Content-Based Sports Video Analysis and Composition

A Dissertation
Presented to the School of Computer Engineering
of the Nanyang Technological University

by

Wang Jinjun

in Fulfilment of the Requirement for
the Degree of Doctor of Philosophy

2008

Abstract

This thesis proposes solutions for content-based sports video analysis, including multi-modal feature extraction, middle-level representation and semantic event detection. In addition, solutions for sports video composition and personalization are also examined.

The first part of the thesis describes our methodology to detect semantic events and event boundaries from both broadcast sports video and non-broadcast sports video. Specifically, to process broadcast sports video, our approach not only uses visual/audio features, we also analyze the web-casting text information associated with the video and synchronize it with the visual/audio features to detect event, locate event boundaries and identify involved players/teams; To process non-broadcast sports video, we select the raw unedited main-camera soccer video as the input and use visual, audio and motion features extraction with multi-level modeling to detect event and event boundaries. Our proposed techniques are evaluated using large sports video data set and achieved satisfactory user acceptance score.

The second part of the thesis introduces three novel applications based on our proposed sports video analysis techniques. The first application is a live sports highlight generation system which can automatically detect multiple events from a live sports game and extract a suitable video segment for each event to provide live and personalized game viewership via various mediums. The second application attempts to automatically generate broadcast soccer video composition from multiple raw video captures by detecting events from unedited soccer video and mimicking human director's practice to control the replay insertion and camera view switching operations. The third application is a personalized music sports video generation system to automatically select and align desired sports video scenes with music clips to generate music video clips. The three proposed systems are tested using objective evaluations and subjective user studies.

Acknowledgments

Many people have contributed their ideas, time, and energy in my pursuit of this PhD research. I wish it was possible to thank them all.

First thanks must go to my supervisor, Dr. Chng Eng Siong, and co-supervisor, Dr. Xu Chang Sheng, who have been truly inspirational throughout my candidature. I would like to thank them for all their guidance, rewarding discussions, cooperation, encouragements, and lasting support to my study and my life. I have been blessed to find in them all the good qualities of a supervisor.

I am indebted to the School of Computer Engineering, Nanyang Technology University (NTU), for offering me the PhD scholarship. I am also thankful to the Institute for Infocomm Research (I^2R) and the Centre for Multimedia and Network Technology, NTU for providing me an excellent working environment with all the much needed facilities and services, as well as financial supports including travel allowances.

Many people from NTU and I^2R have contributed to this research by exchanging their ideas, challenging my approaches, commenting on drafts, and participating in subjective user studies. I am also grateful to my thesis examiners who took time out from their busy schedule and gave valuable insights to my thesis work. In addition, I am very appreciative of the friendly help from many anonymous reviewers who helped me to improve the argument in my publications.

I thank Ms. Zhou Si Bo for being such a wonderful wife and giving me so much support and encouragement to do well during this period. She makes the whole journey so enjoyable.

Last, but most importantly, I'd like to dedicate this thesis to my Mum and Dad to express my deepest gratitude. They provided the bedrock of support on which my career is built. They are the best parents who are so willing in giving me the best in my life

even in the most difficult time in their life, without hoping for anything in return. They showed me what the true purpose in life is.

Contents

List of Figures

List of Tables

Acronym

ASR Automatic Speech Recognition

BBC British Broadcasting Cooperation

BDA Boundary Detection Accuracy

BN Bayesian Network

CC Closed Caption

CN Competition Network

DP Dynamic Programming

DVE Digital Video Effect

EPL English Premiere League

ESPN Entertainment and Sports Programming Network

FSM Finite State Machine

GCR Game Clock Recognition

GMM Gaussian Mixture Model

GSD Game Start Detection

HMM Hidden Markov Model

HT Hough Transform

LPC Liner Prediction Coefficient

LPCC Liner Prediction Coefficient Cepstral

MAD Mean Absolute Differences

MB Micro Block

MFCC Mel-Frequency Coefficient Cepstral

MSV Music Sports Video

MV Music Video

NN Neural Network

OCR Optical Character Recognition

S-League Singapore Soccer League

SBD Shot Boundary Detection

SSC Semantic Shot Classification

SVM Support Vector Machine

TNPB Temporal Neighboring Pattern Similarity

UEFA Union European Football Association

Chapter 1

Introduction

DIGITAL video has become a major information storage and exchange media in our modern era. It plays a very important role in the current multimedia computing and communication environments, with various applications in entertainment, broadcasting, education, publishing, etc. The pervasive reach of multimedia documents can be demonstrated by the following statistics: On January 2006, there are more than 469 million digital media documents created globally with 315 million of them being actively accessed [1]. *AltaVista* [2] has been serving around 25 million search queries per day, with its multimedia search featuring over 45 million images, videos and audios. People watch more than 70 million videos on *YouTube* [3] daily to see first-hand accounts of current events, find videos about their hobbies and interests, and discover the quirky and unusual through the Internet. *Google Inc* [4] in 2000 started the world's first open online video marketplace (*Google Video*) where users can search for, watch and buy an ever-growing collection of TV shows, movies, Music Video (MV), documentaries, personal productions etc. What's more, with the ability to watch and share videos worldwide through the Internet, amateurs can also capture their own moments on video and become future broadcasters [3].

The wide adoption of video media is due to its ability to convey rich semantic presentations through synchronized audio, visual and textual information. However, while video content provides rich multimedia information to users, its ever-increasing volume has also posed challenging problems on retrieval, delivery, access and editing. To manage video content more efficiently, some early researchers extended existing systems and algorithms used in image, text and sound analysis for video data. This strategy is not

always optimum for all media analysis applications as video has its own syntax, semantic, rules, and formats [5]. For example, early *Content-Based Video Retrieval* system using color-texture-shape based image analysis techniques is generally ineffective when users need to search for particular events in sports video.

Another challenging issue with video media applications is that viewers are becoming more demanding in terms of quality, quantity, and the ability to influence the presented information. For example, survey shows that there is a clear demand for viewer's interaction and possibilities to change the presented material on TV [6]. In order to address these demands, entertainment, education and broadcasting corporations are seeking new possibilities to create enhanced content to maintain their competitive advantages, and advanced video composition techniques are required. However, without efficient ways to intelligently analyze lengthy multimedia data, the task to compose video/audio materials remains labor-intensive for professionals and difficult for general public. "Video will only become an effective part of everyday computing environments when we can use it with the same facility that we currently use text" [7].

This thesis aims to develop more advanced tools for semantic video analysis and composition applications. Since a diversity of video types exists, we choose to focus on sports video. There are many reasons why analyzing sports video is important and necessary. First, sports video is widely distributed over various networks and appeals to large global audiences, hence sports video analysis and composition techniques are highly sought-after. Second, interesting segments in a sports video generally occupy only a small portion of the whole content, and viewer's interest to the video is normally decreased during uninteresting segments [8]; hence the ability to index and reorganize important segments of the sports videos to produce a condensed or customized video clip is highly desirable. Third, recognizing semantically meaningful segments from sports video is possible and easier as compared with other types of videos such as commercials or movies. This is due to the existence of well-defined content structure and domain rules in sports games. For example, a tennis match is divided into sets, and each set has several games and serves. A soccer game has two halves, and within each half there are goal, foul, injury etc events. For these reasons, efficient automatic techniques to characterize and edit sports video documents are both possible and desirable.

1.1 Problems Statement

This thesis addresses two main problems: *Content-based sports video analysis* and *Automatic sports video composition/editing*. In the first problem, we examine semantic sports video structuring and modeling techniques to detect events, event boundaries and event details from both the broadcast sports video and the raw unedited sports video. In the second problem, we focus on personalized sports video composition/editing task with quality comparable to human productions.

1.1.1 Content-Based Sports Video Analysis

The ability to automatically identify important contents from lengthy sports video documents is a key requirement in many important applications such as sports video indexing [9, 10], sports highlight generation [11, 12, 13, 14], structural analysis [15, 16, 17], etc. Semantic sports video analysis is also a fundamental technology used in automatic sports video composition/editing application. However, extracting semantics from sports video is very difficult as it is still unclear how human perceive concepts. Recent publications have reported many systems that achieve good sports video processing results without semantic understanding of the video content. For example, Wan *et al.* [18, 19, 20] proposed an interesting system to automatically insert virtual content (advertisement, logo, etc) into an existing sports video. Their system attempts to minimize the interference of the inserted content to the viewer by analyzing low-level image/video features rather than performing high-level understanding of where and when would the insertion be undesirable. In another example, Pingali *et al.* [21] created the LucentVision system which can track a tennis player's movement and ball's three-dimensional trajectory. Their system can then generate presence maps of where the players spent their time during a match, create virtual replays of ball trajectory that can be viewed from any position, and present a variety of numerical statistics. However the system is limited as it is unable to further analyze the game semantics such as what is the event, whether the player's movement is appropriate, or when a replay should be launched. Recently, some works that focus on analyzing the semantic video content to characterize sport programs have been reported [22]. However, many of these approaches are limited to recognizing only a

small set of events from certain sports domains and video types. In contrast to existing work, this thesis aims to develop more generic event analysis techniques to process both the broadcast sports video and the raw unedited sports video, using multiple-modality, multiple-level analysis techniques.

1.1.2 Automatic Sports Video Composition/Editing

Digital video editing and composition are common tasks in both film and broadcast post-production process. Although technological advances in digital video standard and computer technologies have created tools for these tasks, video editing/production remains a time-consuming and labor-intensive undertaking, which has resulted in increased research activities for this problem in recent years. Reported work include video search/skimming [23, 24, 25], video visualization [26], augmented reality [6, 19, 21, 27] meta-data creation [28], non-linear editing, video composition [29, 30, 31, 32], etc. These techniques are however not specifically designed for sports video input, *i.e.*, they utilize neither the domain-knowledge for automatic video content analysis nor the common practices in sports video production, and therefore the performance of directly applying these techniques to edit or compose sports video is limited. In this thesis, we will specifically propose solutions for automatic video composition/editing for sports applications.

1.2 Contributions

The main contributions of the thesis are to propose a framework for reliable sports video event analysis and perform automatic sports video composition. We develop three applications using algorithms/techniques introduced in the thesis to discuss actual real-world implementation issues. The followings paragraphs list the main topic discussed in the thesis:

- We propose a framework to analyze broadcast sports video for accurate and robust event detection, event boundary detection and player/team identification.
  - In this framework, textual information is utilized to assist sports video analysis. Particularly, we use the web-casting text for sports event detection and

discuss various gamecast analysis scenarios. The incorporation of web-casting text significantly improves the event detection reliability and helps to extract event semantics.

  - We also propose novel approaches to detect video event boundary. Specifically, we present several low-level visual/audio feature extraction/classification algorithms and introduce a robust synchronization method between the web-casting text and the visual/audio features. We describe two video event boundary detection algorithms by modeling the visual/audio features in ranges given by the synchronized text. The selected visual/audio features are generic for broadcast sports video of different sports types.

- We propose a non-broadcast sports video event detection and event boundary detection framework. We use the raw unedited soccer video to demonstrate the functionality of our algorithms. We examine methods to generate several mid-level representations to bridge the gap between low-level features and high-level events. These mid-level representations are statistically modeled and classified to detect high-level event and event boundary.

- Finally, we discuss three sports video composition/editing applications using the proposed techniques. The three applications are:

  - A live soccer highlight generation system: The system detects events from a live sports game video and generate customized highlight in near real-time.

  - A broadcast soccer video composition generation system: We use our non-broadcast sports video event detection techniques to analyze raw unedited soccer video and compose broadcast soccer video by mimicking human director's operation.

  - An automatic Music Sports Video (MSV) composition system: We propose a fully- and semi-automatic video/music matching algorithm to select and align sports video scenes (events) and music clips.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 reviews the related work in content-based sports video analysis.

Chapter 3 proposes a broadcast sports video event detection framework. We use visual/audio/textual features extraction and synchronization to detect event, event boundaries and event details (*e.g.* player or team involved) from broadcast sports video. Experimental results are listed to validate the accuracy and robustness of the proposed method.

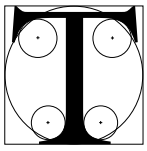
Chapter 4 presents a non-broadcast sports video event detection framework. We use visual and audio features extraction and multi-level modeling to detect event and event boundaries from the raw unedited main-camera soccer video. Various visual/audio feature extraction and modeling methods are presented to create several mid-level representations. These mid-level classifications are then used to perform automatic event detection and event boundary detection. Several experiments are conducted to evaluate the performance of various modules in the system.

Chapter 5 introduces three interesting sports video composition and editing applications based on the techniques proposed in chapter 3 and chapter 4. The performance of each system is evaluated using both objective criteria and subjective user study.

Finally, chapter 6 presents the conclusions and discusses new frontiers of the content-based sports video analysis and composition research.

Chapter 2

Related Works



HE methods proposed in the current literature for content characterization of sports documents could be addressed in various ways: The first possible classification could be based on the type of sports considered, such as baseball [11, 14], soccer [33, 34, 35], tennis [36, 37], basketball [38], Formula 1 car races [39, 40], snooker [41], fighting [42], etc. Another possible classification method could be to consider the key methodology used. We will use the second classification method to briefly review the related literature. We first examine the features commonly used by researchers and then review some rule-based and statistical modeling methods. Finally various application areas related to sports video analysis and processing are discussed.

2.1 Feature Extraction

2.1.1 Visual Feature Extraction

In image/video processing research, features such as edge [43], texture [44], etc are widely adopted. As features however do not possess high-level semantics, they are refereed to as "low-level" features. In content-based sports video analysis literature, researchers usually use these low-level features to extract semantic information for high-level analysis. The extracted semantic information is referred to as "middle-level" features. The following subsections review some common "middle-level" visual features.

2.1.1.1 Semantic Shot Classification (SSC)

In typical broadcast video, a shot is a sequence of frames captures by a single camera in a single continuous action. It is the most basic unit of video data. Broadcast sports video often intertwines different shot types. A quantitative study by Assfalg *et al.* [34] covering more than ten hours of sports video, including gymnastics, field-and-track, car race, ball game, etc, revealed that three types of shots/scenes are most prevalent: the playing field view, the player view, and the audience view. In addition, the transition between the types of views is closely related to the semantic state of the game; hence SSC is extremely useful for sports video analysis. Assfalg *et al.* proposed to use Neural Network (NN) classifier and edge, segment and color features to distinguish these three view types. Other researchers have also reported many different SSC approaches. For example, Denman *et al.* [41] recognized the "full-table" shots in snooker videos from the second order moment of the Hough Transform (HT) parameter space; Ekin *et al.* [35] used dominant color region detection and Golden-Section spatial composition to classify the soccer video into long shot, in-field medium shot, close-up shot and out-field shot; Wang *et al.* [45] proposed to model both the playing field color and the players' uniform color to classify the soccer and basketball game video shot. Xu *et al.* [16] computed the ratio of grass pixel count to frame size to classify each frame in soccer video into either far view, medium view or close-up view; Duan *et al.* [46] presented a generic sports video SSC framework that was able to classify seven different tennis shot types, nine different basketball shot types and nine different soccer shot types.

2.1.1.2 Replay Scene Detection

Replay scenes in broadcast sports videos are excellent indicators of semantically important segments. Hence replay scene detection is useful for many sports highlight generation applications. The most common characteristic of replay scenes are that they are usually played in a slow-motion manner. Such slow-motion effect is produced either by repeating frames of the original video sequence or by playing the sequence captured from a high-speed camera at the normal frame rate. The first slow-motion effect, *i.e.*, by repeating frames, will result in the presence of still and shift frames in the replay video. These frames will generate unique feature patterns on the frame differences [35, 47], macroblock

types [48], vector flow, encoding size [49], etc, and hence can be used to detect replays. However, if the replays are produced from high-speed camera, the above-mentioned methods cannot be used. Hence other replay detection techniques have been proposed. For example, Nitta and Babaguchi *et al.* [50, 51] used Bayesian Network (BN) together with six textual features extracted from Closed Caption (CC) to detect replay shot; Pan *et al.* [52] proposed to detect the transition effect, *e.g.* flying-logo, before and after the replay segments to detect replay; Tong *et al.* [53] introduced a method to automatically discover the flying-logo for replay detection; Mihajlovic and Babaguchi *et al.* [40, 54] attempted modeling the "Digital Video Effect (DVE)" to locate pairs of DVEs for replay identification.

2.1.1.3 Player/Ball Tracking

In ball games such as basketball or soccer, the ability to understand the movements of players and ball is essential to analyze the tactics used in the matches. This requirement can only be satisfied if the players/ball can be accurately tracked. Tracking the positions of players and ball has been attempted from videos with both moving cameras [33, 55, 56] and fixed cameras [57, 58, 59, 60]. The methods proposed include template [59] and color matching [55], HT [60] and, for fixed camera input, background subtraction. For fixed camera work on soccer videos, partial coverage is obtained using two [57] and four [59, 61] cameras, complete coverage requires eight cameras [58]. The object being tracked includes base ball [62], tennis ball [63, 64, 65], soccer ball [56] or player [66] etc.

2.1.1.4 Sports Field Shape Extraction

The ability to identify certain shapes from the sports field can facilitate the extraction of middle-level features, *e.g.*, to segment the video [33, 67], to reduce noise in signal [68], and to locate anchor marks in sports field [6, 69]. Due to the diversity of sports games, most reported works are specially designed for a certain type of sports. For example, for tennis game, Sudhir *et al.* [70] extracted tennis court lines of four different classes, namely carpet, clay, hard and grass, together with the players' location information to analyze high-level tennis events. For soccer game, Wan *et al.* [18] used HT based method to detect the soccer goalmouth and modeled the viewer relevance for virtual

content insertion; Gong *et al.* [33] enhanced the edges from field-line and center circle and introduced a signature method to identify four different locations in soccer field; Ekin *et al.* [35, 15] detected the three parallel lines in the soccer penalty area for "Goal scoring" event detection; They also proposed an adaptive dominant color region detection model to segment the region of soccer pitch for event detection; Kang *et al.* [71] divided the soccer field into 15 symmetrical areas and applied field-line detection, center circle detection and goalmouth detection to distinguish between these regions. Hayet *et al.* [72] used 2D template registration based on vanishing point to recognize the soccer pitch and tennis court shapes and localize the field of play.

2.1.1.5 Motion Parameters Extraction

Motion is a useful visual cue available from video and is invariant to changes in color and lighting. Motion features include motion histogram, dominant motion, global motion parameters, etc. Many researchers focus on using global motion parameters for analysis because the global motion, either caused by big object movement or camera motion, is closely related with the game action. Algorithms for fast global motion estimation have been reported for compressed [73] and un-compressed video [74]. Some example of researches using global motion for sports video analysis include: Huang *et al.* [75] extracted the motion histogram, dominant motion and global motion estimation to classify different TV programs including news reports, weather forecasts, commercials, live basketball games, and live football games; Ma *et al.* [76] used "Motion Texture" to distinguish amongst several sports program such as diving, high jump and racing; Xiong *et al.* [13] combined the MPEG-7 intensity of motion activity descriptor with audio features to generate sports highlights; Peker *et al.* [12] applied MPEG-7 motion activity descriptor to recognize game highlight from golf and soccer; Wan *et al.* [77] used motion direction and motion intensity information to predict the soccer play-region.

