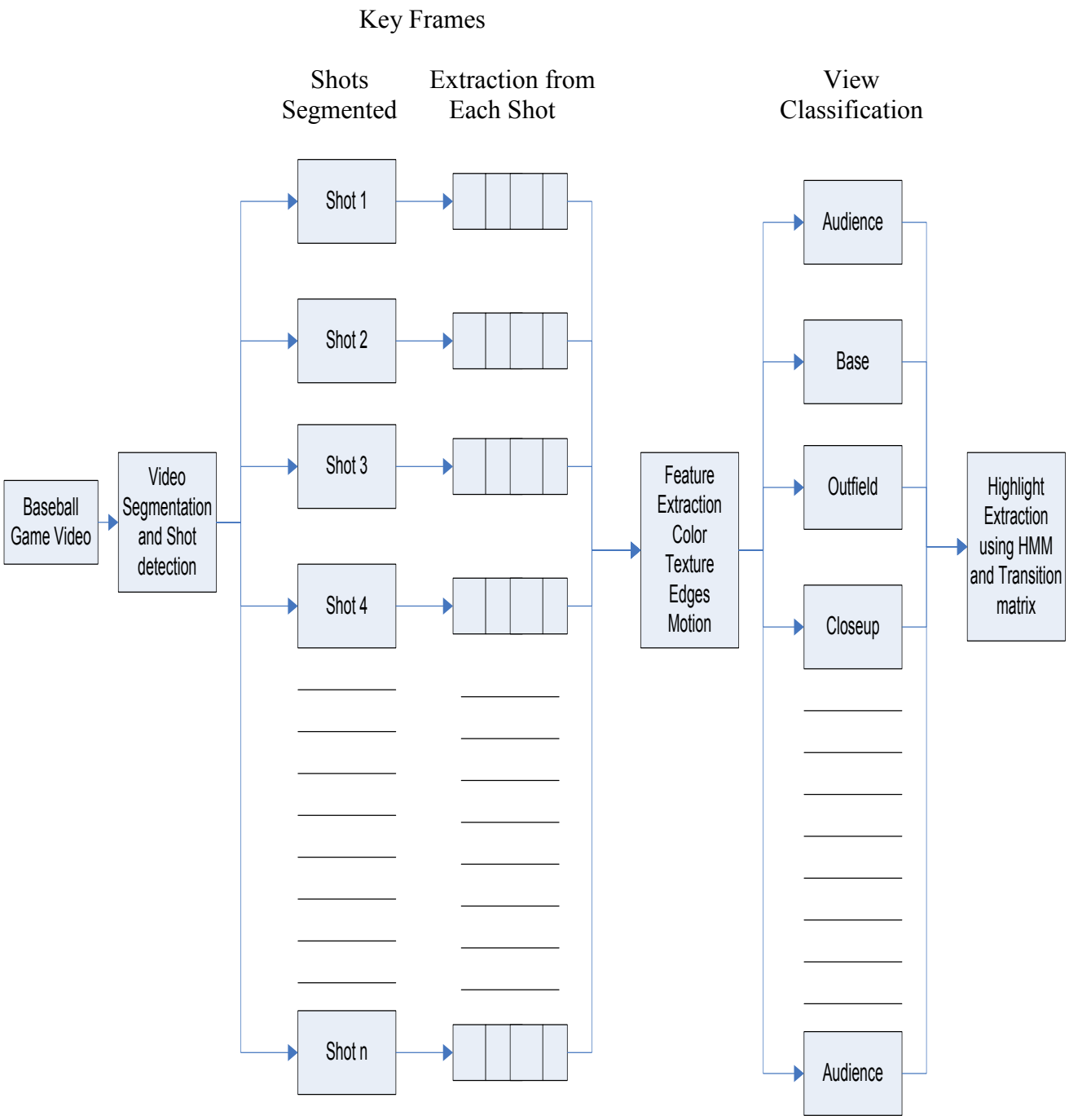

Figure 1.1 Overview Diagram

CHAPTER 2

LITERATURE SURVEY

2.1 Video Segmentation and Shot Detection

Video segmentation is the important step towards the content-based video classification and retrieval. Perfect automatic segmentation of video is hard to achieve with the current state-of-the-art. Dividing the video into different shots is the basic step in video segmentation. The identification of the shot is done at the shot boundaries, making the boundaries as a transition for shots. So identifying the boundaries will make segmentation of the video easier into shots. A shot transition can take the form of an abrupt change. Joining two shots together, one after another, produces an abrupt change known as a hard cut. A gradual effect is created by inserting edited versions of the two adjoining frames, resulting in a fade, dissolve, morph, or wipe.

Hard cuts are easily the most common form of shot boundary probably due to the fact that minimal editing is required in order to achieve such a transition [1], [2], [3]. Since the change between frames on a hard cut is drastic, it is easier to detect when compared to gradual changes. Due to the presence of this dissimilarity characteristic on hard cut boundaries, the measure of change between two consecutive frames can give an indication as to whether a shot boundary has been found or not.

A fade is a shot transition where either the last frame in a shot is dimmed to black by gradually lowering brightness levels, or the brightness of a black frame is gradually increased to reveal the first frame of a shot [1], [2], [3]. A dissolve is when a fade occurs in both directions, where one shot fades out while at the same time another shot fades in. As these transitions are gradual from one shot to the next, certain features may only exhibit slight changes, and this can cause problems when differentiating between inter-frame changes and inter-boundary changes.

While specific features may be used to detect fades and dissolve correctly, hard cuts can be found by more features and therefore are easier to detect.

2.2 Extraction of Key Frames and Features from a Shot

Key frames are still images which best represent the content of the video sequence in an abstract manner, and may be either extracted or reconstructed from original video data. Key frames are frequently used to supplement the text of a video log, but there has been little progress in identifying them automatically. The challenge is that the extraction of the key frames needs to be automated and content based so that they maintain the important content of the video while removing all redundancy. Key frames provide abridged representation of the original video sequence, serving a multitude of applications depending on the needs of the user. Key frames can provide a low bandwidth representation of the video sequence can serve as pointers to the desired portion of the video content or can be used in video indexing application. A native approach to key frame extraction is to choose the first frame of the shot as a key frame [4]. However this method fails in case of high intensity shots. The key frame should have little overlap in the content so as to provide maximum information [5]. A more robust method to key frame selection based on color histogram was proposed by Zhang [6]. However all these methods consider the first frames as their key frames. To provide more information, I adopted a method in which I took all frames with significant changes as the key frames. Thus, I have minimum of one or more key frames in a shot.

On a symbolic level, a digital image can be represented by image features which contain the information relevant for subsequent image interpretations. Features involved in an image are classified as spectral features (special color or tone, gradient, spectral parameters etc.,),

geometric features (edges, shape, size etc.,), textural features (pattern, spatial frequency, homogeneity etc.,). Feature extraction is a crucial preprocessing step for video indexing, classification and retrieval system. Most work on video classification and retrieval can be viewed as the extension of traditional image retrieval techniques. They select key frames in the video shots and extract image features based on selected key frames. However, these approaches neglect the important spatial-temporal information of video. In our model for extraction of all these features, I have considered the MPEG-7 format descriptors.