2.1.2 Audio Feature Extraction

There are certain game-specific audio signals such as applause, booing and whistling that are possible indicators of important events occurrence. Hence some researchers have used the audio data to identify high-level semantics. For instance, Leonardi *et al.* [9] computed

the "loudness" feature to find instances of goal scoring in soccer game; Wan *et al.* [78, 79] extracted the frequency-domain audio feature to locate exciting segments in soccer and tennis videos; Huang *et al.* [75] used 6 audio features, namely, "Root mean square volume", "Zero crossing rate", "Pitch Period", "Frequency Centroid", "Frequency Bandwidth" and "Energy Ratio" to discriminate among news reports, commercials, weather forecasts, football videos and basketball video clips; Xiong *et al.* [80] compared the sports video classification performance using Mel-Frequency Coefficient Cepstral (MFCC) features and MPEG-7 audio descriptors.

However, due to the semantic gap between the low-level features to high-level events, some researchers do not directly use these low-level features but rather create a mid-level audio representation known as sound event [81] or audio keyword [82] for the event analysis problem. For example, Zhang *et al.* [81] used rule-based fusion method on MFCC, Liner Prediction Coefficient (LPC) and normalized energy features to detect sound events from basketball video for high-level event analysis; Rui *et al.* [11] detected several sounds such as excited human speech and baseball hitting to catch the highlight segments from baseball video; Xu *et al.* [82] used Support Vector Machine (SVM) with MFCC and Liner Prediction Coefficient Cepstral (LPCC) features to classify the audio segments into several audio keywords, *e.g.* acclaim, commentator speech, etc, to detect event from basketball, tennis and soccer video.

2.1.3 Textual Feature Extraction

Besides the above mentioned visual and audio features, textual information available in some broadcast video has also been examined. For some cases, textual information can serve as useful cues because text usually contains rich semantics and is easy to analyze. For example, one textual information, called the CC, has been successfully used in news video analysis [83, 84]. The following subsections discuss some common textual features available for sports video analysis.

2.1.3.1 Closed Caption

Closed Caption (CC) is a text version of the spoken part of a television, movie, or computer presentation. Although CC was developed to aid hearing-impaired people, it

is nevertheless useful to the general public. For instance, CC can be read when audio cannot be heard, either due to a noisy environment, such as an airport, or because the location restricts the broadcast of audio source such as in a hospital [85]. CC is also useful in sports video analysis. For example, recently researchers have utilized the CC to extract baseball highlights [10], index plays and players [51], and identify semantic structure of sports videos [50]. However, CC is available only in certain countries and its application is thus limited.

2.1.3.2 Automatic Speech Recognition

In sports broadcasting, there is usually a reporter commenting the play, and hence a transcript of his/her speech would be useful for sports video content analysis. Some researchers have proposed to use Automatic Speech Recognition (ASR) system to obtain the text transcripts for content analysis. For example, Chang *et al.* [86] used a keyword spotting system to detect a small set of keywords such as "touchdown", "fumble", etc to detect related event in baseball game; Ariki *et al.* [87] proposed a robust ASR system to produce live CC from sports video commentary and then use keyword matching for highlight scene retrieval.

However, as the live commentary audio recordings usually are very noisy, the ASR result on the recording is often poor and thus is not useful. Hence many other researchers have proposed to only classify sound segments to detect events. This group of research was reviewed in the "Audio Feature Extraction" subsection (subsection 2.1.2).

2.1.3.3 Caption Text Recognition

In sports broadcasting there are always texts that can be recognized from the video frame. These detected texts can be divided into two classes [40]: Scene text such as billboards, text on vehicles, writings on human clothes, etc, and Caption text that is mechanically superimposed over video frames to supplement the visual and audio content. The caption text usually contains much useful information about the game, and many researchers have proposed techniques to detect and recognize the caption text to assist sports video analysis. For example, Assfalg *et al.* [34, 88] defined heuristic rules to detect caption, *e.g.*, the caption must remain stable for a period of time for the audience

to read, and caption should have high luminance contrast, etc; Mihajlovic *et al.* [40] applied horizontal differential filtering to locate the caption area from shaded area and use pattern matching to recognize the caption text to detect highlights from Formula 1 race video; Chen *et al.* [89] used SVM to identify single text line in video frames and perform Optical Character Recognition (OCR) for text recognition; Zhang *et al.* [90] proposed to detect the caption text change event to identify "score" and "last pitch" event in baseball game; Li *et al.* [91] used Connected Component Analysis to segment the caption area and detected the periodicity of caption text change to recognize the superimposed game clock time. Shih *et al.* [92] introduced a robust caption extraction, localization, recognition, and interpretation method to understand the game score, team name, etc game semantics.

2.1.3.4 Web-Casting Text

Web-casting text from the Internet is fast becoming an alternate text source for broadcast sports program. The web-casting text is a running text commentary of a game created by private individuals or cooperation. It possesses very detailed information about the event, related players/team and approximate event time, and hence some researchers have attempted using web-casting text for sports video analysis. For example, Li *et al.* [93, 94, 95, 96] proposed a generic sports video indexing framework by play data detection, scoreboard OCR, web-casting text analysis and the synchronization of these three information; Xu *et al.* [97] utilized the time-stamp in web-casting text to synchronize the text event and the event from visual/audio analysis. Examples of web-casting text can be found in [98, 99, 100, 101].

2.2 Modeling Method

To obtain the required semantic information, a high-level modeling module can be used to classify the visual/audio/text representations. There are two categories of high-level modeling modules [102]: *Rule-based approach* and *Statistical approach*. The following subsections discuss these two categories.

2.2.1 Rule-Based Approach

The rule-based approach uses domain knowledge to define rules to achieve semantic video classification [16, 38, 46, 51, 71]. For example, in [51] keyword sequences that were predefined based on heuristics were used to detect highlight such as "touchdown". Duan *et al.* [46] explored the possibility of defining rules to map mid-level visual and audio representations into high-level semantic events; Xu *et al.* [16] defined heuristic rules to classify the frame transitions to extract the high level structure information and to obtain the play/break status of the game.

The clear advantage of the rule-based approach is the ease to insert, delete, and modify existing rules for a defined video type. However, when the video type changes, the rules must be manually recreated. Hence, algorithms to automatically create rules as per human inspection, and to discover relationships that are not so obvious would be useful. For example, in [38], an entropy-based inductive tree-learning algorithm was utilized to establish the trained knowledge base. This knowledge base was represented as a decision-tree with each node in the tree being an if-then rule.

2.2.2 Statistical Approach

The statistical approach uses statistical machine learning method to classify the semantics of sports video by discovering non-obvious correlations among different video patterns. The following subsections introduce several statistical modeling algorithms commonly reported in literature:

2.2.2.1 Hidden Markov Model

The popularity of Hidden Markov Model (HMM) based methods for sports video analysis can be attributed to the HMM's ability to model the spatio-temporal patterns that sports content possesses [103]. To list some examples, in [75], low-level audio, color and motion features were separately classified using HMM classifiers to discriminate among commercials, football game, basketball games, news reports and weather forecasts. In [104], the HMM classifier used scene type, camera motion and player position features to recognize events such as "Penalty", "Free Kick" and "Corner"; In [105], a HMM based framework for basketball video content analysis using a weighted motion energy distribution

as the input feature was proposed; In [106], a three level HMM classifiers framework using several HMM systems to introduce constraints of higher level semantics was argued; In [107, 108], Xie *et al.* presented HMM based algorithms for automatic feature selection for unsupervised structure discovery from sports video sequences. They proposed a unified framework that used fully unsupervised statistical techniques for automatically discovering salient structures, and simultaneously recognizing such structures in unlabeled data. Xu *et al.* applied a HMM based method to model the basketball, soccer and volleyball game in a hierarchical structure such that semantics with different granularity can be accurately detected.

2.2.2.2 Support Vector Machine

The successful use of Support Vector Machine (SVM) [109] in multimedia documents classification is due to SVM's excellent generalization ability. Examples of sports video analysis using SVM include: Ma *et al.* [76] used SVM to model the motion pattern to distinguish between different video clips. In [11], a robust speech endpoint detection technique in noisy environment was proposed using a combination of generic sports features and baseball-specific features as inputs to the SVM to identify excited speech; In [46] and [82], the SVM classifiers were used to create mid-level sports audio keywords. When SVM cannot distinguish all the required classes in one pass, a hierarchical SVM system could be applied [82]. Sadlier *et al.* proposed a set of audio/visual features that were generic across soccer, rugby, hockey and Gaelic football games and applied SVM to detect events from these games.

2.2.2.3 Neural Network

Neural Network (NN) based approach has also been reported by some researchers. For example, Assfalg *et al.* [34] extracted the edge, segment, and color features for two NN classifiers to classify between different shot types; Kobla *et al.* [48] used NN to model the frequency and orientation information in wavelet coefficient to segment text region from sports video.

2.2.2.4 Other Algorithms

Besides the above mentioned algorithms, there are other supervised learning methods reported, *e.g.*, Controlled Markov Chain [9, 110], Maximum Entropy [111], Naive Bayes classifier [112], Bayes Belief Network [113] and Bayesian Network (BN) [114, 51, 50].

2.3 Application

The above-mentioned feature extraction and modeling methods are combined to solve different application problems. The following sections summarize some common applications in content-based sports video analysis.

2.3.1 Structural Analysis for Sports Videos

2.3.1.1 Play-Break Detection

Sports video is composed of play events, which refer to the times when the ball is in-play, and break events, which refers to intervals of stoppage in the game. Many researchers have reported play-break detection methods. For example, in [15], the play-break was detected by thresholding the duration of the time interval between consecutive long shots; Xu *et al.* [16] segmented the soccer game into play-break by classifying the respective visual patterns, such as shot type and camera motion, during play, break and play/break transition; In [17], six HMM topologies were trained for segmentation and classification of play and break in one pass.

2.3.1.2 Highlight Detection

The sports highlight detection research aims to automatically extract the most interesting segments, also known as game highlights, from the full-length sports video. The highlight portions in sports video can usually be distinguished by the detection of certain low-level feature patterns, *e.g.* the occurrence of replay scene [47], the excited audience or commentator speech [11, 78, 79], certain camera motion [12, 115] or certain highlight event related sound [11, 81]. In [116] Hanjalic introduced a generic game highlight detection method based on exciting segment detection. The exciting segment was detected using empirically selected cinematic features from different sports domains and weighted

combine to obtain a excitement level indicator. Tjondronegoro *et al.* [117] detected the whistle sounds, crowd excitement, and text boxes to complement previous play-breaks and highlights localization method to generate more complete and generic sports video summarization. Generally speaking, the sports video highlight detection task is often associated with sports video event detection. The difference between the two is that the later further recognizes the types of event in the detected game highlight.

2.3.1.3 Event Detection

Event detection research aims to recognize interesting or significant events automatically from sports video and to generate textual indexes to label the events. Currently this research forms the largest body of content-based sports video analysis work. Initially, most event detection methods used only single modality, but in recent years, multi-modal analysis techniques have been widely proposed for event detection [118]. To list some examples, the scene cuts and camera motion parameters were used for soccer event detection in [9] where the author showed that unreliable detection would occur if limited cinematic features were used; The camera motion and object-based features were employed in [104] to detect soccer events; A mixture of cinematic and object descriptors is used in [38, 37] to index basketball video; CC information and visual features were integrated in [10] for event-based football video indexing; In [40, 111], the audio information was used jointly with video/text feature for content characterization of sports video; In [40], audio, video and superimposed text annotation information were utilized to extract highlights from TV Formula 1 program.

2.3.2 Content Adaptation and Enhancement

2.3.2.1 Enhanced Sports Broadcasting

Since the introduction of digital video capturing and processing techniques, there has been a series of attempts to enhance the content of transmission for sports video broadcasting [6]. Examples include adding virtual information such as athletes information, rules and game statistics [6, 69, 21] and virtual advertisements [19] to the content. For example, in [6, 69], several virtual contents, *e.g.* the running speed, the distance between two points, the athletes statistics, etc, were created and superimposed upon the original

video to enhance broadcasting; Pingali *et al.* [21] created a tennis broadcasting system called LucentVision to provide multiple video feeds for individual broadcasters. The system can include additional information such as overlay drawing, reformat the output to fit inside other content, and customize for a broadcaster based on language or look and feel.

2.3.2.2 Video-Based 3D Reconstruction of Sports Game

In [57], a soccer game reconstruction system named SoccerMan was presented. SoccerMan can create an animated 3D scene from a video sequence of a soccer game to enable users to examine the game from any virtual viewpoint. In [119] Koyama *et al.* reconstructed the virtual stadium, players and their actions scene at the client-end, *e.g.* web-casting end, using multi-camera input; In [6], a tennis broadcasting system that could reconstruct the tennis court, player, ball, etc, scene according to the client's profile was proposed. In these above mentioned techniques, the 3D reconstruction parameters can be independently transmitted through various networks, and they have been included in MPEG-4 standard. The 3D athlete information can be used for rendering in a totally virtual environment or mixed real scenario for simulation or entertainment purpose [120].

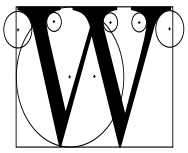
2.3.2.3 Automatic Sports Video Composition

While most current researches for semantic sports video analysis mainly focus on annotation, indexing, summarization and retrieval for sports video, these techniques do not address automatic video editing and production problem. The automatic video composition/editing problem however has been studied for other types of video. For example, in [32] Rui *et al.* introduced an automatic lecture video capturing and broadcasting system by following professional online lecture video composition rules and applied both hardware and software techniques to implement these rules, such as speaker tracking, camera positioning and shot transitions; In [29, 121], Hua *et al.* described a home video editing system using Genetic Algorithm to select an optimal set of home video clips to match a given music; In [30] and [122], a Music Video (MV) editing system called "Hitchcock" was developed to automatically analyze an general video type and select suitable segments for a given music clip; Intelligent video editing tools like muvee [31] are already available for both professional and amateur users.

Automatic editing of sports video is achievable through accurate semantic analysis of original sports video material and the broadcasting syntax. A few examples related to automatic sports video editing include [123, 67]. In [123] Ariki *et al.* used multiple fixed cameras around the soccer pitch to perform event detection such that the composed soccer video can zoom out to the detected action automatically. In [67], we developed a system to recognize events from raw unedited soccer video. Our technique was then extended to perform automatic replay for soccer broadcasting. We also proposed a Greedy algorithm based video/music matching method to automatically compose "Music Sports Video (MSV)" [124], and further improved the video/music matching method in [125] to reduce processing time and enable semi-automatic [122] video composition operation.

Chapter 3

Event Detection from Broadcast Sports Video



ITH the proliferation of sports broadcasting, the ability to identify interesting events from lengthy and voluminous sports videos is becoming more valuable. Traditional methods detect events using low-level visual/audio feature with high-level heuristic or statistical modeling [22]. These methods' performances are however limited due to several reasons: First, the information gap between features and semantics limits the event detection accuracy and robustness; Second, the event boundary detection problem is not well defined, and hence may lead to undesired video segments being included in the extracted event clips; Third, it is difficult to recognize event details using only visual and audio feature analysis, *e.g.*, details such as who scored an identified "goal-scoring" event in a soccer match and how it was scored; And fourth, some events have very similar visual/audio feature patterns and cannot be easily distinguished from each other, *e.g.*, a "yellow/red cards" events from a serious "foul" event.

The limitation of traditional approaches motivates us to use additional cues for analysis. One possibility is to incorporate textual information associated with sports video. In this chapter we present an event detection approach using text analysis to detect event and text/video alignment to identify event boundaries in the video. Compared to existing approaches, our proposed method achieves better event detection and event boundary detection performance due to the additional textual information. In addition, our method can identify more event details such as the involved players/teams.

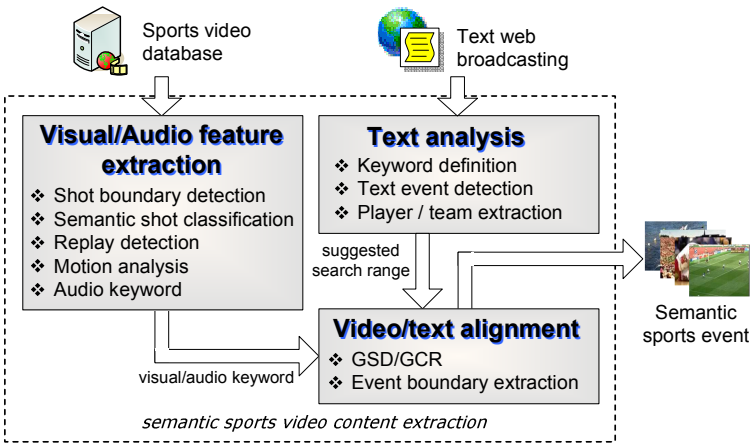


Figure 3.1: A multi-modality event detection framework

Figure 3.1 shows our proposed event analysis framework. It consists of a text analysis module to detect event, a visual/audio feature extraction module to generate mid-level visual/audio representations, and a video/text alignment module to detect video event boundary. The following sections elaborate the details of these three modules.

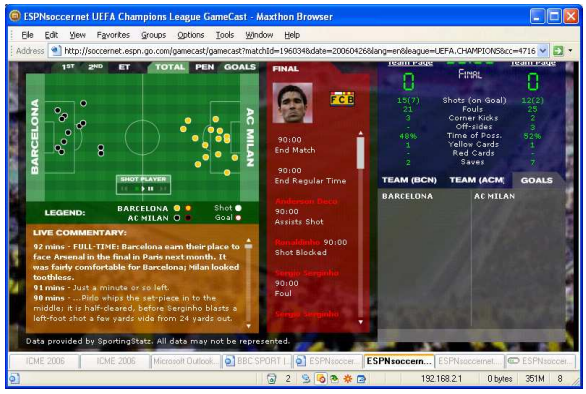
3.1 Text Analysis

In sports broadcast media, textual information can be extracted from caption text [34], Automatic Speech Recognition (ASR) of commentator speech, Closed Caption (CC) [10, 50, 126], game logs [97], web-casting text [93], etc. Previous researchers have used textual information to assist sports video analysis, for example, the CC was utilized to extract baseball highlights [10], index plays and related players [126] and identify semantic structure of sports videos [50]; In [34], the caption text was detected to annotate sports video documents; Xu *et al.* [97] utilized the game log information to refine their visual/audio feature based sports event detection system.

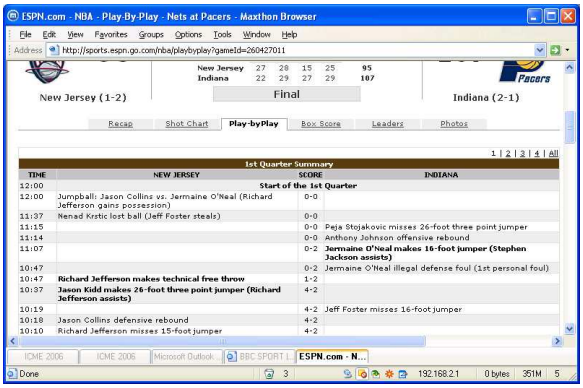
Compared with extracting textural information from caption text or ASR outputs which are heavily affected by video/audio quality, or using CC that is only obtainable in certain countries, web-casting texts are more freely available as live gamecast/matchcast [98, 99], live text commentary [98, 100] or match report [127]. It is a running text commentary of a game created by private individuals or cooperation. In addition, web-casting text

possesses very detailed information about the event, related players/team and approximate event time (Figure 3.2). Because of these advantages, we propose to use web-casting text as the source of textual information for our system. In the following subsections, the methods to detect event and extract player/team information from the web-casting text are presented.

(a) BBC Sport [100]



(b) ESPN Soccernet [98]



(c) ESPN NBA [101]

Figure 3.2: Web-casting text examples

3.1.1 Event Detection from Web-Casting Text

Our algorithm detects desired events from broadcasting texts using keyword matching based search techniques. We classify broadcasting texts into two classes, namely well-structured and free-style text class. For each type of text class, a method is proposed to process it. The details of each method are given below:

The well-structured web-casting text has defined syntax structure and vocabulary, *e.g.*, "*Foul by Jose Antonio Reyes (Arsenal) on Moreno Josico (Villarreal)*", "*Direct free kick taken right-footed by Sanjuan Quique Alvarez (Villarreal) from own half*", etc, and the occurrence of each event is indicated by only one or two different nouns. If these nouns are found in the web-casting text, the relevant event is detected. Table 3.1 lists the event keyword definition for the well-structured web-casting text from [100]. Such keyword definitions are easily extendable.

Table 3.1: Keyword definition for a well-structured web-casting text

Events	definition	keyword
card	serious foul involving yellow/red card shown	dismissed, sent off, booked
foul	referee whistled serious misconduct	foul
goal	goal scored	goal, scored
offside	player caught in an offside position	offside
freekick	direct or indirect freekick fired	free kick, free-kick
save	save made by the goal-keeper	save, blocked
substitution	substitution of players	substitution, replaced

In the free-style web-casting text, the language used is less formal and appears more like a transcript of the commentator speech, *e.g.*, "*Fyssas (Greece) commits a foul after challenging Ronaldo (Portugal), Figo (Portugal) takes the corner*", "*Luis Figo shows his skill, holding on to the ball on the left flank under pressure from three defenders, and wins the set-piece*", etc. Hence, attempting to detect event keywords from the free-style text is more difficult due to the following reasons: First, the broad vocabulary used in free-style web-casting text may not match our desired keyword. Second, the presence of a keyword does not guarantee the occurrence of the related event, *e.g.*, commentary during a game break. To reduce false detection problem, Babaguchi *et al.* [51] suggested searching both the event keyword and the accompanying verb. This idea is also applied in our free-style web-casting text analysis. Table 3.2 lists the keywords defined for the free-style web-casting text from [127] using dtSearch grammar [128]; dtSearch is a text searching tool which supports stemming (tense), phonic (dialect), fuzzy (wrong-spelling) and boolean (noun and companying verb) searching [128].

Table 3.2: Keyword definition for a free-style web-casting text

Event	Keyword
Card: serious foul involving card shown	"yellow card" or "red card" or "yellowcard" or "redcard" or "yellow-card" or "red-card"
Foul: referee whistled serious misconduct	(commits or by or booked or ruled or yellow) w/5 foul[1]
Goal: goal scored	g-o-a-l or scores or goal or equalize - kick
Offside: player caught offside	(flag or adjudge or rule) w/4 (offside or "off side" or "off-side")
Freekick: direct or indirect freekick fired	(take or save or concede or deliver or fire or curl) w/6 ("free-kick" or "free kick" or freekick)
Save: save made by the goalkeeper	(make or produce or bring or dash or "pull off") w/5 save
Injury: player injury	injury and not "injury time"
Substitution: substitution of players	substitution

Note that the method to process free-style web-casting text can also be used to analyze well-structured web-casting text. In practice, the well-structured web-casting text analysis method is applied on known well-structured text to speed up processing, and the free-style web-casting text analysis method otherwise.

3.1.2 Player/Team Extraction

In addition to detecting text event, the players involved in the event can also be identified from the text information. For this purpose, a player/team database is built by analyzing the start-up line information from the web-casting text. During detection process, when a text event is recognized, we further match every word in the text event entry with the names in the database to find the relevant players and teams. This strategy is illustrated in Figure 3.3.

3.2 Visual/Audio Analysis

The second module of the event detection framework (Figure 3.1) is the "visual/audio feature extraction" module whose function is to generate suitable mid-level visual/audio

[1] In dtSearch's grammar, w/5 means the set of words on the left side, *i.e.* \commits", \by", \booked", \ruled", \yellow" occurs within 5 words' distance from the right side word, *i.e.* \foul".

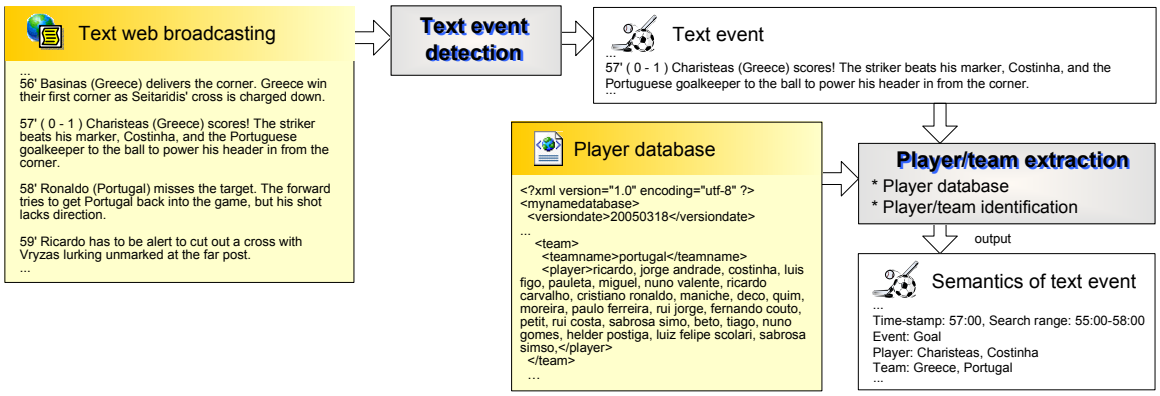


Figure 3.3: Text event detection and player/team extraction

representations for the "video/text alignment" module.

In broadcast sports video analysis, the selection of which visual/audio features to extract is usually affected by specified game rules and/or common broadcast practice. Although various visual/audio features have been examined in literature, to improve generality of our proposed framework, the five representations listed in table 3.3 are selected for modeling because they are available from most broadcast sports video of different sports types. Their associated analyses are described in the following subsections.

Table 3.3: Visual/audio analysis description

ID	Domain	Value
F_1	Visual	$shot\ boundary's\ frame\ number$
F_2	Visual	$\{view\ class\} \in \{FA,\ IM,\ OM,\ IC,\ OC\}$
F_3	Visual	$0/1$
F_4	Visual	$\{pan, tilt, zoom, direction, magnitude, entropy\} \in R^6$
F_5	Audio	$\{keyword\} \in \{whistle,\ acclaim,\ noise\}$

F_1: Shot boundary detection, F_2: Semantic shot classification (FA: Far view, IM: In-field medium view, OM: Out-field medium view, IC: In-field close-up view, OC: Out-field close-up view), F_3: Replay detection, F_4: Camera motion, F_5: Audio keyword

3.2.1 Shot Boundary Detection (SBD)

A shot is a sequence of frames captures by a single camera in a single continuous action. For broadcast video, the shot can be regarded as a basic analysis unit. Two types of shot boundary exist in broadcast soccer video, specifically the hard-cut shot boundary that is

used during play to depict the fast pace game action, and the gradual shot change such as dissolves normally used during game breaks or lull. To detect these two types of shot boundaries, we adopt the algorithm proposed in [129, 130] which operates in the following manner: For hard-cut detection, we compute the Mean Absolute Differences (MAD) of gray level pixels from successive frames, and use an adaptive threshold to decide the frame boundaries of abrupt shot changes. To handle gradual shot change, we additionally compute multiple pair-wise MAD and compare them against the adaptive threshold. Specifically, for each frame f_i, we calculate its pair-wise MAD with frame f_{i-k}, where $k = 7, 14, 20$ have been empirically chosen. The obtained shot boundary feature is F_1 (Table 3.3), a sequence of frame numbers. Each number indexes a boundary between two successive shots.

3.2.2 Semantic Shot Classification (SSC)

The transition between different types of shot in soccer video reveals the state of the game, hence the SSC feature provides a robust feature for soccer video analysis. Figure 3.4 illustrates our method for shot classification [131]. First we extract the dominant color region from each frame. If the region takes more than 2/3 of the whole frame size, the frame is regarded as an in-field view, otherwise it is classified as an out-field view. For an in-field view, we further calculate the non-dominant-color object size to classify the frame into in-field far view/medium view/close-up view if the object size is small/medium/big. An object size is deemed small if it contains less than 0.08% pixels of the whole frame size (i.e., 80 pixels for a 352×288 frame size), medium if between 0.08% 0.6% pixels, and big otherwise. Similarly, for an out-field view, we perform the edge detection to classify the frame into out-field medium view if the number of edge pixels is greater than 7% pixels of the whole frame size. Otherwise it is regarded as an out-field close-up view. These threshold values are selected empirically according to our collected soccer video. In this manner, five semantic shot classes can be identified: far view, in-field medium view, in-field close-up view, out-field medium view and out-field close-up view. The generated SSC feature is denoted as ID F_2 (Table 3.3). F_2 is a sequence with each element indicating the shot class label of the corresponding frame.

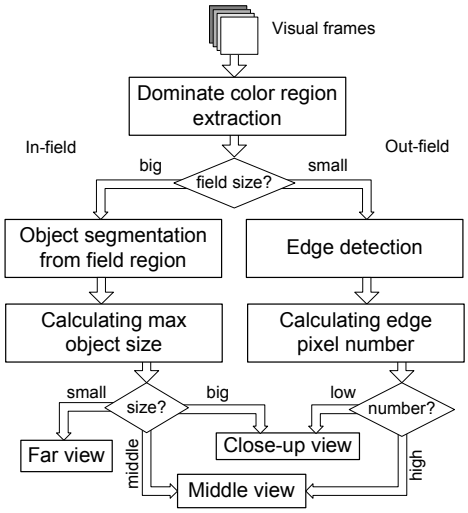


Figure 3.4: Frame classification algorithms

3.2.3 Replay Detection

In a typical broadcast soccer game, the director would normally launch a replay for interesting events. Hence replay detection greatly facilitates the soccer video analysis [131]. Existing techniques used for replay detection can be categorized into two classes: Detecting the editing effect, *e.g.*, flying-logo [52], and detecting the occurrence of slow-motion [47, 49]. Based on our observation, more than 80% of the current broadcast sports videos use flying-logo to launch replays. Hence in our system, we detect replays using flying-logo template matching technique in R, G, B channels. The detected replay/non-replay state of each frame is denoted by value 1 and 0 respectively and is collected as a sequence in feature ID F_3 (Table 3.3). Since the proposed replay detection approach requires the presence of flying-logo, it may not be applicable to those replays without such editing effect. We have therefore attempted other methods to detect these replays. For example, in [131], we analyzed the transition pattern of shot classifications (F_2) to detect replays.

3.2.4 Camera Motion

As the camera normally focuses on the player or the ball, the camera motion will provide a useful cue to represent the activities of the game. In our system, the "motion vector" field information from the compressed video (*e.g.*, MPEG-I/II) is used to estimate the

camera motion parameters. In compressed motion video such as MPEG I/II, the frame being compressed is divided into Micro Block (MB)s, and for each Micro Block (MB), the reconstructed reference frame is searched to find the Micro Block (MB) that best matches the one being compressed. The offset is encoded as a "motion vector". Hence the "motion vector"s are pre-computed during the video compression process, and they are readily available for efficient motion analysis. Our camera motion analysis module extracted six global motion parameters, specifically the "average motion magnitude", the "motion entropy", the "dominant motion direction", the "camera pan factor", the "camera tilt factor" and the "camera zoom factor". To compute these parameters, a texture filtering is first applied to remove the motion vectors from low-textured Micro Block (MB)s as these vectors may not reflect the true camera motion [132]. The surviving motion vectors are then used to compute the camera pan p_p, tilt p_t and zoom p_z motion using algorithms introduced in [73] as follows: Suppose the coordinate of the i^{th} MB in each frame is $\boldsymbol{c}_i$, its coordinate in the estimated frame is $\boldsymbol{c}'_i$ and the motion vector is $\boldsymbol{\mu}_i$, we have

$$\boldsymbol{c}_i = \boldsymbol{c}'_i + \boldsymbol{\mu}_i \tag{3.1}$$

and

$$\bar{\boldsymbol{c}} = \frac{1}{G}\sum_{i=1}^{G} \boldsymbol{c}_i \qquad \bar{\boldsymbol{c}'} = \frac{1}{G}\sum_{i=1}^{G} \boldsymbol{c}'_i \tag{3.2}$$

where G is the total number of MBs selected in the current frame,

$$p_z = \frac{\sum_{i=1}^{G}(\boldsymbol{c}'_i - \bar{\boldsymbol{c}'})^T(\boldsymbol{c}_i - \bar{\boldsymbol{c}})}{\sum_{i=1}^{G} \parallel \boldsymbol{c}_i - \bar{\boldsymbol{c}} \parallel^2} \tag{3.3}$$

$$\begin{pmatrix} p_p \\ p_t \end{pmatrix} = \frac{\boldsymbol{c}'}{p_z} - \bar{\boldsymbol{c}} \tag{3.4}$$

The average motion magnitude p_m is computed by:

$$p_m = \left\| \frac{1}{G}\sum_{i=1}^{G} \boldsymbol{\mu}_i \right\|_2 \tag{3.5}$$

The dominant motion direction p_d is computed by:

$$p_d = \arg\max_{j=0,\ldots,7} \{hist(j)\} \tag{3.6}$$

where $hist()$ is the histogram of the motion vectors quantized to 8 direction bins ($j = 0, ..., 7$). The motion entropy p_e is obtained by

$$p_e = -\sum_{j=0}^{7} \frac{hist(j)}{G} \log_2(\frac{hist(j)}{G}) \tag{3.7}$$

The generated camera motion feature is denoted as ID F_4 (Table 3.3), an R^6 vector sequence.

3.2.5 Audio Keyword

There are some significant game-specific sounds that are indicative of important events. Hence the ability to detect the special audio classes is useful for high-level semantic analysis. In our system we use the Support Vector Machine (SVM) to classify Mel-Frequency Coefficient Cepstral (MFCC) and Liner Prediction Coefficient Cepstral (LPCC) features to generate three audio keywords for broadcast soccer audio, namely "whistle", "acclaim" and "noise". The respective definition of each keyword is as follows: "whistle", the sound from the sports whistle blown by the referee to indicate misconduct or to start/pause/resume/stop the game; "acclaim", the enthusiastic approving and praising sound made by the audiences; "noise", all the rest of the sound available in soccer game, including "commentator speech", "music", "player/coach/audience sound", etc. The generated audio keyword is denoted as ID F_5 (Table 3.3). F_5 is a sequence with each element indicating the audio keyword label of the corresponding frame.

3.3 Alignment of Text Event and Video

This section discusses the video/text alignment algorithm of Figure 3.1. Upon the detection of a text event, our system examines the visual/audio mid-level representation sequence to locate the video event boundaries. We call this task video/text alignment. Some previous researches have attempted similar work, *e.g.*, Babaguchi *et al.* [51] matched the shot sequence located during the time interval given by text (CC) analysis with example sequence to identify the sports highlight as well as the highlight's boundary. Li *et al.* [95, 93] performed scoreboard OCR to generate synchronization points for detected video events and used Dynamic Programming (DP) to find best matching between these

synchronization points with successive time-stamps in the web-casting text to obtain event semantics. Their methods are limited as they required the time tag to be accurate and well synchronized with the video. Although CC can satisfy this requirement, the availability of CC is limited. Hence, similar to [95, 93], we select the web-casting text as the source of textual information in our system as it is widely available from the Internet. However, web-casting text has synchronization problems with the video. For example, a web-casting text logs a basketball dunk event at "10 min 34 sec (2nd Quarter)" of the game. Although the time-stamp of the event is given, it is difficult to identify the video frame corresponding to the given time-stamp. Another disadvantage of web-casting text is that information is usually collected from amateur websites, and hence may have erroneous or non-precise time-stamping. For example, one amateur web-casting text records an offside event at "33 minute". Without more precise time information, the algorithm needs to search an entire minute of video for the offside event which usually lasts less than 15 seconds. In another example, we observed that the time-stamp of an "injury" event is "39 minute" while the actual recorded game clock in the video shows that the event occurred during 37 minute 40 second to 38 minute 19 second. To avoid these problems, Li *et al.* [95, 93] selected the type of web-casting text with accurate time-stamp information, performed scoreboard OCR to generate synchronization points for the video, and used Dynamic Programming (DP) to match between the detected video play and text event. However, as video event analysis result is less reliable than text analysis, the sequential matching result between video play and text event may depress the overall accuracy.

Two issues need tone addressed in the text/video alignment module, they are: How to synchronize the web-casting text to the video and how to extract a suitable video segment boundary for the event. In the following subsections, we introduce a robust alignment algorithm to first synchronize the game time with the video and then locate a suitable video event boundary.