2.3 MPEG – 7 Descriptors

Producing multimedia content nowadays is much easier than before with the digital tools such as digital cameras, personal computers etc which make everyone a potential content producer who are capable of creating, modifying, distributing the data easily. However what would seem like a dream can easily turn into an ugly nightmare if no means are available to manage the explosion in available content. Content analog and digital alike, has value only if it can be discovered and used. Content that cannot be easily found is like content doesn't exist, and potential revenues exist directly on the users finding the content. MPEG-7 also called "multimedia content standardization interface," standardizes the description of multimedia content supporting a wide range of applications. Standardization activities do not focus so much on processing tools but concentrate on the selection of features that have to be described.

The MPEG-7 project has the objective of specifying a standard way of describing various types of multimedia information: elementary pieces, complete works and repositories, irrespective of their representation format and storage medium. The objective is to facilitate the quick and efficient identification of interesting and relevant information and efficient

management of that information. But MPEG-7 is quite a different standard than its predecessors. MPEG-1, MPEG-2, MPEG-4 all represent the content itself – "the bits," while MPEG-7 represents the information about the content i.e., "the bits about the bits." It means the former standards reproduce the content and the later describes the content. The descriptor is a representation of a feature, where the feature is distinct characteristic of the data that signifies something. In this thesis, I have taken only the descriptor extraction methods defined in MPEG-7 standard. The descriptors that I have considered in our work are color, motion and texture. The extraction of these features is described in detail in section 3.6.

2.4 Hidden Markov Model

During the last twenty years, HMMs have been extensively applied in several areas including speech recognition [7], [8], [9], [10], language modeling [11], handwriting recognition [12], [13], [14], [15], facial expressions, human action learning [16], fault detection in dynamic systems [17].

Before describing the Hidden Markov Model, it is necessary to describe its foundation, the Markov process. In any pattern, there is usually sufficient structure to influence the probability of the next event. A HMM [18] is a doubly stochastic process, with an underlying stochastic process that is not observable (hence the word hidden), but can be observed through another stochastic process that produces the sequence of observations. The hidden process consists of a set of states connected to each other by transitions with probabilities, while the observed process consists of a set of outputs or observations, each of which may be emitted by each state according to some output probability density function (pdf). Depending on the nature of this pdf, several kinds of HMMs can be distinguished. If the observations are naturally

discrete or quantized using quantization or vector quantization, and drawn from an alphabet or a codebook, the HMM is called discrete. If these observations are continuous, I am dealing with a continuous HMM, with a continuous pdf usually approximated by a mixture of normal distributions. In some applications, it is more convenient to produce observations by transitions rather than by states. Furthermore, it is sometimes useful to allow transitions with no output in order to model, for instance, the absence of an event in a given stochastic process. If I add the possibility of using more than one feature set to describe the observations, I must modify the classic formal definition of HMMs.

CHAPTER 3

SHOT DETECTION AND CLASSIFICATION

3.1 Introduction

The first step in content based video retrieval is the temporal segmentation of the video. Video segmentation approach is application independent and detects temporally continuous segments that have less content change between frames. These frames together can be called as a shot. I detected the shots at the shot boundaries where an abrupt or gradual change can be seen. I classified these shot boundary transitions depending upon the type of change.

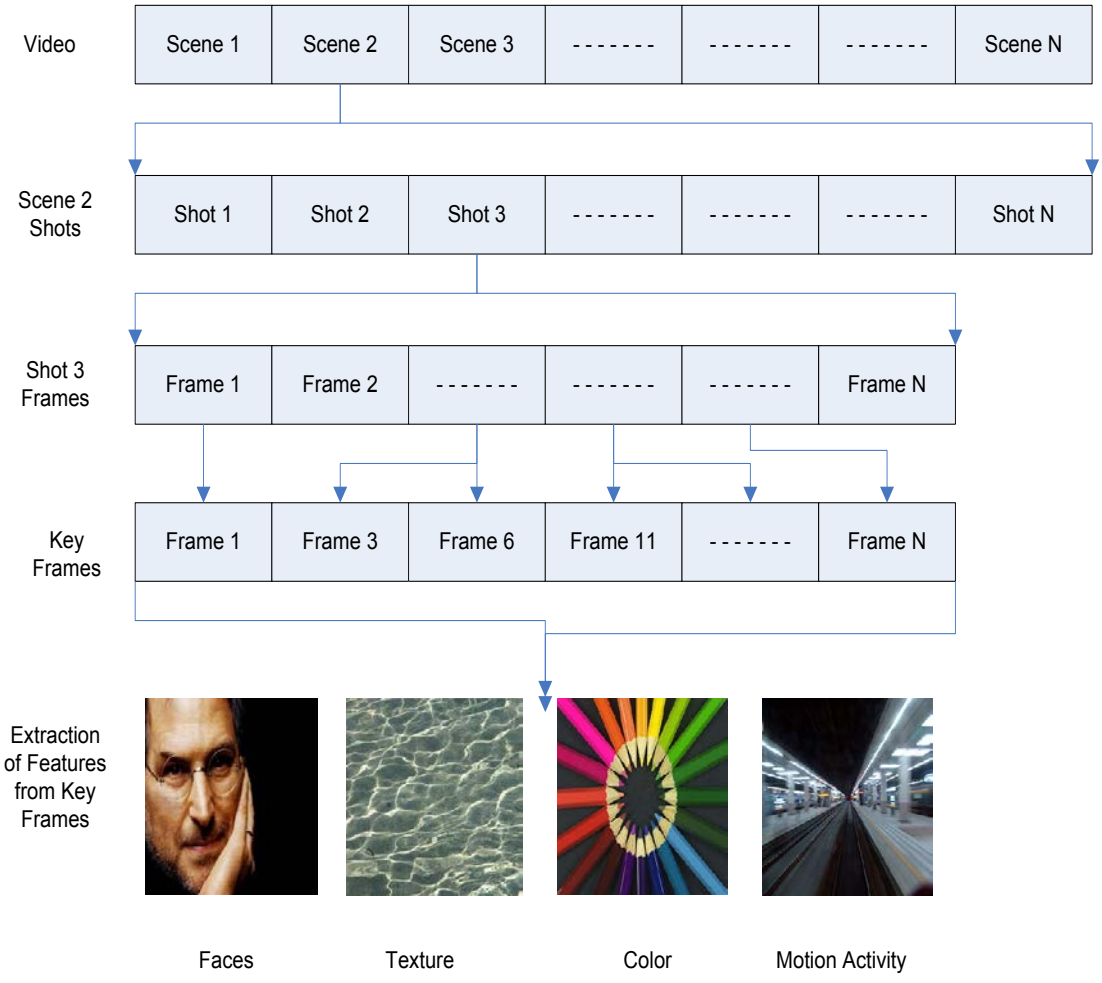

Figure 3.1 Structure of Video

3.2 Structure of Video

Video structure parsing is an initial step to organize the content of video. Video data is typically organized in a typical hierarchical structure as structure as shown in Fig. 3.1. In this step, some elementary units such as scenes, shots, frames, key frames and objects are generated. A successful structure parsing is important for video indexing, classification and retrieval. Lot of research has been done in video structure parsing, especially in shot detection, motion analysis and video segmentation.

3.3 Types of Shot Transition

A shot may be defined as a sequence of frames captured by "a single camera in a single continuous action in time and space" [19]. For example, a video sequence showing two people having a conversation may be composed of several close-up shots of their faces which are interleaved and make up a scene. Shots define the low-level, syntactical building blocks of a video sequence. A large number of different types of boundaries can exist between shots [20].

A cut is an abrupt transition between two shots that occurs between two adjacent frames. A fade is a gradual change in brightness, either starting or ending with a black frame. A dissolve is similar to a fade except that it occurs between two shots. The images of the first shot get dimmer and those of the second shot get brighter until the second replaces the first. Other types of shot transitions include wipes and computer generated effects such as morphing.