3.3.1 Synchronization of Web-Casting Text and Video

The most direct way to synchronize the time-stamp of web-casting text with video is to link each physical video frame to the game time, *e.g.*, frame 33145 represents game time "21 min 43 sec". One possible solution is to recognize the digital game clock superimposed

in each video frame. Since the same game series usually shares a single game clock template for broadcasting, an intuitive idea is to use template-matching based method for game clock recognition. However, it is observed that the location/color/size/transparence of the same template varies among different broadcasters. For example, the game clocks shape in EPL 2005 game videos have small variation among different matches although the same template is used. In addition, the time when the game clock appears in the broadcasting is different, some right after the game start, some 2+ minutes later. For these reasons, an alternative method is to identify the frame representing the game start and label the subsequent frames according to a fixed increment rate. Identifying the game start can be achieved by modeling the transition of cinematic features before, during and after game start, which is invariant against different broadcasters. However, the method is suitable only for games without frequent clock stoppage such as soccer game and is not feasible for games that have many pause/resume such as basketball. Another possibility for web-casting text/video synchronization is to learn the game clock template of each game for recognition. Although the game clock recognition method is applicable for both games with or without frequent game stoppages, this method is computational expensive and not always successful due to poor game clock resolution or occlusion, *e.g.*, by the TV channel logo. To achieve satisfactory synchronization, our system combines the above two methods. First we perform Game Start Detection (GSD) and then Game Clock Recognition (GCR) during the game, *i.e.* we use the detected game start to synchronize the web-casting text with the video before the game clock is recognized and then use the recognized game clock for synchronization. The following subsections discuss our GSD and GCR implementations.

3.3.1.1 Game Start Detection

The GSD module detects the physical frame number corresponding to the start of the match in a broadcast sports video. GSD is a crucial task for sports event detection systems as it links the time-stamp of a text event to an exact frame. For a live system, GSD can also help to reduce the computational cost by suppressing the event detection processing before the game start. In the following paragraphs we introduce our GSD algorithm on broadcast soccer video recording.

Our GSD locates the game start in soccer video by detecting the change of shots in the video. From our soccer video database, we observed that:

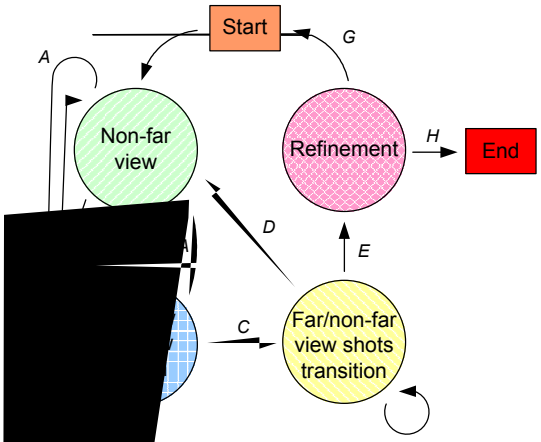
- Before the game start there are many commercials, player/team/stadium statistics, player/referee entrance etc. Such scenes are successive non-far view shots.
- When the game is about to start, a still far view focusing on the center of the soccer pitch will appear.
- Once the game begins, the camera will start to pan/tilt/zoom to follow player's and ball's movements. This results in a certain amount of camera motion.
- During the game, the shot will alternate between far and non-far view scenes.

From the above observations, we propose to detect the game start by modeling the shot type (F_2, Table 3.3) transitions and camera pan motion (F_4) using Finite State Machine (FSM). The FSM has been used to model temporal transition patterns by many previous researchers. For example, Bertini *et al.* [133] used FSM to model playfield zone change to detect soccer highlights; Leonardi *et al.* [110] applied the FSM to exploit the time sequence of the low-level visual descriptors to detect semantic sports events.

The structure of our FSM for game start detection is illustrated in Figure 3.5. The module starts from detecting a far view shot (the "non-far view" state) and sequentially jumps to the "far view with low motion", "Far/non-far view shots transition" and "refinement" states with given conditions listed in Figure 3.5. In the "refinement state", our module selects the frame before the FSM jumps into the "Far/non-far view shots transition" state as the start frame of the video. Figure 3.6 shows an example of how our FSM operates to find the game start frame. The top row of Figure 3.6 shows example video frames with the corresponding middle row indicating the state of the FSM. The bottom row shows the transition conditions as explained in Figure 3.5.

3.3.1.2 Game Clock Recognition

Figure 3.7 shows four examples of the appearance of the digital game clock used in broadcast soccer video. It can be seen that the location/color/size/transparence of the



: FSM for game start detection

Transition condition far view frame is encountered; B: a far view frame is encountered; C: a fa ame with high camera pan motion is encountered; D: undesired view trans ern is encountered; E: the desired far view -> non-far view -> far view trans ttern is encountered; F: cannot decide yet; G: restart to detect the st e second half; H: all required detection is done.

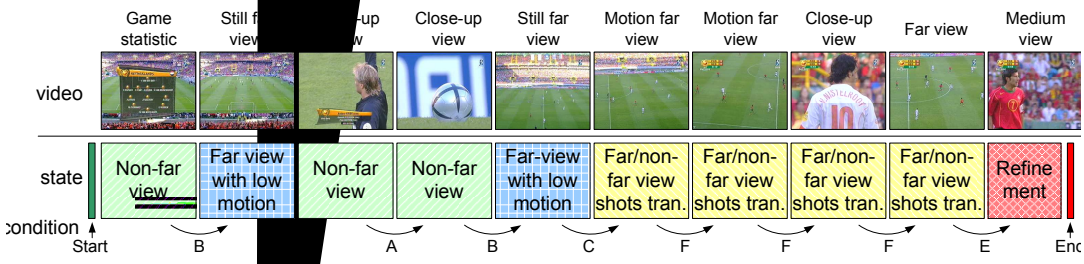
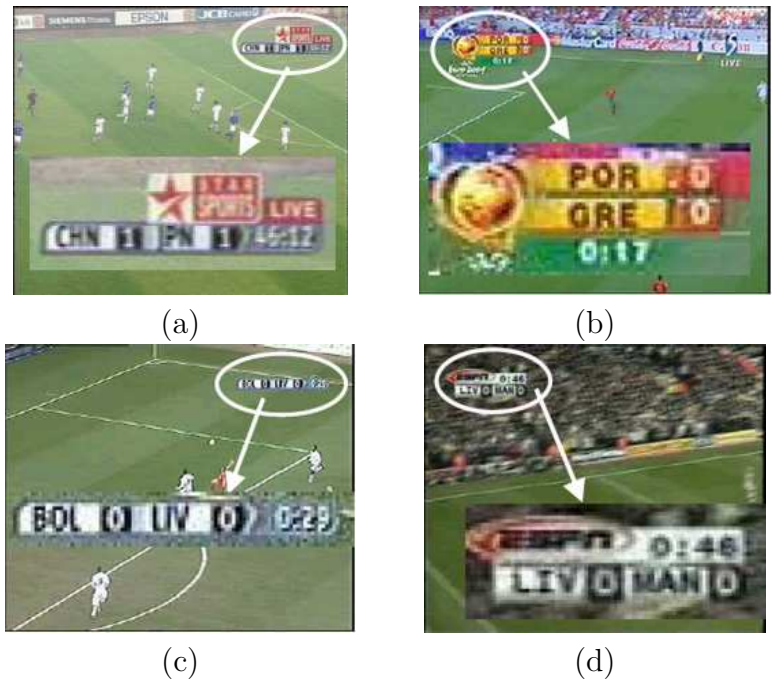


Figure 3.6: mporal pattern modeling for game start detection

A–H: e transition conditions as listed in Figure 3.5

clock can differ widely. The digital game clock is recognized using the I^{2}R [134] media research group's Temp al Neighboring Pattern Similarity (TNPB) method [91]. The method extracts the TNPB feature to obtain a template of each clock digit number and apply template matching method to recognize the game time. The TNPB feature is selected because it is periodic in nature and thus can be used to segment the clock digit from the background. As the templates are extracted directly from the game video, the template matching method can achieve superior recognition accuracy over traditional Optical Character Recognition (OCR) methods which require prior knowledge of the

game clock's appearance.



(a) (b)

(c) (d)

Figure 3.7: Game clock examples
(a) clock digits and other text in different size and color; (b) transparent clock overlay; (c) small and blur clock digits; (d) blur clock digits in different color and location

3.3.2 Video Event Boundary Detection

By GSD and GCR processing, the time-stamp of each event in the web-casting text can be labeled to a physical video frame index. If the time-stamp of an event is accurate, the video event boundary can be easily detected by searching neighboring frames. However, as discussed previously, the recorded time-stamp is not always accurate, and therefore a precise reference time to search for video event boundaries is not available. In our implementation, the system only uses the time-stamp to suggest a search window and the problem of event boundary detection is to locate the correct boundary within this window (Figure 3.8.a). Searching for a desired segment in a specified duration is similar to the problem of event detection using visual/audio analysis where the search for events is conducted on a full-length video instead. As the search range is reduced by text analysis in our system, more accurate and faster detection can be achieved as compared to searching the full-length video.

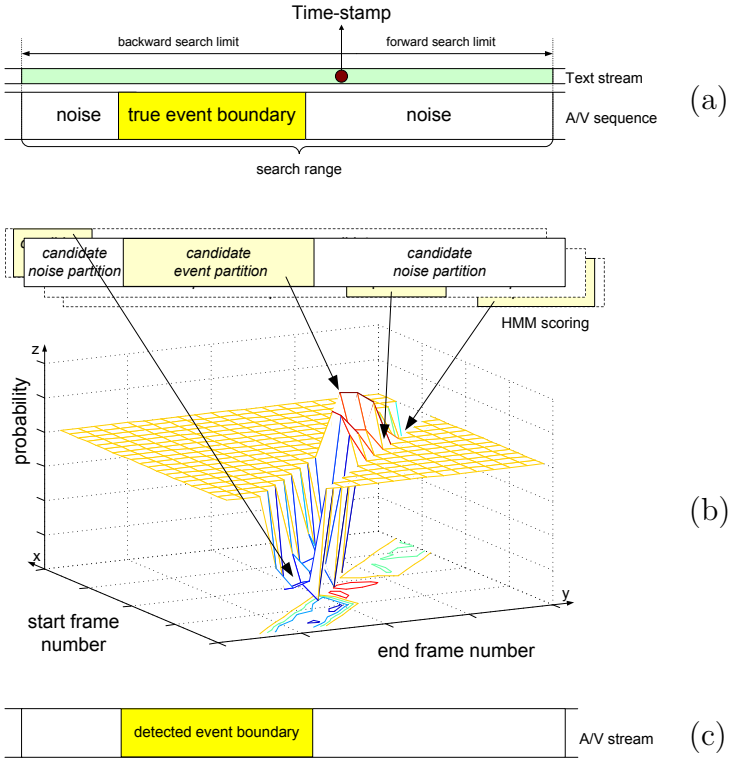


Figure 3.8: Visual/audio sequence and text stream
(a) The relationship between the time-stamp and the event boundary; (b) The search range is decomposed into different "noise-event-noise" partitions, and each partition is converted to a probability score of being the true "noise-event-noise" partition by HMM; (c) The partition that results in the highest probability score gives the detected event boundary

As identical events possess similar temporal patterns in the visual/audio feature sequence, we propose a Hidden Markov Model (HMM) based approach to model the F_1 to F_5 features (Table 3.3) from consecutive video frames to recognize the event boundaries in the video. The HMM is chosen as it has been successfully applied to similar problems. For example, Takagi *et al.* [135] proposed a HMM based system to classify different sports video; Xie *et al.* [17] used a hierarchical HMM to analyze the soccer video structure; And we have also investigated using HMM for soccer event recognition [124, 136].

The HMM classifier works in the following manner [137]: Given a set of states $S = \{s_1, s_2, ..., s_K\}$ and an observation sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, the likelihood of $\mathbf{X}$ with

respect to a HMM with parameters Θ expands as $p(\mathbf{X}|\Theta)$ and:

$$p(\mathbf{X}|\Theta) = \sum_{all\ Q} p(\mathbf{X}, Q|\Theta) \tag{3.8}$$

where

$$p(\mathbf{X}, Q|\Theta) = p(\mathbf{X}|Q, \Theta)p(Q|\Theta) \qquad (Bayes) \tag{3.9}$$

We have

$$\begin{aligned} p(\mathbf{X}|Q, \Theta) &= \prod_{n=1}^{N} p(\mathbf{x}_n|q_n, \Theta) \\ &= b_{q_1\mathbf{x}_1} \cdot b_{q_2\mathbf{x}_2} \cdot ... \cdot b_{q_N\mathbf{x}_N} \end{aligned} \tag{3.10}$$

and

$$p(Q|\Theta) = \pi_i \cdot \prod_{n=1}^{N-1} a_{q_n q_{n+1}} \tag{3.11}$$

$Q = \{q_1, q_2, ..., q_N\}$ is a (hidden) state sequence where each $q_i \in S$; $\pi_i = p(q_1 = s_i)$ is the prior probabilities of s_i being the first state of a state sequence; a_{ij} denotes the transition probabilities to go from state i to state j, and $b_{q_i\mathbf{x}_i}$ is the emission probabilities. $b_{q_i\mathbf{x}_i}$ is modeled by Gaussian Mixture Model.

To apply the HMM method to the alignment problem, a search model with a "noise-event-noise" structure in visual/audio stream is created (Figure 3.8.a) using three HMM models, one for the beginning noise (*Noise* HMM_1), one for the event (*Event HMM*) and one for the ending noise (*Noise* HMM_2). In our implementation, the three HMMs have similar left-right structure with different hidden state and Gaussian Mixture Model (GMM)s for each state (Table 3.4). In addition, different events use different subset features of $\{F_2, F_3, F_4, F_5\}$ (Table 3.3) found empirically (Table 3.4).

During recognition process, the three HMMs were concatenated with the grammar [138] illustrated in Figure 3.9. Two alignment search experiments were conducted. However the results using these two grammars were not satisfactory (accuracy below 51%); the detected event durations were either too short or too long.

To improve performance, we consider the use of shot count feature from as typically the shot count is proportional to the duration of an event. The number of shot in a segment can be easily calculated from the SBD feature (F_1 in Table 3.3) by counting the number of shot boundaries. However, as the rest of the mid-level visual/audio feature vectors (F_2 to F_5) are based on frame while the shot count is calculated from a segment, the shot count feature could not be easily incorporated into our HMM implementation.

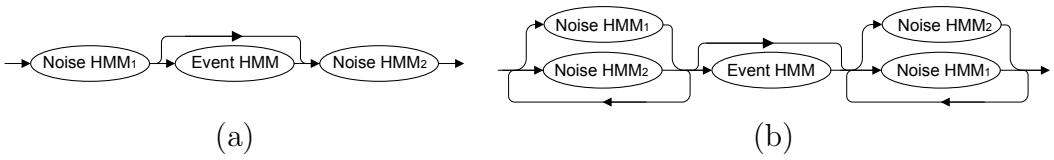


Figure 3.9: HMM grammars [20]

*(a): (noise*1 [*event*] *noise*2 *); (b): (* $< noise1 \mid noise2 >$ [*event*] $< noise2 \mid noise1 >$*)*

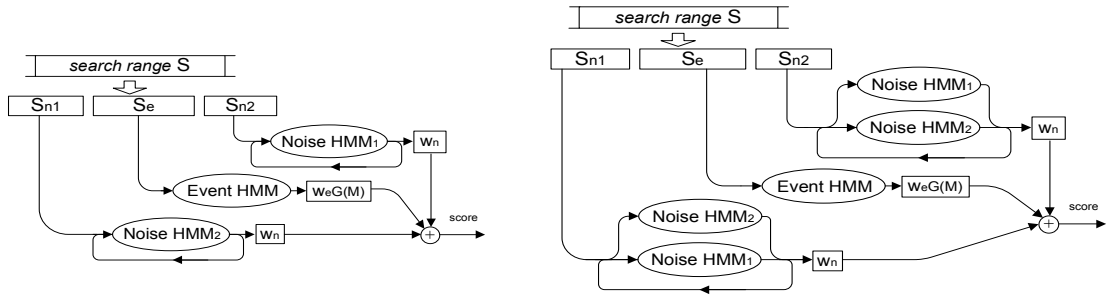


Figure 3.10: Probability score combination strategy

To include the shot count information, we propose to use the three HMMs as three separate probability classifiers and exhaustedly search all possible "noise-event-noise" partitions of the search duration to find the optimum segmentation as event boundary. The details of our method is as follows: As illustrated in Fig.3.10.a, the selected duration of the A/V feature sequence is broken into three non-overlapping shot-segments as shown in Figure 3.8.b. Each segment can contain several shots and acts as an input to each HMM. Let the feature sequences extracted from these three segments ("noise-event-noise") be denoted as $\mathbf{X}_{n1}$, $\mathbf{X}_e$ and $\mathbf{X}_{n2}$, and Θ_{n1}, Θ_e and Θ_{n2} be the parameters of the *noise* HMM_1, *event HMM* and *noise* HMM_2 respectively. We evaluate the combined probability score $P(\mathbf{X}|\Theta)$ of the three HMMs as

$$\begin{aligned} p(\mathbf{X}|\Theta) &= w_{n1}p(\mathbf{X}_{n1}|\Theta_{n1}) + w_eG(M)p(\mathbf{X}_e|\Theta_e) \\ &\quad + w_{n2}p(\mathbf{X}_{n2}|\Theta_{n2}) \end{aligned} \tag{3.12}$$

where $\mathbf{X} = [\mathbf{X}_{n1}, \mathbf{X}_e, \mathbf{X}_{n2}]$ and $\Theta = \{\Theta_{n1}, \Theta_e, \Theta_{n2}\}$. The variables w_{n1}, $w_eG(M)$ and w_{n2} are the weights used to combine the probability scores from the three HMMs. If $w_{n1} = w_eG(M) = w_{n2} = 1$ and the three segments used to compute Eq.(3.12) is the same as that found by Viterbi decoding using model of Fig.3.9.(a), then the probability

computed would be the same. In practice, the noise segments $\mathbf{X}_{n1}$ and $\mathbf{X}_{n2}$ do not present distinguishing temporal patterns, hence a higher weight should be assigned to $p(\mathbf{X}_e|\Theta_e)$. To incorporate shot count information in the probability evaluation, $G(M)$ is introduced to model shot count in the candidate event segment, specifically

$$G(M) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(M-\bar{M})^2/2\sigma^2} \tag{3.13}$$

where $\bar{M}$ and σ represent the mean and standard derivation of the shot count from our training samples of events.

To find the event boundary, all possible segmentations are evaluated using Eq.3.12, and the one that results in the highest score is selected as the event boundary as shown in Figure 3.8.c. Table 3.4 lists the detailed parameters to compute Eq.3.12. Specifically, column two of Table 3.4 lists the selected feature sets for the eight soccer events of Table 3.2, column three and four indicate the number of states and Gaussian Mixtures for the HMMs used for each event. These settings generated the best results on our data set. Figure 3.10.b illustrates a grammar which allows self-jump between the noise HMMs. Our experimental results show that the self-jump structure improves performance.

Table 3.4: Details of HMM structure

Event	Feature (Table 3.3)	Event HMM		Noise HMM	
		#States	# GMM	# States	# GMM
card	F_2, F_3	5	3	5	3
foul	F_2, F_3, F_5	5	3	5	3
goal	F_2, F_3, F_5	5	3	4	4
offside	F_2, F_3	5	3	5	3
freekick	F_2, F_3, F_4	5	6	7	6
save	F_2, F_3, F_5	5	3	5	3
injury	F_2, F_3	5	2	4	2
substitution	F_2, F_4	5	3	5	5

All the HMM models have left-right structure.

Fig.3.11 shows the examples of probability scoring for two events. In these examples, the x/y axis represents the absolute start/end time (frame number) of the event partition, and the z axis is the normalized probability $p(\mathbf{X}|\Theta)$.

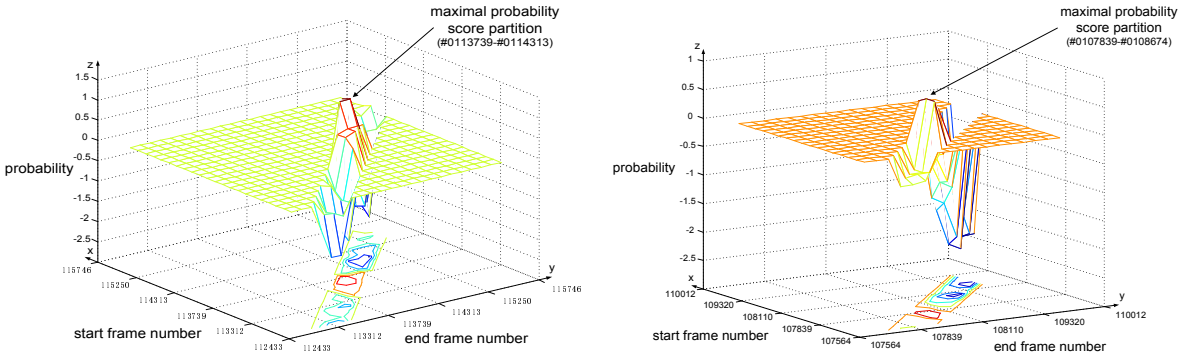


Figure 3.11: Probability scores examples

3.4 Experimental Results

In the following experiments, a broadcast soccer video data set containing seven World-Cup 2002 games, four Euro-Cup 2004 games, four Union European Football Association (UEFA) 2005 games and six English Premiere League (EPL) 2005/06 games with all correspondent web-casting text files collected from the Internet were used. The videos were in MPEG I format, totally 37 hours long. The following subsections describe the performance of text analysis and video/text alignment respectively.

3.4.1 Accuracy of Text Analysis

In our collected data, the web-casting text of the UEFA 2005 and EPL 2005/06 games belonged to the well-structured web-casting text category (Section 3.1.1) and the rest were the free-style web-casting texts. The performance of our proposed two text analysis methods is listed in Table 3.5 and Table 3.6. It is observed that the web-casting text with well-structured syntax gave better results than the free-style web-casting text.

3.4.2 Accuracy of Video analysis

3.4.2.1 Shot Boundary Detection (SBD)

We implemented the algorithms proposed by Amir in [130] for the SBD module. 75% of the shot boundaries were correctly detected from our soccer video database. More detailed performance of the method is discussed in [130].

Table 3.5: Text event detection based on well-structured web-casting text

Text event	Precision	Recall	Text event	Precision	Recall
goal	100%	100%	red card	100%	100%
shot	97.1%	87.2%	yellow card	100%	100%
save	94.4%	100%	foul	100%	100%
freekick	100%	100%	offside	100%	100%
corner	100%	100%	substitution	100%	100%

Table 3.6: Text event detection based on free-style web-casting text

Text event	Precision	Recall	Text event	Precision	Recall
card	96.7%	96.7%	foul	99.3%	97.9%
goal	89.2%	94.7%	save	98.9%	89.7%
offside	100%	97.2%	freekick	99.6%	100%
injury	100%	100%	substitution	95.8%	100%

3.4.2.2 Semantic Shot Classification (SSC)

Totally 60 minutes video from the World-Cup 2002 games and Euro-Cup 2004 games was manually labeled to train our semantic shot classification module. The test result over the rest of the data set is listed in Table 3.7. Errors are mainly due to the game noise such as unbalanced luminance, shadow, caption, etc.

Table 3.7: Precision of shot classification

Class	FA	IM	OM	IC	OC
Precision	94.5%	89.7%	87.7%	76.7%	92.7%

FA: Far view; IM: In-field medium view; OM: Out-field medium view; IC: In-field close-up view; OC: Out-field close-up view:

3.4.2.3 Replay Detection

The template matching technique for replay detection described in subsection 3.2.3 is applicable to our soccer video data because these videos all execute the flying-logo effect. Our system achieved 91% accuracy. The error is due to the absence of the flying-logo in the original broadcast video.

3.4.2.4 Camera Motion

The motion features are directly extracted from the MPEG I/II motion vector field. It is objective and no more experiment was carried on it.

3.4.2.5 Audio Keyword

To evaluate the accuracy of the audio keyword generation module, three audio classes were defined: "Acclaim", "Whistle" and "Noise". 30 minutes of soccer audio data were labeled and segmented into 20ms frames, and each frame was classified into one of the three classes. In this experiment, 50%/50% was used as training/testing data set. The accuracy for each class was: acclaim 93.8%, whistle 94.4% and noise 96.3%, respectively.

3.4.3 Accuracy of Video and Text Alignment

3.4.3.1 Game Start Detection (GSD)

Our GSD module detected the game starts (both the start of the first half and the second half) within 15 seconds after the true game start for 72% of all the game videos, 20% within 30 seconds, and the rest above 30 seconds. Some of the detected game start were delayed due to either the presence of captions which caused incorrect shot view type classification (F_2 in Table 3.3), or the occurrence of early events which lead to game pause and thus mislead the GSD module. As the accuracy of GSD module were not always high enough (*i.e.*, error within 20 seconds), it was only used to calculate the game time before the GCR module could recognize the game clock, as explained in the next subsection.

3.4.3.2 Game Clock Recognition (GCR)

Our GCR module uses the method proposed in [91] to extract the Temporal Neighboring Pattern Similarity (TNPB) feature. In 18 out of 20 games videos the clock digit were successfully segmented where 2 were missed due to the low resolution in MPEG I video recording. When the template of each digital number was successfully extracted, the template matching processing to recognize the game time reached 99%. However, the GCR module needs averagely 5 to 6 minutes to reliably segment the digital number before

it can recognize the game time, some even as great as 20 minutes. Hence to reduce the overall delay of event detection and event boundary detection, the GSD result was used to calculate the game time until the template of each game clock number was obtained.

3.4.3.3 Video Event Boundary Detection

In this experiment 11 hours of World-Cup 2002 and Euro-Cup 2004 game videos from the data set were used to train the HMM models as described in subsection 3.3.2. The rest of the data set (26 hours) was used for testing. The performance of the video/text alignment search is evaluated using the Boundary Detection Accuracy (BDA) measure [67]:

$$BDA = \frac{\tau_{db} \cap \tau_{mb}}{max(\tau_{db}, \tau_{mb})} \tag{3.14}$$

where τ_{db} and τ_{mb} are the automatically detected event boundary and the manually labeled event boundary, respectively. A higher BDA value represents better accuracy. The accuracy of boundary detection for each event is listed in Table 3.8.

Table 3.8: Accuracy of event boundary detection

Event	BDA	Event	BDA
card	83.2%	foul	75.7%
goal	80.7%	save	66.2%
offside	67.5%	freekick	58.6%
injury	92.5%	substitution	80.7%

The major reason that leads to the inaccuracy in video and text alignment is the presence of other events within the search time window. Such occurrence would erroneously produce probability peaks not for the intended event. We also notice that for some cases the suggested search window given by text analysis were invalid as the broadcast video data did not record the stated event at all. However, our video/text alignment module would still suggest a segment as the event boundary and hence would be completely wrong. To solve this problem, we have considered thresholding the probability score $P(\mathbf{X}|\Theta)$ (Eq.3.12) to detect such occurrence. However, this strategy may filter out some correctly detected events. Further investigation of a proper trade-off between precision/recall is on-going.

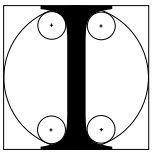
3.5 Conclusion

The chapter presents a generic event detection and event boundary detection method based on synchronization of textual information and A/V features. The approach is generis in the following two aspects: First, the web-casting text is available for many types of sports game and can be easily obtained [98, 100, 101, 127], hence a multi-modality framework using web-casting text in addition to A/V features analysis can be easily realized; Second, the text and A/V feature synchronization problem is a general problem when performing multimodality analysis [93], hence solutions for the problem is required across different sports domains; Third, the selected visual/audio features for event boundary detection are available from most broadcast sports videos and hence are not limited to single sports domain. The identified semantic events can be used to customize soccer video summarization, soccer highlight generation, sports MTV production, etc applications as discussed in Chapter 5.

Although the proposed multimodality analysis method is generic, during feature extraction, the way to extract certain features can be domain-specific. For example, although the broadcast basketball video and soccer video can both be classified into far view, medium view and close-up view, the green pixel count measure used for soccer video won't apply for basketball game. However, when enough training data is available, defining a generic set of visual/audio representations and discovering generic feature extractor are possible. This is one of the future works listed in Chapter 6

Chapter 4

Event Detection from Non-Broadcast Sports Video



N the previous chapter, we proposed a robust event detection approach for broadcast sports video based on textual, audio and visual feature analysis and alignment. Other researchers have also reported many encouraging algorithms and systems on broadcast video data to facilitate sports video management, summarization and retrieval. However, these existing works based on broadcast video mainly rely on information such as the occurrence of replay [35, 48], multi-camera view transition [107, 16, 46, 126] and presence of closed caption text [112, 34, 90]. Hence, these techniques would not be suitable for applications that process raw videos. In this chapter, we propose a novel technique to recognize events from raw unedited sports video with the intention to automatically create broadcast sports video. We use the raw soccer game video for our discussion due to the popularity and high commercial potential of soccer games.

We visited the Singapore MediaCorp broadcast control center in an on-site live broadcasting setup to study how a Singapore Soccer League (S-League) game is produced. Figure 4.1.a illustrates the placements of seven cameras used to record the S-League games. Each camera produces a raw capture from different angle, specifically, camera #1 (Figure 4.1.a) is called the main-camera which provides a panoramic view of the match; Camera #2,...,#7 are sub-cameras that provide additional medium or close-up views from different angles. We call the videos captured by these multiple cameras the raw unedited soccer videos. These raw videos are used to construct broadcast soccer

video using additional visual/audio equipments (Figure 4.1.b, c) to create the necessary editing effects (Figure 4.1.d, e).

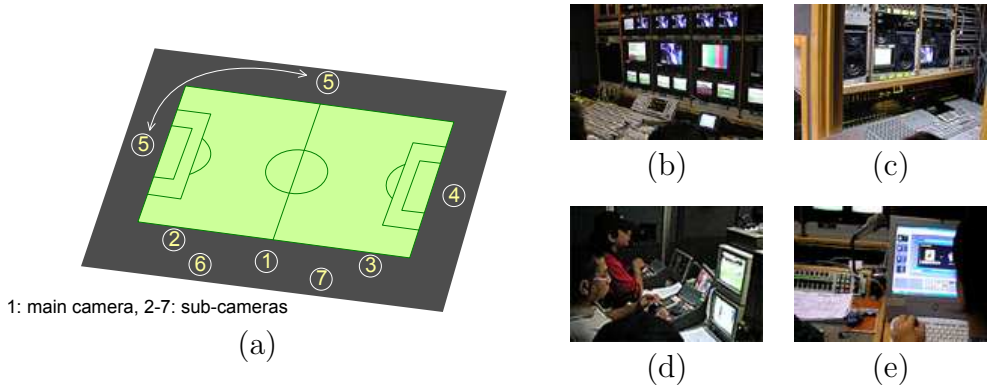


Figure 4.1: Typical setup in Singapore soccer broadcasting

(a) Multiple camera setup; (b) Video equipments; (c) Audio equipments; (d)Replay control equipments; (e) Caption insertion software

4.1 System Overview

Our objective in this chapter is to automatically detect events from the raw videos. As the view of the sub-cameras is too small, they are not suitable for event analysis. Our focus is to detect events from the raw video produced by the single main-camera. Detecting events from the single main-camera soccer video is more challenging as compared with detecting events from broadcast soccer video due to the lack of post-production information to facilitate analysis, *e.g.*, there is no shot transition patterns as analysis is performed using only the main-camera video. There is also no live web-casting text information to assist analysis. Due to these limitations, the ability to detect many types of event from the video is severely limited. Our goal is to recognize only the most important event types in soccer games. To define which events are important, a quantitative study using 5 World-Cup 2002 games is conducted [67]. We deem an event as important if a replay was launched, and we found 143 such events. These 143 event can be classified into three types, specifically attack, foul and miscellaneous (others), and their distribution are listed in Table 4.1.

Table 4.1: Events for replay

	Total	Attack	Foul	Misc
Number	143	70	67	6
Percentage	100%	49%	47%	4%

The **Attack** event is defined as play consisting of scoring or just-missing shot of a goal, the **Foul** event is defined as incidents involving a referee decision (referee whistle), and the **Misc** events consist of other replay-worthy events besides attack and foul event, *e.g.* injury event.

Our focus is to detect those Attack, Foul and Misc events from the raw main-camera soccer video. To achieve satisfactory event detection performance, we use multiple domain features to improve the robustness and accuracy, and also multiple level structure to bridge the low-level features to high-level semantic events. Figure 4.2 illustrates our proposed framework. Specifically, the low-level modules extract the visual, audio and motion vector streams from the input video. These low-level features are then analyzed by the mid-level system to generate keyword sequences to be processed by the high-level system to detect events and event boundaries.

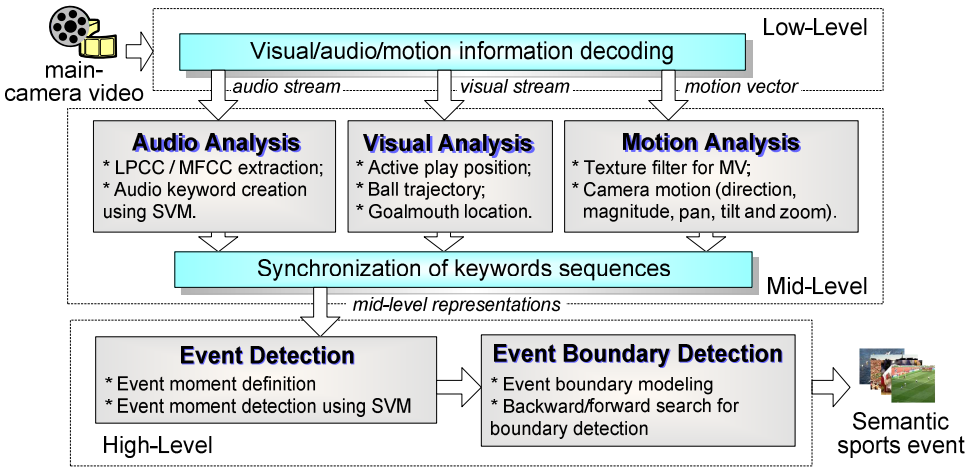


Figure 4.2: Framework of non-broadcast sports video event detection

As the low level implementation is straightforward, it will not be discussed and we directly proceed to the mid-level representations section.

4.2 Mid-level Representation

The mid-level system generates a synchronized feature sequence based on the visual, audio and motion information extracted from the main-camera recording. Details of each representation are listed in Table 4.2 where column 1 lists the feature IDs, column 2 and 3 describe the associated analysis and column 4 lists the values of the features. The following subsections discuss the creation of each keyword.

Table 4.2: Visual/audio analysis description

ID	Description	Analysis	Value
F_1	Active play position	Visual	$\{region\ label\} \in \{1, 2, 3, 4, 5, 6\}$
F_2	Ball trajectory	Visual	$\{x, y\} \in R^2$, coordinate of the ball
F_3	Goalmouth location	Visual	$4 \times R^2$, coordinates of the goalposts vertexes
F_4	Camera motion	Motion	$\{pan, tilt, zoom, direction, magnitude\} \in R^5$
F_5	Audio keyword	Audio	$\{keyword\} \in \{whistle,\ acclaim,\ noise\}$

4.2.1 Active Play Position

F_1 is the active play position keyword reflecting the region of play captured by the main-camera for each video frame. The soccer pitch is divided into 15 regions in which symmetrical regions are labeled the same. This results in 6 keyword labels, $\{1 \ldots 6\}$, as shown in Figure 4.3. Similar works have been reported in [77, 34]. In comparison with [34] which has 12 coarser field regions, our field division is finer with greater precision.

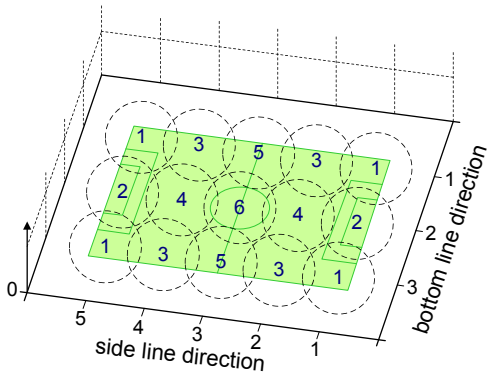


Figure 4.3: Soccer field model

To identify the active play region in the field, three field shape features are extracted, specifically the field-line locations, goalmouth location, and center circle location. The methods to extract these three shapes are discussed below:

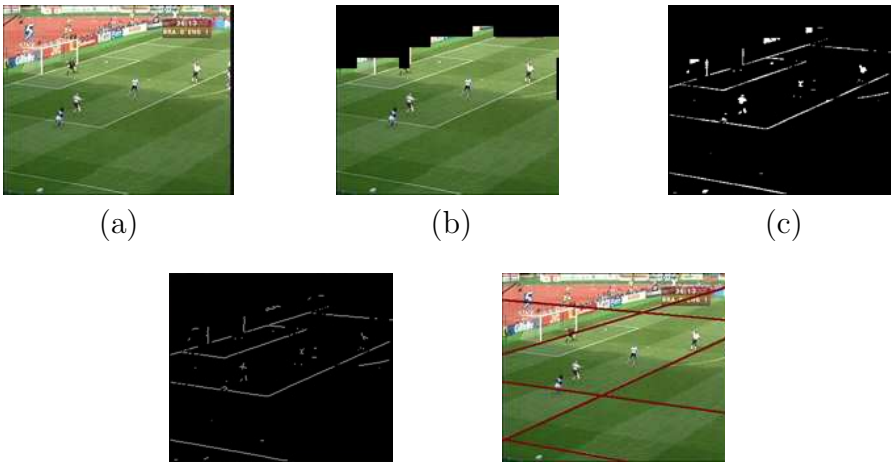
Field-line location detection: Each frame is first divided into blocks of 16x16 pixels in size. Dominant color analysis is applied to ignore blocks with less than half green pixels. A pixel with $(G - R) > T_1$ and $(G - B) > T_1$ is deemed as a green pixel, where R, G, and B are the three color components of the pixel in RGB color space, and the threshold T_1 is empirically set 20. This threshold value is found to be suitable for most soccer playground condition in our video database.