3.4 Shot Detection

3.4.1 Features Used in Detection

The process of shot detection requires comparisons to be made between particular

features of the video. These features can be different perspectives of the video content, such as color, objects, and motion, or video stream attributes such as compression information.

Here I gave the different features with which different shot transitions can be detected. Pixel based comparison which has been well discussed by Carrato S. Koprinka [3]. However, this pair-wise comparison of two frames can be too sensitive during camera motion, since a large number of pixels may be classified as changing regardless of the fact that the motion may only move them one or two pixels away, block based comparison is studied well by Zhang in his work [2]. The complexity of the process makes it slow to compute, and there is also a possibility that a boundary may be missed if two different blocks have the same mean and variance. Histogram representation is the method I used in our shot detection since it has shown good results in detecting shots [21], [22], [23], [24]. Histogram similarity measure is studied in [19], [25]. Edge change ratio involves calculating the number of new edge pixels entering the shot and the number of old pixels leaving the shot [26], [1]. Motion analysis is another method which requires the computation of motion vectors. This process of motion compensation can be computed using the block-matching algorithm [2], [27].

3.4.2 Previous Work

Much research has been done in automatic content analysis and segmentation of video. Researchers have focused mainly on different methods of shot boundary detection, with most techniques analyzing consecutive frames to decide if they belong to the same shot. Zhang et.al [2] use a pixel-based difference method, which, although slow, produced good results once the threshold was manually tailored to the video sequence. Another more common method is to use histograms to compare consecutive video frames. Nagasaka and Tanaka [28] compared several

statistical techniques using grey level and color histograms. Zhang et. al [2] used a running histograms method to detect gradual as well as abrupt shot boundaries. Cabedo and Bhattacharjee [19] used the cosine measure for detecting histogram changes in successive frames and found it more accurate than other similar methods. Gong et. al [29] used a combination of global and local histograms to represent the spatial locations of color regions. Borezcky and Rowe [20] compared several different methods of shot boundary detection using a variety of video content types. They concluded that histogram-based methods were more successful than others.

3.4.3 Proposed Approach for Shot Boundary Detection

Lot of research has been done in detecting shot transitions from video. In this thesis I used color histogram based segmentation. Generally in histogram segmentation systems two forms of histogram are used in shot boundary detection, the global histogram which represents the frame as a whole, and the local histogram which represents one block within a frame. In either case, the histogram would have colors (or intensities) grouped into bins along the x-axis, and the frequencies of these values present up the y-axis. Each bin holds a range of similar colors (or intensities), this range being dependent on the number of bins used. When using a global histogram, pixel values over the whole frame are categorized into their respective bins. The bin to bin difference can then be used as a measure of change between two such histograms, and these differences are then summed to give a total difference over the two consecutive frames. If this exceeds a given threshold then a boundary is declared.

$$D_{\mathrm{L}}(\mathrm{i}, \mathrm{i}+1) = \max_{1 \le j \le N} \left| H_i(\mathrm{j}) - H_{i+1}(\mathrm{j}) \right| \qquad (3.4.1)$$

where H_i is the histogram for the i^{th} frame and j takes the bin number given N bins and D_L gives the dissimilarity in the bins. This method [5] considers only the current and the previous frames. But in our approach I have considered previous, current and the next frames which will give more accuracy in identifying the shot boundaries. This can be achieved by considering the dissimilarity between the D_{pc} and D_{cn} exceeding a given threshold, where D_{pc} and D_{cn} represent the dissimilarity in bins between previous-current frames and current-next frames. This small change has proven to be efficient for detection.

$$D_{pcn} = \begin{cases} shot\ boundary & if\ |D_{cp} - D_{cn}| > threshold \\ not\ a\ shot\ boundary & otherwise \end{cases} \quad (3.4.2)$$

3.5 Extraction of Key Frames from Shots

3.5.1 Definition

Although video skimming conveys pictorial, motion and audio information, I focused on the creation of a visual summary using still images extracted from the video. Still images which are key frames can summarize the video content in a rapid and very compact way. The user can grasp the overall video content more quickly that by watching a set of video sequence.

3.5.2 Automatic Key Frame Selection

Different methods can be used to select representative frames. A naïve method discussed in [4] considers the initial frame in the shot as the key frame. In our approach I computed the differences between the consecutive frames in a shot in terms of color histogram. The reason for using color is that it enables a reliable measure of change from frame to frame and decreases the amount of computational complexity.

Instead of considering the first frame as the key frame, I compared the first frame with consecutive frames and if there is a frame whose dissimilarity is more than a threshold then it is considered as the next key frame. I continued this process on the whole shot, generating one or more key frames.

3.6 Features Extracted from Key Frames based on MPEG 7 Descriptors

MPEG -7 offers a comprehensive set of audio visual description tools to create descriptions which will form the basis for applications enabling effective and efficient access to multimedia content like searching, filtering and browsing.

Because the descriptive features must be meaningful in the context of the application, they will be different for different applications. This implies that the same material can be described using different types of features, based on the application. A lower abstraction level would be a description of shape, size, texture, color, movement, and position. The highest level could give semantic information. The level of abstraction is related to the way the features can be extracted: many low level features can be extracted in fully automated ways, whereas high level features need more human interaction.

3.6.1 Dominant Color

Color is an important visual attribute for both the human and the computer processing. The dominant color descriptor (DCD) in MPEG -7 provides a compact description of the representative color in an image or image region. Unlike the traditional histogram based descriptors, the representative colors are computed from each image space, thus allowing the color representation to be accurate and compact. The extraction procedure for the dominant color

uses the generalized Lloyd algorithm to cluster [30] the pixel color values. The clustering has to be performed in a perceptually uniform color space such as CIE LUV. The distortion D_i in the i_{th} cluster is given as

$$D_i = \sum_n h(n) \; ||x(n) - C_i||^2 \qquad (3.6.1)$$

where n is the total number of clusters and $x(n) \in C_i$, C_i is the centroid of the cluster, x(n) is the color vector at pixel n and h(n) is the perceptual weight for the pixel n. The perceptual weights are calculated from the local pixel statistics to account for the fact that human visual perception is more sensitive to change in smooth regions than in texture regions [30]. I am mainly interested in the extraction of the dominant color from the images and the number of these clusters for particular color e.g. green for grass. With this count of clusters on each key frame I will evaluate the observation probabilities that are used in the HMM model.

3.6.2 Texture Extraction

Image texture has emerged as an important visual primitive for searching and browsing through large collections of similar looking patterns. Pictures of water, grass or a pattern on fabric are examples of image texture. Many natural and artificial objects can be distinguished by their texture. Texture is a region property, as is evidenced by these examples. While it is easy to visualize what one means by texture there is no universally accepted formal definition of texture. One can think of a texture as consisting of some basic primitives, whose spatial distribution in the image creates the appearance of a texture.

3.6.3 Motion Activity

I generally use motion activity to indicate the level of motion within the shot. I usually

see high level of motion in sports and certain action shots such as the goal scenes. Thus, I classified the motion into low (like in an Anchor- person shot where only the head region has some movements), medium (such as those shots with people walking), high, or no motion (for still frame shots).

Motion activity may describe several attributes that contribute towards the efficient use of these motion descriptors in a number of applications. For example, the MPEG -7 frameworks allows for motion descriptors relative to intensity, spatial distribution, temporal distribution and directional distribution of activity. I considered the computation and implementation of motion intensity in spatial distribution of activity. In previous works, these features have been used successfully for shot boundary detection and key frame extraction for the purpose of video indexing, retrieval and summarization.