The color image is then converted to gray scale and edge detection is applied using Laplace-of-Gaussian (LOG) [139] method. To reduce the effect of unbalanced luminance, the LOG edge detection threshold T_2 is updated adaptively for each block. An initial small threshold is used, and the threshold is allowed to increase until no more than 50 edge pixels are generated from each block. This is because a line such as field-line will typically only generate 50 edge pixels within a 16x16 block. The edges are then thinned to 1 pixel width and finally the Hough Transform (HT)[140] is used to detect lines. Figure 4.4 illustrates the field-line detection process. The lines detected in each frame are represented in polar coordinates,

$$(\rho_i, \theta_i) \quad i = 1, 2, ..., N \tag{4.1}$$

where ρ_i and θ_i are the i^{th} radial and angular coordinate respectively and N is the total number of lines detected in the frame.

Goalmouth location detection: The detection of the two goalposts is used to identify the presence of the goalmouth. Since the goalposts and crossbar must be white [141], we adopt a color based detection algorithm. The image is first binarized into a black/white image, with white pixels to 1 and other pixels to 0. Vertical line detection and region growing operation are subsequently applied to detect and fix the broken goalpost candidates, respectively. When performing region growing, every black valued pixel can grow into a white pixel if it is connected with no less than 2 white pixels (using 4-connection).



(a) (b) (c)

(d) (e)

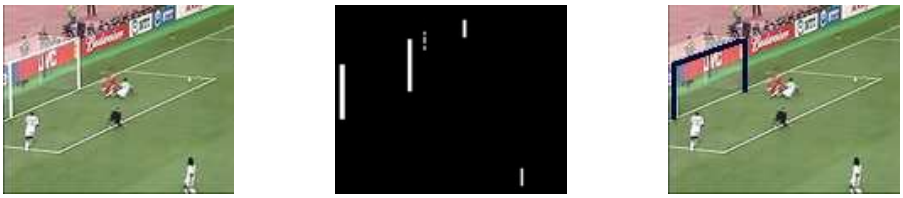
Figure 4.4: Field-line detection
(a) original frame (b) green region (c) edge detection (d) thinning operation (e) field-lines (marked in red)

As the main-camera is usually at a fixed location overlooking the middle of the field, the image of the goalmouth will be slanted (Figure 4.5.a). We apply the following domain rules to eliminate non-goalpost pixels:

- The height of two true goalposts should be nearly the same and within a suitable range.
- The distance between two true goalposts is within a suitable range.
- The two true goalposts should form a parallelogram, not other shape such as square or trapezium.
- There should be some white pixels connecting the upper of the two true goalposts due to the presence of the crossbar.

If there are more than one goalpost candidates in the frame (as in Figure 4.5.b), the candidates that forms the biggest goalmouth are selected as the true goalposts. If a goalmouth is detected, the goalmouth central point (x_g, y_g) is initialized, otherwise $x_g = y_g = -1$.

Center circle location detection: Due to the position of the main-camera, the central circle's image capture appears to be an ellipse. To detect this ellipse, the previous



(a) (b) (c)

Figure 4.5: Goalmouth detection
(a) original frame (b) goalposts candidates (c) goalmouth (marked in blue)

section's line detection results are first used to locate the center vertical line, namely the halfway line [141]. Secondly, the upper and lower border lines of the possible ellipse are located by horizontal line detection (Figure 4.6).

In a horizontal ellipse expression, there are 4 unknown parameters $\{x_0, y_0, a^2, b^2\}$ where (x_0, y_0) is the center of the ellipse, a and b are the half major axis and half minor axis of the ellipse. They can be found by the following routine:

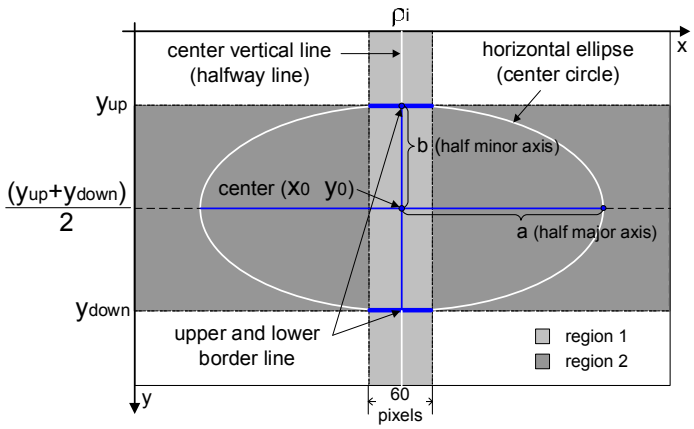


Figure 4.6: Fast center-circle detection

Suppose the y-axis location of the two horizontal border lines under consideration are y_{up}, y_{down}, we have:

$$x_0 = \rho_i \tag{4.2}$$

$$y_0 = \frac{y_{up} + y_{down}}{2} \tag{4.3}$$

$$b^2 = \left(\frac{y_{down} - y_{up}}{2}\right)^2 \tag{4.4}$$

where ρ_i is the center vertical line found in Eq.(4.1). The unknown parameter a^2 can be computed by the following transform to 1-D parameter space:

$$a^2 = \frac{(x - x_0)^2}{1 - (y - y_0)^2/b^2} \tag{4.5}$$

To improve efficiency, we only need to evaluate (x, y) from region 2 (Figure 4.6) to compute a^2.

The above steps are applied to all possible border line pairs and the a^2 found with the largest accumulated value in parameter space is considered to be the ellipse's position. This method is able to locate the ellipse even if it is cropped provided the center vertical line, upper and lower border are present. The detected center circle is represented by its central point (x_e, y_e). If no center circle is detected, then $x_e = y_e = -1$.

With the obtained field shape features, a Competition Network (CN) is proposed to detect the active play region using the three features described above. The CN consists of fifteen dependent classifier nodes, each node representing one area of the field as illustrated in Figure 4.3.a. The fifteen nodes compete amongst each other, and the winning node is identified as the chosen region of play.

The CN operates in the following manner: at time t, every detected field-line (ρ_{it}, θ_{it}), together with the goalmouth (x_{gt}, y_{gt}) and center circle (x_{et}, y_{et}), forms the feature vector $\mathbf{v}_i(t)$ where $i = 1...N$, N is the number of lines detected at each time t. Specifically, $\mathbf{v}_i(t)$ is

$$\mathbf{v}_i(t) = [\rho_{it}, \theta_{it}, x_{gt}, y_{gt}, x_{et}, y_{et}]^T \quad i = 1, ..., N \tag{4.6}$$

The response of the j^{th} node is

$$r_j(t) = \sum_{i=1}^{N} \mathbf{w}_j \mathbf{v}_i(t) \tag{4.7}$$

where

$$\mathbf{w}_j = [w_{j,1}, w_{j,2}, ..., w_{j,6}] \tag{4.8}$$

is the weight vector associated with the j^{th} node, $j = 1...15$ for the 15 regions. The set of wining nodes at time t is

$$\{j^*(t)\} = \arg\max_j \{r_j(t)\}_{j=1}^{j=15} \tag{4.9}$$

The wining node $\{j^*(t)\}$ is sometimes not a single node. There are 3 possible scenarios for $\{j^*(t)\}$, i.e, a single winning entry, a row winning or column winning entry of the 15 regions. This instantaneous winner list is not the final output of the CN as it is not robust. To improve classification performance, an accumulated response is computed as

$$R_j(t) = R_j(t-1) + r_j(t) - \alpha \cdot Dist(j, j^*(t)) - \beta \tag{4.10}$$

where $R_j(t)$ is the accumulated response of node j, α is a scaling positive constant, β is an attenuation factor, and $Dist(j, j^*(t))$ is the Euclidean distance between node j to the nearest instantaneous wining node within the list $\{j^*(t)\}$. The variable $\alpha \cdot Dist(j, j^*(t))$ in Eq(4.10) corresponds to the amount of attenuation introduced to $R_j(t)$ based on the Euclidean distance of node j to the winning node. A large $Dist(j, j^*(t))$ will result in stronger attenuation in Eq 4.10.

To compute the final output of CN at time t, the maximal accumulated response is found at node $j^{\#}(t)$ where

$$j^{\#}(t) = arg\max_j \{R_j(t)\}_{j=1}^{j=15} \tag{4.11}$$

If magnitude $R_{j^{\#}}(t)$ is larger than a predefined threshold, the value of position keyword F_1 at time instant t is set to $j^{\#}(t)$, otherwise it remains unchanged.

4.2.2 Ball Trajectory

F_2 is the ball trajectory feature captured from the main-camera frame. It is an R^2 vector stream recording the $\{x, y\}$ coordinates of the ball with respect to each frame's origin (top-left of the screen). The ball trajectories are obtained using the I^2R [134] media research group's ball detection and tracking algorithm [56]. The algorithm uses a Kalman filter to evaluate whether a candidate trajectory is a ball trajectory by sequentially executing "ball size estimation", "ball candidate detection", "candidate trajectory generation" and "trajectory processing" to obtain a ball position list (*i.e.*, the trajectory) over frames.

4.2.3 Goalmouth Location

F_3 is the goalmouth location feature. It consists of four pairs of $\{x, y\}$ coordinates marking the four edge positions of the goalmouth with respect to frame origin. If no goalmouth is detected, $\{x, y\}$ default to $\{-1, -1\}$. The method to detect the goalmouth is described in the above "Active Play Position" subsection 4.2.1.

4.2.4 Camera Motion

F_4 is the camera motion feature extracted from each frame of the captured video. The F_4 feature is an R^5 vector consisting of the pan, tilt, zoom, dominant motion direction, and average motion magnitude values of a camera's motion. We apply the same method as we introduced in subsection 3.2.4 to extract the camera motion keywords.

4.2.5 Audio Keyword

F_5 is the audio keyword sequence extracted from the audio track of the main-camera capture. In subsection 3.2.5 we discuss the use of Support Vector Machine (SVM) to classify the Mel-Frequency Coefficient Cepstral (MFCC) and LPCC audio features to obtain the audio keywords. The same method is applied in this subsection.

4.2.6 Post-Processing

In the mid-level analysis discussed in the previous sections, each module generates one keyword vector for one video/audio frame to obtain a keyword sequence. As the mid-level keywords possess coarse semantic concepts and since semantic concept normally has notable duration, the sequence values of the mid-level keyword vectors should not change abruptly. For example, a "whistle" should last at least 0.5 seconds, *i.e.* 12 continuous frames for video of 25 fps. Hence if we observe keyword changed for only 1 or 2 frames, these should be regarded as erroneous classifications. To remove such erroneous classifications, we apply a post-processing operation using majority-voting on a window of w_l elements with step-size w_s to smooth the keyword sequence. For our system, the following values are found to be suitable empirically:

- Active play position F_1: $w_l = 25$ and $w_s = 10$

- Goalmouth location F_3: $w_l = 12$ and $w_s = 8$, the sliding window is conducted only on detected goalmouth coordinates.
- Audio keyword F_5: $w_l = 5$ and $w_s = 1$

After post-processing, these mid-level keywords are ready for the next high-level system module to perform the event detection and event boundary detection tasks. The following sections describe the detailed implementation of the two tasks.

4.3 High-level Event Detection

4.3.1 Event Detection

To recognize certain interesting events from sports video, both the spatial correlations between different features and their temporal transition patterns can be utilized. For example, to detect a goal scoring event, there are spatial patterns such as close distance between goal-mouth (the goal-mouth detection feature F_3) and the ball (The ball tracking feature F_2), companying audience acclaim (The audio keyword feature F_5); There also exist temporal patterns such as transition from far-view shot to close-up view shot and slow-motion replay shot (The Semantic Shot Classification (SSC) feature in section 3.2.1). If these patterns can be recognized for the feature sequence, the corresponding events can be identified.

4.3.1.1 Modeling Temporal Patterns for Event Detection

Many reported works applied to model the temporal patterns for sports event detection [103, 104, 106, 107, 142, 143]. In our previous work [136], we also evaluated event detection using Hidden Markov Model (HMM) based classifier to capture the statistics of shot-type and audio keyword transitions from broadcast soccer video. The idea in [136]works in the following manner:

We define a generic HMM model (Prototype) for these events. For simplicity, we limit this model to a left-to-right HMM (later state cannot transit to earlier state). This limitation does not decrease the detection accuracy because events in sports video are from real world, time passes only from the early to the late. Figure 4.7 illustrates the structure of the HMM prototype used in our framework.

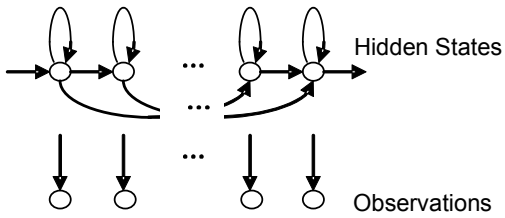


Figure 4.7: Left-to right prototype HMM structure

To collect the training data, we first manually segment events from the full length soccer video. The feature extracted from each event clips are used to train respective HMM models. During recognition, the same set of features are extracted from input video and fed to all the HMMs. The one that achieves the highest likelihood score gives the recognized event type. The features we selected are the SSC (section 3.2.1) and the audio keyword feature F_5.

When applying the prototype for sports event detection, two issues should be addressed: First, the prototype should include enough states so that it can explicitly model every event. If too few states are used, the detection accuracy will significantly drop as observed in our experiment; Second, HMM needs large amount of training data for each event. however in most games, for example in soccer game, while the event of "Shot" or "Goal kick" happens quite often, the event of "Goal scoring" happens rarely. The biased amount of training data will cause inaccuracy. We noticed that if the HMM of "Goal" is not fully trained, this type of event will be recognized as "Shot". One way to solve this problem is to collect balanced number of training events for each HMM. An alternative way is to use the same set of training data repeatedly for events with less samples.

4.3.1.2 Modeling Spatial Patterns for Event Detection

The above HMM based method works well for event detection from broadcast soccer video, because the selected features, especially the SSC feature, are post-edited in the broadcasting process and hence possess strong correlations between their temporal transition patterns and the event. However in this work, as the five mid-level keywords ($\{F_1, F_2, F_3, F_4, F_5\}$, Table 4.2) do not possess strong transition patterns for event detection, we did not apply the same HMM system but instead use a template classifier for event detection. We observe that events can be detected directly by evaluating the

mid-level features without temporal history, i.e., we observe that certain patterns recur for certain events, and these patterns could be used as templates to detect such occurrences. We name such moments with distinguishing feature pattern as "event moment". To illustrate what constitute an event moment, we examine a typical "Attack" event. Figure 4.8 shows a segment of the mid-level representation sequences of an "Attack" event. Figure 4.8.a is a manually tagged partition of no-event/event/no-event sequence. Note that the precise start/end of the event boundary marked by a human operator is highly subjective and may differ operator to operator. We note however that within the event boundary, there exists a shorter segment with feature patterns recurring for all other "Attack" events. Specifically, we observed that all attack events have ball close to goalmouth location (Figure 4.8.b), the detected active play position (F_1, Table 4.2) is at region 2 (penalty area, Figure 4.8.c) and the audio keyword equals to "acclaim" (Figure 4.8.d). We define this shorter sequence with features corresponding to the above criteria the "Attack" event moment. We can similarly define criteria for the "Foul" and "Misc" event moments.

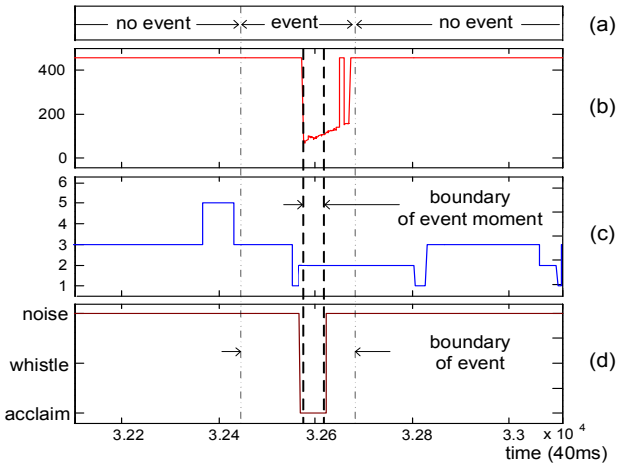


Figure 4.8: Event moment of an "attack" event
(a) event segment; (b) ball-goalmouth distance (pixel); (c) position (labels in Figure 4.3); (d) audio (label)

To detect event moments, the SVM classifiers with Gaussian kernels are used. The SVM classifier is chosen due to its speed and accuracy advantage in modeling feature patterns without considering temporal information. The input to each SVM classifier uses

a different set of mid-level keywords to achieve good event moment detection performance. For example, using F_1, F_2, F_3 and F_5 (Table 4.2) only for "Attack" detection gives better result than using all five mid-level keywords. By exhaustively evaluating all combinations of mid-level features, the set of optimum inputs for each classifier is found:

- **Attack classifier**: position keyword (F_1), ball trajectory (F_2), goalmouth location (F_3) and audio keyword (F_5);
- **Foul classifier**: position keyword (F_1), motion activity keyword (F_4) and audio keyword (F_5);
- **Misc classifier**: position keyword (F_1) and motion activity keyword (F_4).

The SVM is trained using manually tagged data for each event moment, *i.e.* the input features of the required event moment segments are marked as positive examples, and all other frames are marked as no-events (negative examples). 3 hours of World-Cup 2002 videos consisting of 102 "Attack", 85 "Foul" and 7 "Misc" events are used to train the three SVM classifiers respectively. In the detection process, the selected sets of keyword sequences are fed to the SVM classifiers to detect respective event moments. As different events would not occur simultaneously, the identified event moments from different detector will not overlap, and hence we can simply merge the results from different SVM detector into one event list chronologically as illustrated in Figure 4.9. Theoretically, the identified event moments from different detector may overlap due to erroneous classification, however this has never been observed in our experiments. During our experiments, we note that intermittent frames within an event moments may be mis-classified as no-event. Such erroneous classification would generate false break of the same event. To reduce this type of sporadic error, we again apply a majority voting step to post-process the event list. The final output of the event moment detector is a list of event moments where the starting time (denoted as T_{ms}) and ending time (T_{me}) of each event moment are recorded (Figure 4.9).