Motion activity is typically measured using the magnitude of motion vectors. For a game video frame, let x(i,j) and y(i,j) denote motion vectors in the x and y directions respectively, where (i,j) indicates the block indices. The spatial activity matrix z(i,j) defined in [31] as

$$z(i,j) = \begin{cases} R_{xy}(i,j) & if\ R_{xy}(i,j) \geq avg(R_{xy}(i,j)) \\ 0 & otherwise \end{cases} \quad (3.6.2)$$

where

$$R_{xy}(i,j) = \sqrt{x(i,j)^2 + y(i,j)^2} \quad (3.6.3)$$

and the average

$$avg(R_{xy}(i,j)) = \frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N} R_{xy}(i,j) \quad (3.6.4)$$

Here MxN denotes the size of each frame. This approach ignores low activity blocks and maintains high activity blocks unaltered to form the spatial activity matrix.

Intensity of activity is expressed by an integer in the range (1-5), where higher values of intensity correspond to higher motion activity [32]. The intensity of motion for each frame is determined as follows

$$I_n = \frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N} z(i,j) \tag{3.6.5}$$

where n is the frame index. The intensity activity is normalized and quantized based on its variance across all frames.

3.6.4 Edge Detection Using Canny's Method

Canny proposed the optimal edge detection algorithm [33]. An optimal edge detector is characterized by good detection (the algorithm should mark as many real edges in the image as possible), good localization (edges marked should be as close as possible to the edge in the real image), and minimal response (a given edge in the image should only be marked once, and where possible, image noise should not create false edges). To satisfy these requirements, Canny used the calculus of variations – a technique which finds the function which best satisfies a given functional. The optimal detector was described by the sum of four exponential terms, which can be closely approximated by the first derivative of Gaussian.

3.7 Shot Classification

After the video is segmented, there are several ways in which the contents of each shot can be modeled. I can model the contents of the shot (a) using a representative key frame; (b) as feature trajectories; (c) using a combination of both. I adopted the hybrid approach as a compromise to achieve both efficiency and effectiveness. Visual content features color, texture and edges will be extracted from the key frames while motion feature will be extracted from the

temporal contents of the shots. This is reasonable as I expect the visual contents of shots to be relatively similar so that a key frame is a reasonable representation of a shot.

3.7.1 Types of Views

The next step is to determine an appropriate and complete set of categories to cover all types of shots. The categories must be meaningful so that the category tag assigned to each shot is reflective of its content and facilitates the subsequent stage of segmenting and classifying a highlight. I studied the set of categories employed in related works, and the structure of game videos to arrive at the following set of shot categories:

1. Close up view
2. Outfield view
3. Base view
4. Audience view

3.7.2 Shot Classification

After the preprocessing, each shot is classified based on its feature. Note that the classification rules are based on combinations of relatively simple feature extraction processes, which make the method applicable to large video data.

I noticed that scene shots of same type often have similar distributions of color; texture and camera motion etc., while scene shots of different types usually differ in those distributions. For example, the color distribution and texture distribution of outfield view is very different from an audience view because audience view has more texture similarity and outfield views have more of the green color. I expect that with proper statistical methods it is feasible to extract the

common properties among the same type of scene shots and use those statistics to discriminate among different types of scene shots.

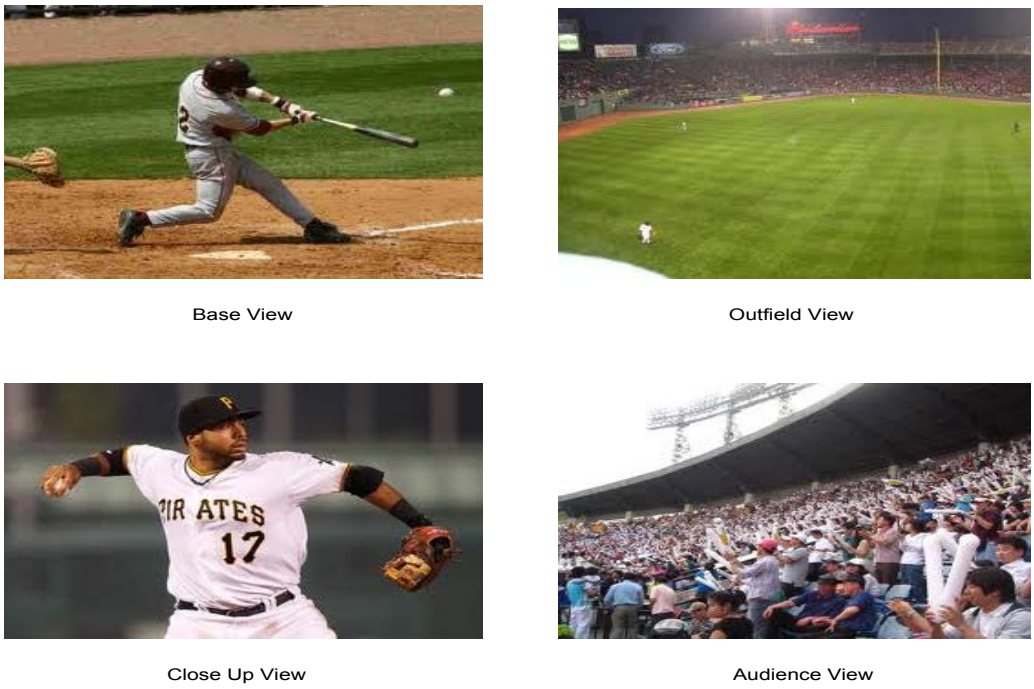

Figure 3.2 Different Views

I classified the view type of the shot based on the feature distribution. The following equations give the probability of a shot being in a particular view

$$P(V_i|O_k,\ k = 1,\ \ldots..,\ m) = \frac{P(O_k,k = 1,\ldots.,m|V_i)P(V_i)}{P(O_k,k = 1,\ldots.,m)} \quad (3.7.1)$$

where $P(V_i|O_k)$ is the probability of a shot being a particular view V_i, given the feature distribution O_k. $P(O_k|V_i)$ is the probability of features in a given view which is the posterior probability. $P(V_i)$ is the prior probability of the views which can be calculated from the training data.

CHAPTER 4

HIDDEN MARKOV MODEL FOR HIGHLGHT DETECTION

Before describing the hidden Markov model (HMM) [18], it is necessary to describe its foundation, the Markov process. In any pattern there is usually sufficient structure to influence the probability of the next event. For example, in the English language, the probability of detecting the letter M depends very much on whether the letter L was last detected, since M almost always follows L. A stochastic process is called jth–order Markov process if the conditional probability density of the current event, given all past and present events, depends only on the j most recent events. An HMM is a doubly stochastic process. Each state is a possible observation of the Markov process, and a transition probability from state A to state B is P(st+1 = B|st = A), the probability of going to state B at time t+1 given that the state at time is t is A. The second stochastic layer of the HMM is the set of output probabilities for each state. For example, the output probabilities of state A specifies the likelihood of seeing certain observations, given the HMM [18] is actually in state A. This second layer of probabilities creates a veil so that, given a sequence of observations, the actual sequence of states in ambiguous; it is "hidden" from the observer. Algorithms exist for both training and testing the HMMs. The goal of HMM training is to lift the veil so that, with good probability, the actual sequence of states S can be determined from the sequence of observations X. However, enough training data must be provided so that a good internal statistical model can be built. Proven to converge, the Baurn-Welch re-estimation procedure can be find locally optimal HMM parameters for a given set of training data. Reasonable initial estimates can help the procedure find the globally optimal solution. For testing and recognition, the Viterbi algorithm determines

the state sequence $\vec{s}$ with the highest probability, given a particular observation sequence $\vec{x}$(i.e., it maximizes $P(\vec{s}|\vec{x}, \lambda)$, where λ represents the HMM model)

4.1 Equations and the Parameters Used in HMM

- N: Number of states in the model.
- M: Number of possible observation symbols per state.
- V: Denote the possible symbols $\{v_1, v_2, \ldots, v_M\}$.
- $S = \{s_0, s_1, \ldots, s_{N-1}\}$, set of all possible states of the model.
- $Q = \{q_t\}$, $t = 0, 1, \ldots, T-1$, $\{q_t\}$: state of the process at time t.
- $A = \{a_{ij}\}$, the state transition probability distribution.

  where $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$, $1 \leq i, j \leq N$.
- $B = \{b_j(k)\}$, the observation symbol probability in state j.

  where $b_j(k) = P(v_k \text{ at } t | q_t = s_j)$, $1 \leq j \leq N$, $1 \leq k \leq M$.
- $\pi = \{\pi_i\}$, initial state distribution.

  where $\pi_i = P(q_t = s_i)$, $1 \leq i \leq N$.

For all optimal values given to M, N, A, B and π the HMM generates a maximum probable observation sequence $O = o_1, o_2, \ldots, o_T$. With all the above parameters I can make a compact notation of the HMM model as $\lambda = (A, B, \pi)$. With this given model there exist three problems that need to be solved for HMM model to be used in real world applications.

4.2 Three Basic Problems in HMM

4.2.1 Problem 1

Given an observation sequence $O = o_1, o_2, \ldots, o_T$ and the model λ, how to compute the

probability of observation sequence P(O|λ). This is an evaluation problem solved using forward-backward algorithm.

4.2.2 Problem 2

Given an observation sequence O = o_1, o_2, …., o_T and the model λ, how to find the optimal state sequence that produces the observation sequence O. This is decoding problem, solved using Viterbi algorithm.

4.2.3 Problem 3

Given an observation sequence O = o_1, o_2, …., o_T and the model λ, how to re-estimate the model parameters so as to increase the likelihood of generating this set of sequences. This is a training problem solved by using Baurn-Welch method.

4.3 Solutions to the Problems

4.3.1 Forward-Backward Procedure

To compute P(O|λ), I used the well-known Forward-Backward procedure. I took into account the assumption that symbols are emitted along the transitions. Hence I defined the forward probability as $\alpha_n(i) = P(o_1, o_2, \ldots\ldots, o_n, q_n = i|\lambda)$, where $\alpha_n(i)$ is defined as the probability of partial state sequence until time n and state i given model λ. Now the probability of observation sequence given the model is given by

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \qquad (4.3.1)$$

This probability can be calculated inductively as

Initialization:

$$\alpha_1(i) = \pi_i * b_i(o_1), \qquad 1 \le i \le N \tag{4.3.2}$$

Induction:

$$\alpha_{n+1}(j) = [\sum_{i=1}^{N} \alpha_n(i) a_{ij}] b_j(o_{n+1}) \qquad 1 \le n \le T\text{-}1, \quad 1 \le j \le N \tag{4.3.3}$$

Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{4.3.4}$$

The induction step takes into account that state j can be reached from all states, and calculates the probability of being in state j using $\sum_{i=1}^{N} \alpha_n(i) * \alpha_{ij}$. The multiplication with $b_j(o_{n+1})$ adds observation o_{n+1} into the picture. Recursion in $\alpha_n(i)$ makes it possible to calculate $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$.

4.3.2 Viterbi Algorithm

Correct state sequence cannot be found out for a given observation sequence. Hence, an optimal sequence is sought with different definitions of optimality. The most popular solution is using Viterbi algorithm, which finds the single best state sequence as a whole, maximizing $P(q|O,\lambda)$. Let

$$\delta_n(i) = max_{q_1, q_2, \ldots\ldots, q_{n-1}} \; P(q_1, q_2, \ldots\ldots, q_{n-1}, q_n = i, o_1, o_2, \ldots, o_n|\lambda) \tag{4.3.5}$$

where $\delta_n(i)$ is the highest probability among the probabilities of all single paths, at time n, accounting for the first n observations and ending in state i.

$\delta_n(j)$ can also be written recursively as

$$\delta_n(j) = max_{1 \le i \le N} \left[\delta_{n-1}(i) * a_{ij}\right] * b_j(o_n) \quad 1 \le n \le T, \quad 1 \le j \le M \tag{4.3.6}$$

where a_{ij} and b_{ij} are defined earlier. Viterbi algorithm is implemented for $\delta_n(i)$using induction

Initialization:

$$\delta_i(i) = \pi_i * b_i(o_1), \qquad 1 \le i \le N \tag{4.3.7}$$

$$\psi_1(i) = 0 \tag{4.3.8}$$

Recursion:

$$\delta_n(j) = max_{1 \le i \le N}\left[\delta_{n-1}(i) * a_{ij}\right] * b_j(o_n) \quad 1 \le n \le T, \quad 1 \le j \le M \tag{4.3.9}$$

$$\psi_n(j) = arg\ max_{1 \le i \le N}\left[\delta_{n-1}(i) * a_{ij}\right], \quad 2 \le n \le T, \quad 1 \le j \le N \tag{4.3.10}$$

Termination:

$$P^* = max_{1 \le i \le N}[\delta_T(i)] \tag{4.3.11}$$

$$q_T^* = arg\ max_{1 \le i \le N}[\delta_T(i)] \tag{4.3.12}$$

Back Tracking:

$$q_n^* = \psi_{n+1}(q^*_{n+1}), \quad n = T-1, T-2, \ldots, 1 \tag{4.3.13}$$

where $\psi_n(i)$ is a matrix used for storage of most likely states at time n-1 that will transit to state i at time n. In other words it holds the arguments that maximize $\delta_{n+1}(j)$ for all n and j.

4.3.3 Baurn-Welch Method for Estimation

Although the model parameters (A, B, π) that maximize the probability of an observation sequence cannot be solved analytically, there are iterative methods that can locally maximize $P(O|\lambda)$. First of all the backward variable $\beta_n(i)$ is introduced, which is similar to the forward variable. Backward variable $\beta_n(i)$ is the probability of observation sequence from n+1 to the end, being guaranteed by the model λ with state i, at time n.

$$\beta_n(i) = P(o_{n+1}, o_{n+2}, \ldots\ldots, o_T|q_n = i, \lambda) \tag{4.3.14}$$

The probability of being in state i at time n is defined as

$$\Upsilon_n(i) = P(q_n = i|O,\lambda) \tag{4.3.15}$$

which can also be expressed as

$$\Upsilon_n(i) = \frac{P(q_n = i, O|\lambda)}{P(O|\lambda)} \tag{4.3.16}$$

$$= \frac{P(q_n = i, O|\lambda)}{\sum_{i=1}^{N} P(q_n = i, O|\lambda)} \tag{4.3.17}$$

Using the fact that $P(q_n = i|O,\lambda) = \alpha_n(i)*\beta_n(i)$, i.e., the probability of being in a state is equal to the combined probability of reaching the state from the start and the end.