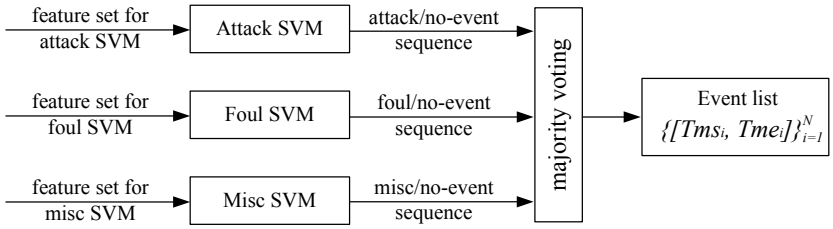


Figure 4.9: Event detection scheme

4.3.2 Event Boundary Detection

To define suitable boundaries for detected events, another quantitative study is conducted. 144 event boundaries from three hours of World-Cup 2002 videos are manually tagged. We found that the audiences' interest from the start to the end of an event can be classified as "zero interest"-"increasing interest"-"maximal interest"-"decreasing interest"-"zero interest again" pattern. Let's take a goal-scoring event as an example, the audiences' interest increases eventually after the ball is passed into the penalty area, reaches the highest value when the goal is scored, and then decreases to zero during the resulted game break. To model how viewers perceive events, we assume that the event moment represents the peak of viewers' interest in the event and those video contents further away from the start/end-points of the event moment are of less interest to the audience. For simplicity, we use a linear relationship with respect to time to model the change in viewer's interest to an event is used (Figure 4.10). We have also observed that the audiences' interest rises/falls faster if the start/end boundary includes an active play position (F_1, Table 4.2) change. For example, if a long-pass to penalty area (*i.e.*, a position change in F_1) results in an immediate goal-scoring, the audiences' interest would have increased very quickly to the event. Hence the manually labeled event start boundary will be shorter as compared to a goal-scoring event with no play position changed where the audiences' interest rise slowly. We therefore created two models to depict the viewers' interest as illustrated in Figure 4.10. The solid line in Figure 4.10 represents the audiences' interest to an event that remains at the same play position throughout the event duration (denoted as *model 1*) while the dashed line in Figure 4.10 illustrates the interest model for an event that does not remain at the same play position (*model 2*). The difference between the two models is that the slopes for *model 2* is steeper

therefore the detected event boundary by *model 2* is shorter than *model 1*. To build *model 1*, the event moment is first detected as described in previous subsection. Let the event moment's start and end points be denoted as T_{ms} and T_{me}, and the corresponding audiences' interest (y value) between them be equal to 1 as illustrated in Figure 4.10. To linearly model the audiences' interest, two straight lines are drawn to intercept the x-axis to estimate the event boundary start $(T_{ms} - D_s)$ and end $(T_{me} - D_e)$ respectively. The average duration D_s between the event start boundary and event moment start is found by

$$D_s = \frac{1}{N}\sum_{i=1}^{N}(T_{ms_i} - T_{s_i}) \tag{4.12}$$

where N is the number of *model 1*'s training events, T_{ms_i} is the start of event moment boundary, T_{s_i} is the start of event boundary for training event i respectively. The average duration D_e between the event end boundary and event moment end is estimated similarly. The *model 2*'s D_s and D_e are evaluated similarly using training events with a change in play position alternately. We denote *model 2*'s D_s and D_e as D'_s and D'_e respectively. For our implementation, the values of D_s, D_e and D'_s, D'_e are found to be $\{2, 1\}$ and $\{1.2, 0.2\}$ seconds. These values are estimated using our training database consisting of 122 events that have no play position change and 72 events that have play position change. We note that the experimentally found D'_s and D'_e have a smaller value than D_s and D_e. This agrees with our assumption that a change in play position will result in a shorter event boundary.

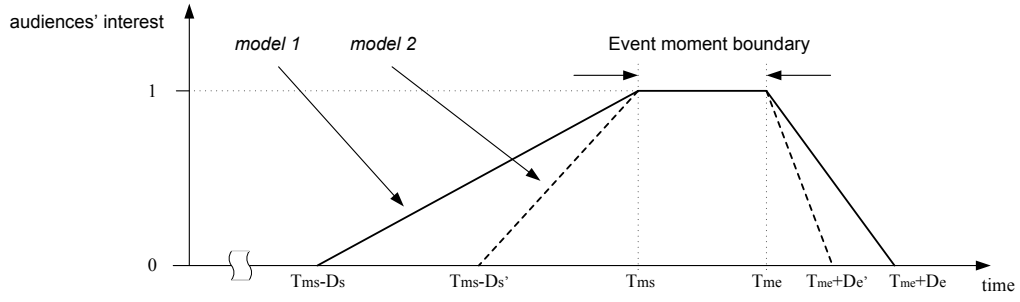


Figure 4.10: Audiences' interest model

In the detection process for event start boundary, a backward search is performed from the event moment start T_{ms} following the slope of *model 1* until $T_{ms} - D_s$ or if

a change in active play position for a frame is encountered. If the active play position is changed, the backward search jumps to *model 2* and select $T_{ms} - D'_s$ as event start boundary. The process to detect the event end boundary is the same as above but using forward search from event moment end T_{me} instead.

4.4 Experimental Results

In this section experiments are conducted to evaluate the accuracy of individual modules in our framework as well as the performance of the whole system. The following subsections summarize the results.

4.4.1 Accuracy of Mid-level Representation

The set of parameters used to extract the five mid-level representations ($F_1, ..., F_5$ in Table 4.2) were trained using 3 hours of World-Cup 2002 game videos. These parameters include the thresholds for field line/goalmouth/center circle detection, the weight of each CN node, the parameters of the Kalman filter for trajectory tracking and the SVM models for audio keyword creation. The parameters were kept unchanged throughout the testing process. In the evaluation process, 10 minutes (15000 frames) of labeled main-camera soccer video (camera #1 in Figure 4.1.e) from our S-League video database were used. The experiments for each representation are detailed in the following subsections.

4.4.1.1 Active Play Position

In this experiment, our active play position detection module recognized the region of play in each frame as illustrated in Figure 4.3.b. The result was then compared with our manually identified region labels and the precision is listed in Table 4.3. It is noted that the detection accuracy for region 4 is low compared with the other labels. This can be easily explained: Field region 4 (Figure 4.3.b) has fewer cues than other regions, *e.g.*, it does not have field-lines or goalmouth or central circle. The lack of distinct information thus results in poorer accuracy. This agrees with our previous work in [67].

Table 4.3: Accuracy of active play position detection

Position	Precision	Position	Precision
1	72.7%	2	98.3%
3	76.6%	4	51.1%
5	83.0%	6	85.2%

The position is the 6 labels given in Figure 4.3.b

4.4.1.2 Ball Trajectory

In this test, the ball trajectory tracking module uses the reported algorithm in [56] to extract the $\{x, y\}$ location of the ball in each frame. In contrast to [56] where a very clean data set was used, when noisy soccer videos were used the ball trajectory tracking performance drops significantly. Hence we had to manually label the ball position for these videos to allow our system to perform the next level processing. More work is being carried out to improve the accuracy of the ball tracking module.

4.4.1.3 Goalmouth Location

Our goalmouth detection module correctly detected 85% of the goalmouth locations for the test video data. One reason for the missed detection is the occurrence of fast camera motion which blurs the appearance of the goalposts in the image. Another reason is the change in luminance and field condition which renders the recorded color of the goalposts to be non-white. Both these two reasons caused the goalmouth detection error.

4.4.1.4 Camera Motion

Since the motion features are directly extracted from the MPEG motion vector field, no more experiment was carried on it.

4.4.1.5 Audio Keyword

Our system defined three audio classes to be detected, specifically “acclaim”, “whistle” and “noise”. The accuracy for each class was: “acclaim” 91.7%, “whistle” 85.9% and “noise” 99.3%, respectively on a 10 minutes test data set.

4.4.2 Performance of High-level Application

The first set of experiments evaluates the event detection performance from a 15.7 hours soccer video database. These videos consist of World-Cup 2002 video (5.5 hours), Euro-Cup 2004 video (6.5 hours), English Premiere League (EPL) video (2.2 hours) and S-League video (1.5 hours). For these experiments, our system analyzes the raw main-camera video for event detection. To use the World-Cup 2002, Euro-Cup 2004 and EPL video data that are actually broadcast video, we applied pre-processing to remove the non-main-camera view segments from these video. The reason why the broadcast videos are included is due to the limited availability of raw unedited soccer video. In the experiments, event moment detection is evaluated using "precision/recall" measure, and event boundary detection is evaluated using Boundary Detection Accuracy (BDA) measure (Eq.3.14).

Table 4.4 lists the obtained results from the respective video data set. It is observed that our system generates satisfactory ($> 75\%$ accuracy) event detection results from the main-camera capture from most of the video set. Our analysis on the false/missed detection showed that these errors occur due to poor performance in the mid-level feature representation. One example is in the World-Cup 2002 videos for the false detection of "Foul" event. To detect "Foul" event, the audio keyword "whistle" plays an important role. In the World-Cup database, we however encountered many "whistle" keyword detected due to audience whistling rather than the referee which leads to many false detections and hence lower "Precision" score (72.8%). Another example illustrates what causes missed event detection in the Euro-Cup 2004 videos. In our Euro-Cup database, we found that due to the rapid camera movement and thin goal-post appearance, our goalmouth detection module performed poorly which led to missed detection of "Attack" events resulting in a lower recall score (47.1%).

We note that the event BDA for "Misc" event is lower than the other two events. In our database, the "Misc" event consists mainly of occurrences resulting in game stoppages, *e.g.* player injury. In our current implementation, we cannot detect the correct event boundary for this event easily as our event moment algorithm activates only when the camera begins focusing on an injured player. We observe that in some cases however that play was allowed to continue after the occurrence of the event and that the

camera-man continued to track the ball until play is stopped. It was only then that the camera would focus on the fallen player and that the "Misc" event moment was detected. However, by our current implementation of searching from the detected moment to find the event boundary, we will fail to find the actual occurrence of when the player was tackled if the play continues too long after the actual occurrence.

Table 4.4: Accuracy of event moment detection and event boundary detection

Event	Detected	Precision	Recall	BDA
Attack	138	78.3%	94.7%	69.4%
Foul	162	72.8%	83.7%	80.9%
Misc	21	66.7%	77.8%	65.0%

(a) World-Cup 2002 data set

Event	Detected	Precision	Recall	BDA
Attack	117	81.2%	47.5%	79.9%
Foul	122	78.7%	75.6%	66.8%
Misc	34	61.8%	84.0%	58.3%

(b) Euro-Cup 2004 data set

Event	Detected	Precision	Recall	BDA
Attack	27	88.9%	72.7%	67.7%
Foul	33	78.8%	81.3%	69.7%
Misc	6	83.3%	83.3%	26.0%

(c) EPL data set

Event	Detected	Precision	Recall	BDA
Attack	34	82.4%	93.3%	66.7%
Foul	48	91.6%	88.0%	81.4%
Misc	2	50.0%	50.0%	60.0%

(d) S-League data set

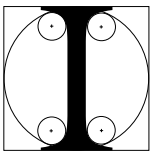
4.5 Conclusion

This chapter presents a novel system to recognize soccer events from raw unedited soccer videos. This technique adds to existing sports event detection literatures and provides

possibilities for broadcast sports video composition/creation applications and will be introduced in chapter 5.

Chapter 5

Automatic Sports Video Composition and Editing



n chapter 3 and 4 we described our semantic event detection techniques for both broadcast sports video and non-broadcast sports video analysis. With the ability to recognize certain desired events from sports video, we are able to re-edit existing sports video materials to generate new video documents. In this chapter three such sports video composition and editing applications are introduced. Specifically section 5.1 discusses a live highlight generation system which automatically detects a large set of events from a live broadcast soccer game and extracts a short video segment for each event to produce personalized highlights. Section 5.2 presents a system that can automatically compose broadcast soccer video from multiple raw camera inputs by analyzing unedited sports video and mimicking human director's practice to control replay insertion and view switching operations. Section 5.3 introduces a personalized Music Sports Video (MSV) generation system which automatically selects and aligns videos scenes and music phrases to compose Music Video (MV)s. Detailed discussion of the three applications are presented in the following sections.

5.1 Live Sports Highlight Generation

With the proliferation of sports broadcasting, the ability to skip un-interesting portions from lengthy sports video programs and view only the events/highlights is highly valued. Currently, sports events and highlights are manually detected and generated by studio

professionals, and access to these highlight segments is very limited for several reasons: First, the highlight are presented only at game breaks, *e.g.* the half-time break in a soccer game; Second, events are exclusively selected by the director and may not meet all the audiences' appetites; And third, the process to manually identify, cut and concatenate different events into a game highlight is an extremely labor-intensive operation and does not allow for personalization easily. Hence the availability of automatic tools to detect events from live sports game and to generate personalized highlights will be sought-after. It is foreseeable that such tool will not only improve the production efficiency of the broadcast professionals, it will also benefit both service providers and consumer client-based applications to offer better game viewership for sports fans [144].

In this section, we present a system to generate personalized highlights from live broadcast sports videos. The soccer game is selected as our development platform because soccer is globally the most popular sport and its broadcast videos are widely produced and watched. The possibility to extend the system to other sports domains is discussed later.

To generate live and personalized highlight from broadcast soccer video, the key research issue is to recognize various soccer events and to locate video event boundaries in real-time. In chapter 3, we proposed a method to detect events from broadcast sports video using web-casting text analysis, video feature extraction and text/video alignment. In this section, we adopt similar technique with modifications for real-time processing. The two major modifications are: We reduce the set of video features used for video event boundary detection and introduce a fast video event boundary modeling algorithm.

Figure 5.1 illustrates our proposed system: First, the sports videos are captured and sent to the video analysis module for Game Start Detection (GSD) and Game Clock Recognition (GCR). At the same time, the text analysis module will detect events from live web-casting text from British Broadcasting Cooperation (BBC) [100] or Entertainment and Sports Programming Network (ESPN) [98] websites. If the occurrence of an event is identified from the text, the time-stamp of the event will be used by the video analysis module to extract features from a tentative video segment. The generated video features, together with the time-stamp of the event, are then sent to the video/text alignment module for video event boundary detection. The event segments found are

concatenated into a game highlight, or directly sent to the users via alternative medium besides TV broadcasting.

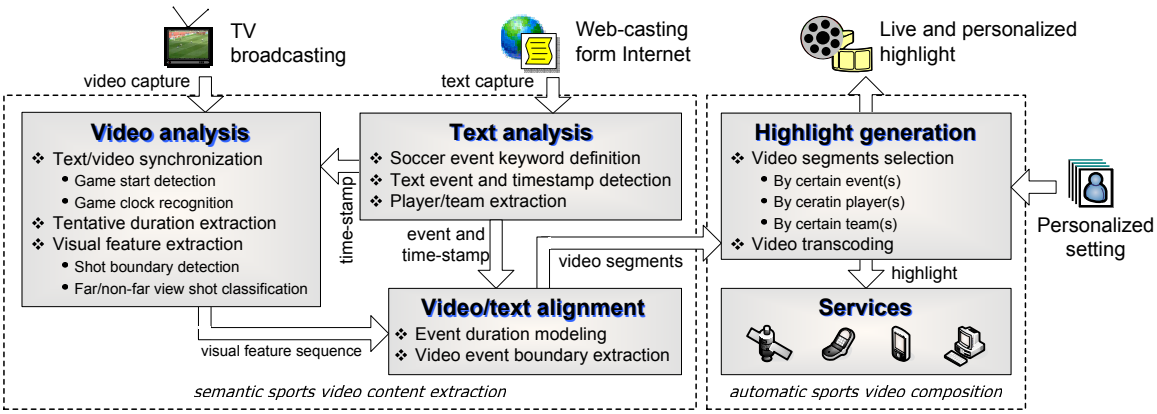


Figure 5.1: A live and personalized highlight generation system

The following subsections discuss our system implementation. We elaborate on the feature extraction and event boundary modeling modules as they are modified from the techniques proposed in chapter 3.

5.1.1 Text Analysis

To develop our live system, the web-casting texts collected from BBC [100] and ESPN Soccernet [98] are used. We selected these web-casting texts as they belong to the well-structured web-casting text category (section 3.1) and thus can be easily mined for text event detection. In addition, the time-stamp of each event entry is very accurate; this allows our system to perform the search in a relatively small segment neighboring the time-stamp to find the event boundary. The overall computation complexity is therefore reduced significantly and thus enables real-time processing.

In this implementation, we directly apply our well-structured web-casting text analysis method introduced in section 3.1 to detect event and identify the name of players/teams involved in the event. The time-stamp of each event is then used to trigger the feature extraction and event boundary detection processes as explained in the next subsections.

5.1.2 Video Analysis

The video analysis module performs a series of operations including GSD/GCR for text/video synchronization and video feature extraction for event boundary detection. The GSD and GCR implementations are the same as that described in subsection 3.3.1. The feature extraction strategy is modified to speed up processing. The new strategy is as follows: When an event is detected by text analysis, the video frame corresponding to the time-stamp of the event is used as a reference time (denoted as f_r) to select a tentative video segment ranging from frame $[f_r - 60, f_r + 120]$ seconds for video event boundary detection, and feature extraction is performed only over this segment. To further reduce computation, only a subset features of those listed in Tablet 3.3 are extracted, specifically the Shot Boundary Detection (SBD) and Semantic Shot Classification (SSC) features. The method to extract SBD and SSC are the same as that proposed in section 3.2.

5.1.3 Video Event Boundary Detection

The video/text alignment module tries to locate a suitable video segment boundary for each detected event from text analysis. In section 3.3.2, a Hidden Markov Model (HMM) based classifier was introduced to model the visual/audio features (Table 3.3) for event boundary detection. This algorithm is robust for alignment between video and web-casting text with inaccurate time-stamping, but is computational expensive. Since the web-casting text selected by our system has accurate time-stamp for each event, we consider simpler algorithms to model the feature transition of an event's duration for boundary detection. To achieve this, we observe the following patterns from the broadcast soccer video and the web-casting text of BBC [100] or ESPN [98].

Observation 1: In broadcast soccer video, the event is usually captured within one live far view shot, or sandwiched between two live far view shots. For the latter, there are successive live non-far view shots and/or replayed shots between the two live far view shots.

Observation 2: The time-stamp in the web-casting text from BBC [100] or ESPN [98] usually logs the most representative moment of each event, and such representative moment falls into the event duration described in observation 1. Figure 5.2.a. shows an example where the event is sandwiched between two live far view shots.