$$\Upsilon_n(i) = \frac{\alpha_n(i)*\beta_n(i)}{\sum_{i=1}^{N} \alpha_n(i)*\beta_n(i)} \tag{4.3.18}$$

Finally, the probability of being in state i at time n and in state j at time n+1, is defined as:

$$\xi_n(i,j) = P(q_n = i, q_{n+1} = j|O,\lambda) \tag{4.3.19}$$

This can also be written using backward and forward variables as

$$\xi_n(i,j) = \frac{P(q_n = i, q_{n+1} = O|\lambda)}{P(O|\lambda)} \tag{4.3.20}$$

$$= \frac{\alpha_n(i) a_{ij} b_j(o_{n+1}) \beta_{n+1}(j)}{P(O|\lambda)} \tag{4.3.21}$$

$$= \frac{\alpha_n(i) a_{ij} b_j(o_{n+1}) \beta_{n+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_n(i) a_{ij} b_j(o_{n+1}) \beta_{n+1}(j)} \tag{4.3.22}$$

Note that the model parameters (A, B, π) can be calculated by counting the following event occurrences such as

π_j = expected frequency of state j at time n = 1

$$a_{ij} = \frac{expected\ number\ of\ transitions\ from\ state\ i\ to\ state\ j}{expected\ number\ of\ transitions\ from\ state\ i}$$

$$b_j(k) = \frac{expected\ number\ of\ v_k observations\ in\ state\ j}{expected\ number\ of\ times\ in\ state\ j}$$

Also note that $\psi_n(i)$ is the probability of being in state i at time n = 1.

$\sum_{n=1}^{T} \xi_n(i,j)$ is the expected number of transitions from state i to state j, and $\sum_{n=1}^{T} \psi_n(i)$ is the expected number of times in state i.

Using these, set of re-estimation formulas are defined as:

$$\bar{\pi}_j = \psi_1(j) \tag{4.3.23}$$

$$\bar{a}_{ij} = \frac{\sum_{n=1}^{T} \xi_n(i,j)}{\sum_{n=1}^{T} \psi_n(i)} \tag{4.3.24}$$

$$\bar{b}_j(k) = \frac{\sum_{n=1, o_n=v_k}^{T} \psi_n(i)}{\sum_{n=1}^{T} \psi_n(i)} \tag{4.3.25}$$

Starting with an initial model $\lambda = (A, B, \pi)$ and using it to compute the new model parameters $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$. It is proven that either $\lambda = \bar{\lambda}$ or $P(O|\bar{\lambda}) > P(O|\lambda)$, i.e., the new model is same as the old one or it is a better estimation. Therefore an iterative method to re-estimate model parameters by using $\bar{\lambda}$ in place of λ until there is no change in the model is feasible.

4.4 HMM Model Used for the Highlight

Description of terminology:

N – Number of scene shots

V – View type of the given shot.

O – The observation features extracted from the key frames.

$P(V_i|O_k)$ – Probability of a shot being in V_i given O_k.

V_i given the observation O_k (k = 1, …., m).

A – Transition probability matrix learned from the training data.

π – Initial state distribution which is also learned from the training data.

$P(O_k|V_i)$ – Probability of the observations in a given shot view, V_i.

$P(V_i)$ – Prior probability, of shot V_i which can be estimated from the training data.

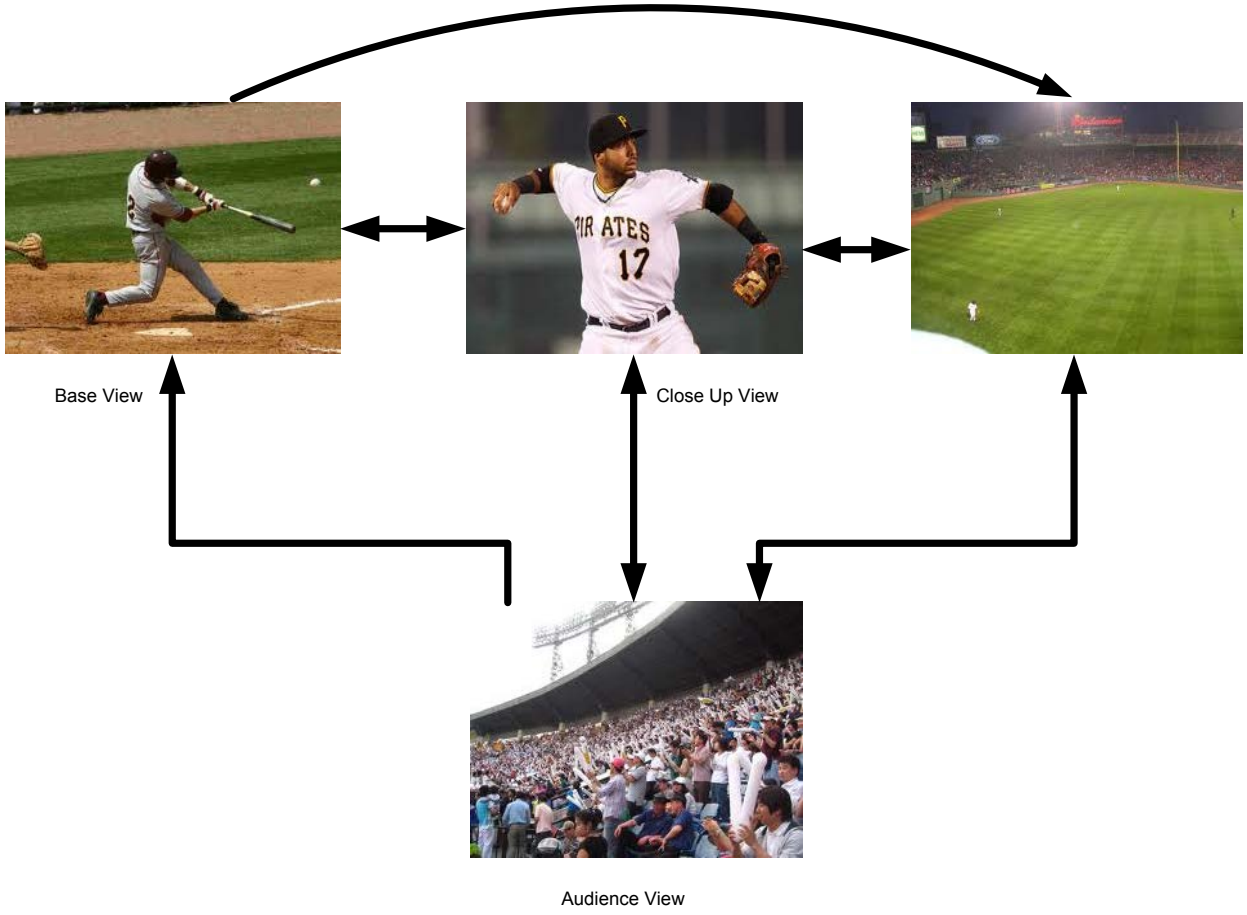

Figure 4.1 HMM Model

With all these parameters defined, I can compute the probability of a view given the features distribution using Bayesian rule

$$P\ (Vi|Ok,\ k = 1,\ldots..,\ m) = \frac{P(O_k,k=1,\ldots\ldots,m|V_i)P(V_i)}{P(O_k,k=1,\ldots..,m)}$$

4.5 Detection of a Highlight

For a given highlight I built a HMM model as given in the previous Fig. 4.1. This model takes the input as the initial probabilities and the transition probability matrix of the states. With all the inputs I can detect a particular highlight using the following method. Steps involved.

1. Divide the video into number of shots depending upon the color changes at the shot boundaries.
2. Extract the key frames from each shot and use the key frames for feature extraction.
3. From the key frames extract the features such as dominant color, texture, motion and edges.
4. For each shot these features will become the observations O_i,i = 1,2, …., N.
5. With these observations compute the classification probability P ($O_k|V_i$) for each scene shot and each view type.
6. Compute the probability of the optimal view transition sequence associated with the given observations. This probability is denoted as α_h = P (V_1, …., $V_n|O_k$, k = 1, …., m), where m is the number of features used and n is the number of shots, is the optimal view transition. This α_h can be computed with the standard Viterbi algorithm.
7. If for a given α_h, α = max (α_h) exceeds certain threshold, then I say that the highlight has been detected.

CHAPTER 5

EXPERIMENTS AND RESULTS

The whole data set I used contains three games each lasting for approximately 1 hour 20 min. The game video is in audio video interleaved (AVI) format and has a frame rate of 29 fps. The system for automated highlight extraction is designed and implemented completely in C#. Though initial experimentation was simple, poor performance was observed on a modestly complicated task, and simple tasks led to sufficient but inaccurate models. Rather than discouraging further studies, these experiments demonstrate that many factors contribute to the success of an HMM extraction system. Further experimentations will aid in identifying those factors.

Before processing the video for highlight extraction it has to divide in different shots. The identification of the shots is done at the shot boundaries. Hard cuts are easily the most common form of shot boundaries so I focused our shot boundary detection on hard cuts using color histogram method. I considered different clips of varying length in time from three different games and extracted the shots. Table 5.1 shows our experiment results I obtained for shot detection. Perfect automatic segmentation of video is hard to achieve with the current state of art.

Table 5.1 Shot Detection

Format	AVI – 352 X 240 pixels				
Frame rate	29 fps				
Sample size	24 bits				
	# of frames	Duration (seconds)	Desired # shots	Obtained # shots	Accuracy (%)
Game 1	17680	610	125	105	84.90
Game 2	12850	443	79	66	82.54
Game 3	14530	501	105	91	81.58

After shots are detected key frames are extracted from these shots which maintain the important content for the video while removing all the redundancy. With the proposed approach I extracted key frames from the shots with an accuracy of more than 94 percent.

From these key frames I extracted low level features as color, texture, motion and edges defined by MPEG -7. With these features I evaluated our HMM highlight extraction approach by classifying the shots into four view types: close up, medium, audience, pitch.

The whole data set for view classification contains thirty minutes of baseball video data of thirty five different clips. These clips are taken at random from three different baseball games. Table 5.2 summarizes the results that I obtained in classifying the shots into different views.

The data set that I took for training the HMM in highlight extraction consists of 46 clips of highlights taken at random from the games. The final result obtained from in extracting the highlight was 78 percent accurate. I believe that with a larger experimental data for training, the performance of this extraction system can be improved.

Table 5.2 View Classification

View Type	Game 1			Game 2			Game 3			Overall Accuracy (%)
	Required	Obtained	Accuracy (%)	Required	Obtained	Accuracy (%)	Required	Obtained	Accuracy (%)	
Close up	38	31	81.58	18	13	72.22	30	24	80.00	77.93
Base	21	17	80.95	16	12	75.00	19	15	78.95	78.30
Outfield	52	44	84.61	32	29	90.63	49	47	95.91	90.38
Audience	14	13	92.86	13	12	92.31	7	5	71.43	82.27

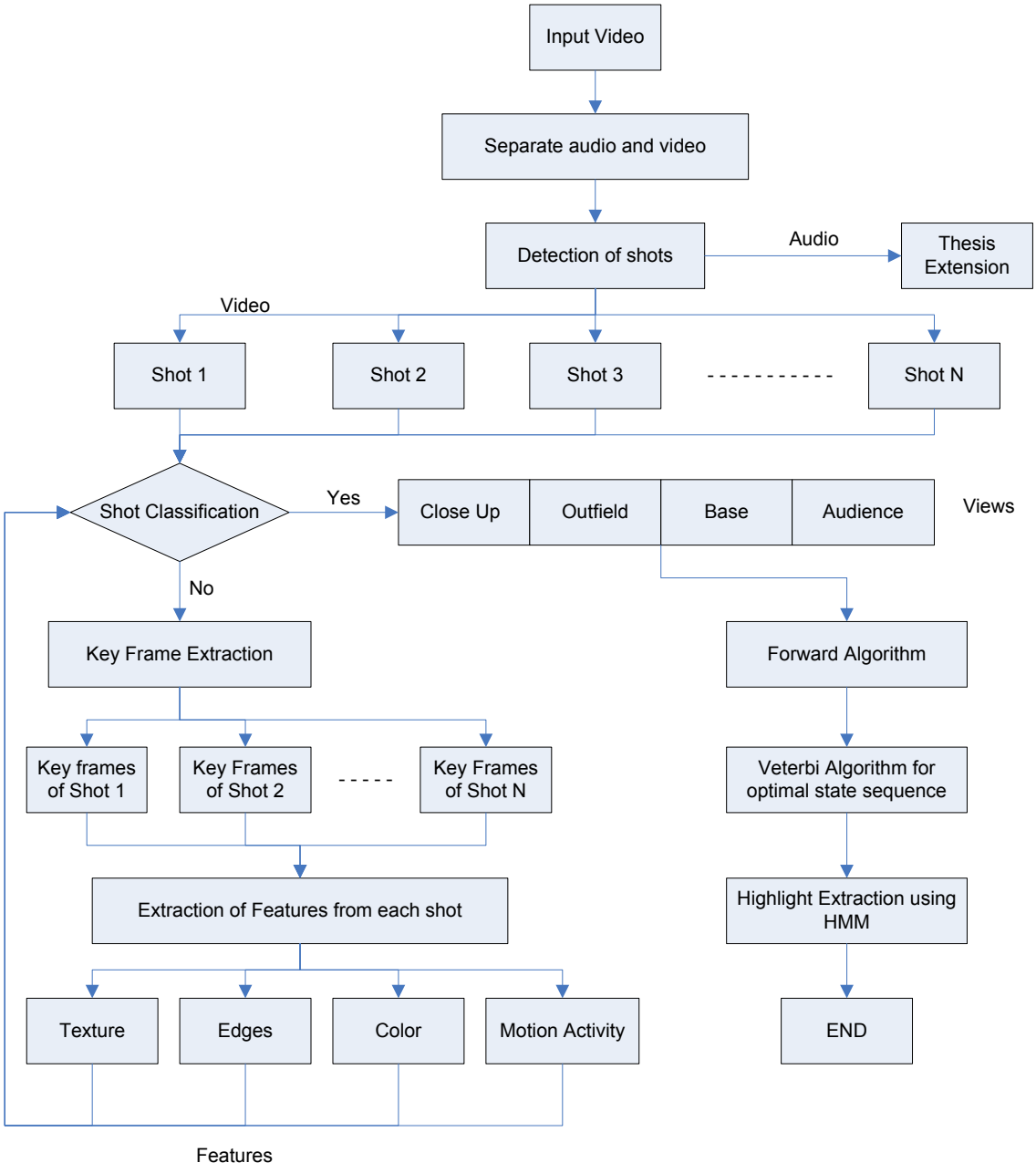

Figure 5.3 Flow Chart for Highlight Detection

CHAPTER 6

CONCLUSIONS

This thesis provides an automatic highlight extraction system using hidden Markov model (HMM). I used color, texture, motion and edges as low level features for the view classification. Base, outfield, close up, audience views were used as the states of the HMM. The HMM has been proved to be a viable method for highlight extraction. The system was trained using varying clips from different games to have proper estimate of the parameters. With a larger data sample the system performance can be improved. The proposed method is limited to only few features and views. By considering more low level features, classification of the views can be done more accurately relatively increasing the overall performance of the extraction system. Our experiments show satisfactory results. Motion and edge change ratios for shot detection and audio for view classification can be considered for the future work.

BIBLIOGRAPHY

[1] K. Mai, R. Zahib, and J. Miller, "A featured-based algorithm for detecting and classifying scene breaks ," *ACM Multimedia International conference* , vol.3, pp, 189-200, 1995.

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

[3] S. Carrato and I. Koprinska, "Temporal video segmentation: A survey," *Signal Processing Image Communication*, vol.16, no. 5, pp. 477- 500, 2001.

[4] B. Shahraray and D. C. Gibbon, "Automatic generation of pictorial transcripts of video programs," *Proc. SPIE Multimedia Computing and networking*, vol. 2417, pp. 512- 518, 1995.

[5] A. Divakaran, R. Regunathan , and K. A. Peker, "Video Summarization using descriptors of motion activity: A motion activity based approach to key-frame extraction from video shots," *Journal of Electronic Imaging*, vol. 10,pp. 909-916, Oct 2001.

[6] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content based video retrieval and browsing," *Pattern Recognition*, vol. 30,pp. 643-653, 1997.

[7] K. F. Lee, H. W. Hon, M.Y. Hwang, and X. Huang, "Speech recognition using hidden markov models: A cmu perspective," Speech *Communication, Elsevier Science Publishers B. V., North-Holland*, pp. 497-508, 1990.

[8] L. R. Rabiner, " High performance connected digit recognition using markov model," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 8,pp 1214-1224, 1989.

[9] Averbuch, L.Bahl, R.Bakis, P.Brown, G.Daggett, S. Das, K. Davies, S. D. Gennaro, P. V. D. Souza, E. Epstein, D. Fraleigh, F. Jelinek, B. lewis, R. Mereer, J. Moorhead, A. Nadas, D. Nahamoo, M. Pichney, G. Shichman, P. Spinelli, D.V. Compernolle, and H. Wilkens, " Experiments with the tangora 20,000 word speech recognizer," *IEEE International Conference on Acoustics, Signal and speech processing,* pp.701 – 704,1987.

[10] Y. L. Chow, M. O. Dunham, O. A. Kimball, M. A. Kraner, G. F. Kubala, J. Makhoul, S. Roucos, and R. M. Schwartz, "Biblos: the bbn continuous speech recognition system," *IEEE International Conference on Acoustic ,Signal and Speech Processing,* pp. 89-92, 1987.

[11] F. Jelinek, R. L. Mercer, and S. Roukos, "Principles of lexical language modeling for speech recognition," *Advance in Speech Signal processing*, pp .651 – 699, 1992.

[12] A. M. Gillies, "Cursive word recognition using hidden markov models," *Proceedings of the 5th U.S. Postal Service Advanced Technology Conference*, pp. 557 – 562, 1992.

[13] M. Gilloux, M. Leroux and J. M. Bertille, "Strategies for cursive script recognition using hidden markov models," *Machine Vision and Applications*, vol. 8, pp. 197-205, 1995.

[14] Kundu, Y. He, and P. Bahl, "Recognition of handwritten word: First and second order hidden markov model based approach ," *Pattern Recognition* ,vol. 22, no.3, pp. 283-297.1989.

[15] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini, "Offline cursive handwriting recognition using hidden markov model," *Pattern Recognition*, vol.28, no.9, pp.1399 -1413, 1995.

[16] J. Yang, Y. Xu, and S. Chen, "Human action learning via hidden markov models*," IEEE Transactions on Systems, Man, and Cybernetics- Part A: Systems and Humans*, vol.27, no.1, pp. 34 – 44, 1997.

[17] P. Symth, "Hidden markov models for fault detection in dynamic systems," *Pattern recognition*, vol.27, no.1, pp. 149 – 164, 1994.

[18] L. Rabiner, "A tutorial on hidden markov model and selected applications in speech recognitions," *Proceeding of the IEEE*, vol.77, no.2, Feb 1989.

[19] S. K. Bhattacharjee and X. U. Cabedo, "Shot detection tools in digital video," *Non- Linear model based image analysis*, pp. 231-238, 1998.

[20] J. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection technique," *IS&T/SPIE proceedings: Storage and Retrieval for Images and video Database IV*, vol.2670, pp.170 – 179, Feb 1996.

[21] C. M. Modena, R. Brunelli, and O. Mich, "A survey on video indexing," *IRST Technical Report*, 1996.

[22] D. Bhat ,K. Sakiewicz ,N. NandhaKumar, W.Chang, W.Zhao, and J. Wang, "Improving color based video short detection ," *IEEE International Conference on Multimedia Computing and Systems*, vol.2, pp. 752 – 756, 1999.

[23] M. Mitra, W. J. Zhu, R. Zahib, J. Huang and S. Kumar, "Image indexing using color correlograms" *IEEE Computer Society conference on Computer Vision and Pattern Recognition*, pp. 762 – 768, 1997.

[24] M. Orengo and M. A. Stricker, "Similarity of color images." *Proceeding – SPIE The International Society for Optical Engineering, Storage and Retrieval for Images and Videos Databases III (issues 2420)*, pp. 381 – 392 ,1995.

[25] N. Murphy, S. Marlow, C. OToole, and A. Smeaton, " Evaluation of automatic shot boundary detection on a large video test suite," The Challenges of Image Retrieval – 2nd UK Conference on Image Retrieval, 1999.

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

[27] W. Y. Lu, J. F. Yang and S. S. Hao," simplified block matching criteria for motion estimation," IEICE Transaztions on Information and Systems E series D, vol.83, no. 4, pp. 922 – 930, 2000.

[28] Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object apperances," *Visual Database System II, Elsevier Science Publisher*, pp. 113- 117, 1992.

[29] Y. Gong, C. H. Chuan and G. Xiaoyi, "image indexing and retrieval based on color histograms," *Multimedia Tools and applications*, Vol.2, pp. 133 – 156, 1996.

[30] Y. N. Deng, C. Henry, M. S. Moore and B. S. Manjunath, "Peer group filtering and perceptual color image quantization," *IEEE International Symposium on circuits and Systems*, vol.4, pp. 21 – 24,1999.

[31] X. Sun, D.Ajay, and B. S. Manjunath, "A motion activity descriptor and its extraction in compressed domain," *IEEE Pacific – Rim Conf. Multimedia (PCM),* pp. 450 – 453, Oct 2001.

[32] S. Jennin and A. Divakaran, "Mpeg-7 visual motion descriptors," *IEEE Tran Circ. Sys. Video Tech.,* vol. 11, pp. 720 – 724, June 2001.

[33] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and machine Intelligence,* Vol.8, pp. 679 – 714, 1986.

[34] Chappidi, Pradeep; Kothinti, Kalyan Reddy; Namuduri, Kameswara; "Automatic extraction of highlights from a cricket video using HMM and MPEG-7 descriptors", *Paper presented to the 1st Annual Symposium on Graduate Research and Scholarly Projects (GRASP) held at the Hughes Metropolitan Complex, Wichita State University*, April 22, 2005.

[35] www.mlb.com

[35] www.wikipedia.com