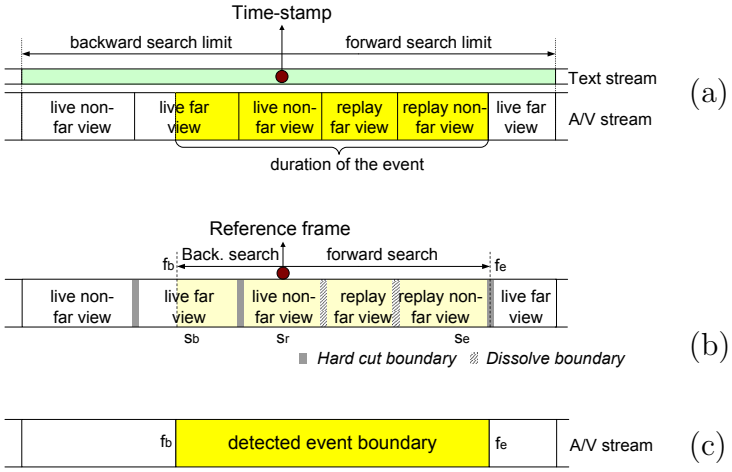


Figure 5.2: Visual/audio stream and text stream
(a) The relationship between the time-stamp and the event boundary; (b) The search range is decomposed into different shot types and boundary types for event boundary detection (c) The detected event boundary

With these two observations, a simply way to find the event boundary is to use a backward/forward search starting from the reference time (f_r, subsection 5.1.2) to locate the nearest live far view shot(s) as the event boundary. To apply this method, our system requires the shot boundary location and the live/non-live, far/non-far view state information of each shot. However, our extracted features only possess the shot boundary location/type information from SBD and far/non-far view type information from SSC but not the live/non-live state of each shot. To solve this problem, we use the following additional rules to link our features to event boundary detection.

Rule 1: A far view shot with hard cut at either start or end boundary is considered to be a live far view shot.

Rule 2: Most events last between 20 to 60 seconds.

With these rules, a Finite State Machine (FSM) is used to locate a smaller segment boundary $[f_b, f_e]$ inside the tentative segment from $[f_r - 60, f_r + 120]$ seconds (Subsection 5.1.2). The FSM operates in the following manner: Assume that our SBD module detects N shots from $[f_r - 60, f_r + 120]$ seconds. First the shot that the reference time f_r belongs to is identified and denoted as s_r (Figure 5.2.b). Starting from s_r, the FSM

performs a backward search along $\{s_i\}_{i=r-1,\ldots,1}$ and changes states at each shot boundary with given conditions listed in Figure 5.3 to find a suitable event start shot s_b. Then, as the event start boundary f_b may not always be the same as the start boundary of shot s_b (Figure 5.2.b), a suitable time in shot s_b need to be selected as the exact event start boundary. This is achieved in the "Start boundary refine" state where the greater of either the starting time of s_b or the ending time of s_b minuses 10 seconds is selected as the event start boundary f_b. After f_b is found, the FSM performs a forward search along $\{s_i\}_{i=b+1,\ldots,N}$ to find the event end shot s_e. The end boundary of shot s_e is selected as the event end boundary f_e. The forward search is similar to the backward search except that it is working in a forward direction. The detected video event boundary is illustrated in Figure 5.2.c.

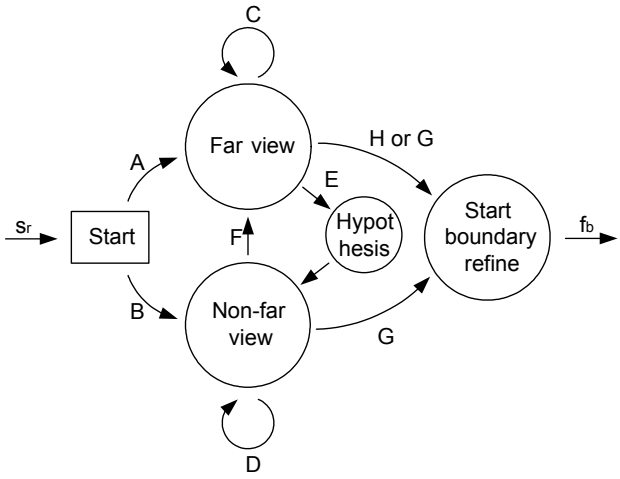


Figure 5.3: FSM for event boundary modeling
Transition condition A: s_r is a far view shot; B: s_r is a non-far view shot; C: Transit from current shot s_i to another far view shot s_{i-1} via a dissolve shot boundary; D: Transit from current shot s_i to another non-far view shot s_{i-1}; E: Transit from current shot s_i to a non-far view shot s_{i-1} via a dissolve shot boundary. s_i value is recorded as $s_h = s_i$ (The hypothesis state); F: Transit from current shot s_i to a far view shot s_{i-1}; G: Rule 2 failed ($s_i - f_r > 60$ seconds). $s_b = s_h$ if s_h is set, or $s_b = s_i$ otherwise; H: transit from current shot s_i to a shot s_{i-1} via a hard-cut shot boundary. $s_b = s_i$

5.1.4 Experimental Results

We conducted our experiments on both live games and recorded games. The following subsections only list the video/text alignment accuracy as the rest of the modules'

performance have been discussed in chapter 3.

5.1.4.1 Performance of Video/text Alignment on Recorded Data

To assess the suitableness of the automatically located video event boundary, the Boundary Detection Accuracy (BDA) measure (Eq.3.14) is adopted. Table 5.1 lists the results using 4 recorded English Premiere League (EPL) game videos.

Table 5.1: Accuracy of event boundary detection on recorded data

Event	BDA	Event	BDA
goal	90.0%	red card	77.5%
shot	86.9%	yellow card	77.5%
save	97.5%	foul	77.7%
freekick	43.3%	offside	80.0%
corner	40.0%	substitution	45.0%

We observed that the BDA of corner and freekick events is lower than the other events. This is because BBC [100] usually puts other events (*e.g.* foul) in the same entry with the corner and free kick events in its web-casting text, and hence the extracted time-stamp is not accurate and thus affectes the video/text alignment performance.

5.1.4.2 Live System Setup and Real-Time Performance

We built a prototype system on a Dell Optiplex GX620 PC (3.4G dual-core cpu, 1G memory) with Hauppauge PCI-150 TV capture card to conduct a series of live trial. Although the average processing delay for video feature extraction and event boundary detection is approximately 10 seconds, the overall processing delay of the whole system depends on the availability of live web-casting text from the BBC [100] or ESPN [98] websites, which is typically 2-3 minutes after the occurrence of the event.

5.1.4.3 Performance of Video/text Alignment on Live Data

Our system went "live" over the April 13th-15th 2006 EPL broadcast. Some integration oversight restricted the system to only complete its run for the 1st half of 4 games. We improved our system and a second live trial was conducted from June 10th to July 10th for all the 64 World-Cup 2006 games. All processing modules were able to execute

seamlessly for the whole match of 61 games; 3 games were missed due to erroneous system configuration. The event boundary detection performance for live EPL games and World-Cup games is listed in Table 5.2 and Table 5.3 respectively.

Table 5.2: Accuracy of event boundary detection on live EPL

Event	BDA	Event	BDA
goal	75.0%	red card	*NA*
shot	82.5%	yellow card	83.0%
save	90.0%	foul	77.7%
freekick	40.0%	offside	85.3%
corner	66.7%	substitution	*NA*

NA: The event did not occur

Table 5.3: Accuracy of event boundary detection on live World-Cup 2006 data

Event	BDA	Event	BDA
goal	76.7%	red card	82.0%
shot	76.1%	yellow card	84.0%
save	60.0%	foul	77.7%
freekick	43.3%	offside	70.5%
corner	75.0%	substitution	78.1%

In the "Live Highlight Generation" folder on the DVD attached to this thesis, readers can view the generated video segments for the first match in World-Cup 2006 (Germany *vs* Costa Rica).

5.1.5 Conclusion

Event detection from live sports games is a challenging task. In this section, we presented a novel application framework for live sports event detection by web-casting text analysis, video features extraction and the alignment of the two. Within this framework, we developed a live event detection system for soccer game and conducted off-line and live trials on various soccer games. The experimental results are promising and validate the proposed techniques.

We believe that the incorporation of web-casting text analysis into sports video analysis will open up a new possibility for personalized sports video highlight generation.

Web-casting texts for various sports games are accessible from many websites; they are generated by professionals or amateurs using various styles (well-structured or free-style) and different languages. Our future work will focus on exploiting more web-casting text sources, investigating more advanced text mining approach to deal with web-casting text with different styles and languages, and conducting live trials on more sports domains.

5.2 Automatic Broadcast Soccer Video Composition

In this second application, we will examine the possibility to compose broadcast sports video from raw camera captures automatically. Our motivation is to address the time-critical and labor-intensive task of commercial broadcasting operation. To the best of our knowledge, there is currently no research or commercial system to automatically compose broadcast sports video from raw camera inputs. In fact, the automatic broadcast sports video composition problem has mostly been overlooked in related literature due to two major reasons: the unavailability of raw multiple camera video feeds, and the difficulty to analyze raw unedited sports video as compared to broadcast sports video where rich post-production information is available to assist event detection.

Producing professional soccer game broadcasting involves many personnel; *e.g.* multiple camera-men to operate cameras installed around the soccer pitch, staff to operate the video/audio/ replay/content-augmentation equipments, and a director to coordinate and control camera feed to television viewers. Although the broadcast soccer video composition process is complex, the generated broadcast soccer video has a well-defined structure [9, 17]. This makes it possible for the video to be generated automatically. The existence of this structure is mainly due to the fact that only a fixed number of cameras (Figure 4.1.e) is available for selection and a limited types of shot is possible, *e.g.* far view, medium view or close-up view. In addition, the look and feel of a typical broadcast game is similar, *e.g.*, sub-camera segments are intermittently launched to break the monotony of a panoramic view shot, replays which consist of far/medium/close-up views are displayed to recall game actions, etc. Hence we believe that if the structure of broadcast soccer video can be emulated, we can automatically compose broadcast soccer video from multiple raw camera captures to reduce the workload of the director, the replay control personnel and the content augmentation personnel.

5.2.1 Broadcast Soccer Video Composition Rules

From discussion with professional directors and also through observation of our soccer video database, we list the following common practices used to compose broadcast soccer video:

Rule 1: During normal game play, the main-camera (camera 1 in Figure 4.1.a) capture is launched 40%-50% of the entire broadcasting duration. The sub-camera (camera 2-7) captures are intermittently inserted to break the monotony of the far view and also to display detailed game actions. This camera switching pattern therefore results in an alternate far and medium/close-up view shots in broadcasting composition (Figure 5.4.a, 4^{th} row).

Rule 2: During major soccer event such as foul/goal/save/just-missing shoot, the director will launch sub-camera captures to display player/referee/coach/audience details and also present replays to highlight game actions (Figure 5.4.b, 4^{th} row).

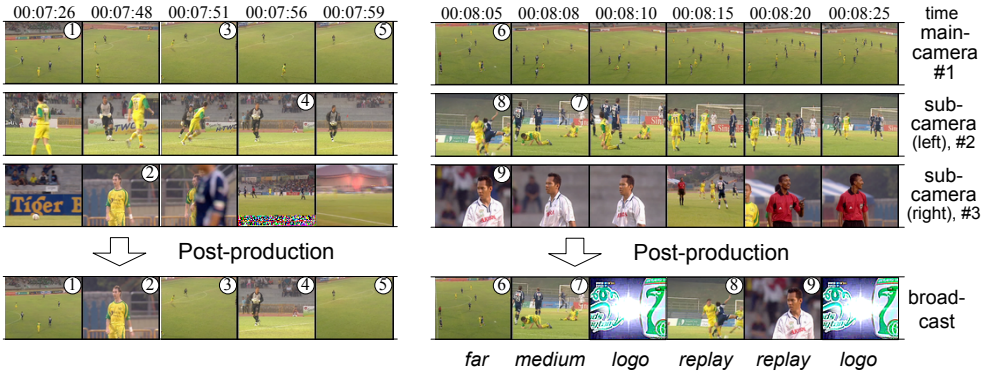
Rule 3: A replay segment usually consists of one or several main/sub-camera shots shown in a slow-motion manner (Figure 5.4.b, 4^{th} row).

Rule 4: Video effects are sometimes inserted to enhance content. The effects include flying-logo (Figure 5.4.b, 4^{th} row), score-bar and caption to provide game/player/team statistics.

To successfully implement the automatic broadcast soccer video composition task, three major research issues must be examined: *Automatic replay generation*, *Automatic view selection and switching*, and *Statistics insertion*. In this application, we mainly focus on the scene composition portion of *Automatic replay generation* and *Automatic view selection and switching*. The *statistics insertion* problem is not dealt with as player/team statistic is not available for our research.

5.2.2 System Overview

Figure 5.5 illustrates our proposed framework. Specifically, the low-level modules extract the visual, audio and motion vector streams from the input video. These raw feature



(a) no-event (b) event

Figure 5.4: Broadcast video composition examples
The first three rows show the raw captures of the main camera, the left and the right sub-camera, respectively. The , 4^{th} row shows the composed broadcast video.

streams are analyzed by the mid-level system to generate keyword sequences. High-level system uses these mid-level representations to detect events and event boundaries. The event detection results, together with the mid-level representations, are processed in the application-level system to perform automatic replay generation and view selection and switching to generate a broadcast soccer video composition.

The techniques proposed in chapter 4 are applied to extract the mid-level features and detect the high-level events. The following subsections present our application-level replay scene generation and automatic view selection and switching implementation.

5.2.3 Automatic Replay Generation

The replays are launched after events, hence recognizing important events from raw unedited soccer video is mandatory to generate replays in broadcast soccer video. The techniques described in chapter 4 are used to detect the "Attack", "Foul" and "Misc" events from the raw main-camera soccer video as these events are usually selected for replay. When an event is detected, a suitable time slot to repeat the event as a replay is required. Our focus in this subsection is to find a suitable time slot such that the loss of interesting live game actions is minimized.

Since the decision of when to insert a replay is subjective, we perform a quantitative study based on the database described in section 4.1 to find a set of criteria for replay decision. We found that there are two classes of replays: Most replays are instant replays,

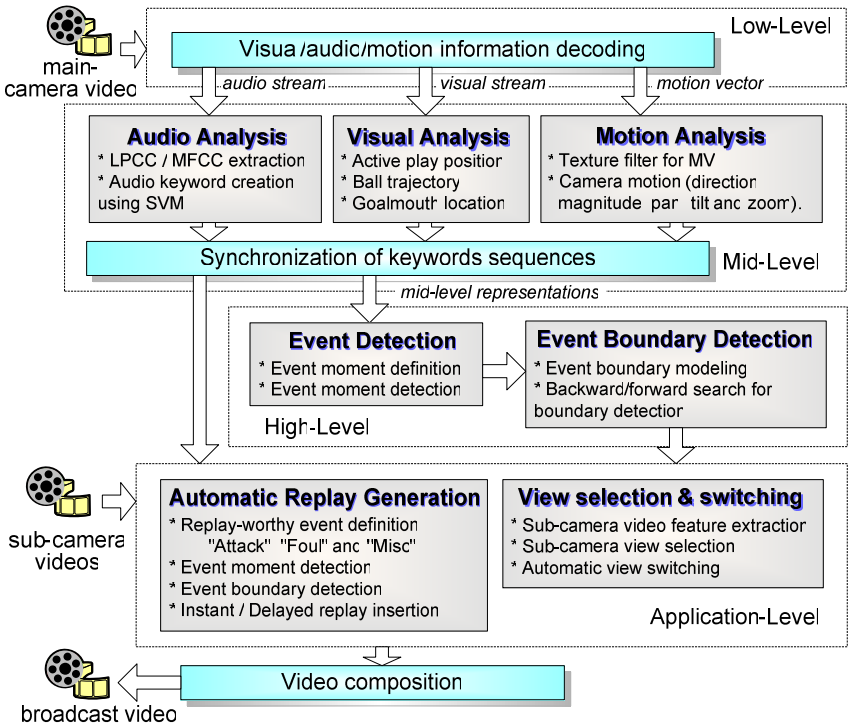


Figure 5.5: Framework of automatic broadcast soccer video composition

i.e., replays which are inserted almost immediately following the event (Table.5.4, Instant replay). The other class are replays which are not inserted immediately; we denote such replays as "delayed replay" and classify the delayed replay into three categories according to the reasons why a delayed replay should be launched:

Category 1: The time slot immediately after the event is occupied by other interesting game actions or event, hence the director has to delay the replay (Table.5.4, Cat. 1).

Category 2: The event is very important and worth being replayed many times (Table.5.4, Cat. 2).

Category 3: The occurrence of an event is missed by the main-camera, hence replays using sub-cameras' captures are launched to recall the event (Table.5.4, Cat. 3).

The input to the replay generation system is the event detection result which has segmented the game into sequentially "event"/"no-event" structure, as illustrated in Fig. 5.6 row 1. If an event segment is identified, the system examines whether an instant

Table 5.4: Possible replay insertion place

Total	Instant replay	Delayed replay		
		Cat. 1	Cat. 2	Cat. 3
143	133 (93%)	5 (3.5%)	3 (2.1%)	2 (1.4%)

replay can be inserted at the following no-event segment, and react accordingly. This is shown in Fig. 5.6 row 2 and 3 where instant replays are inserted for both event 1 and event 2. In addition, the system will examine whether the same event meets the delayed replay condition. If so, the system buffers the event and inserts the replay in a suitable subsequent time slot. This is shown in Fig. 5.6 row 2 and 3 where a delayed replay is inserted at a later time slot for event 1. Fig. 5.6 row 4 shows the generated video after replay insertion.

The following subsections introduced our searching algorithms to find suitable time slots for replay insertion [67].

5.2.3.1 Instant replay generation

The replay starting time T_{rs} and ending time T_{re} are computed as:

$$T_{rs} = T_{ee} + D_3 \tag{5.1}$$

$$T_{re} = T_{rs} + (T_{ee} - T_{es}) * \nu \tag{5.2}$$

where $T_{es} = T_{ms} - D_s$ and $T_{ee} = T_{me} + D_e$ are the starting and ending time of the detected event moment as described in chapter 4 (Figure 4.10). D_3 represents the time duration between the end of an event and the start of the instant replay. We arbitrarily set D_3 to 1 second and this is adjustable. ν is a factor defining how slow the replay is displayed compared with real-time.

Then the system examines whether the time slot from T_{rs} to T_{re} in the subsequent no-event segment meets one of the following conditions:

- no/low motion;
- high motion but position not at area 2 in Fig. 4.3b – the penalty area.

If so, an instant replay is inserted.

5.2.3.2 Delayed replay generation

As we mentioned at the start of this section, delayed replays should be inserted for Cat. 1, 2 and 3 events. Currently the Cat. 3 events are unable to be processed by our system as they cannot be detected using the raw main camera video. Our replay generation system will buffer the Cat. 1 and 2 events and find suitable time slots to insert delayed replays. In addition, to identify whether an event is an Cat. 2 event, an importance measure I is given to the event based on the duration of its event moment as generally the longer the event moment, the more important the event:

$$I = T_{me} - T_{ms} \tag{5.3}$$

And those events with $I > 3$ *seconds* are deemed as important events. In our system, the threshold 3 seconds is selected empirically so that only 5% events detected become important ones. This ratio is consistent to broadcast video identification of important events.

The duration of the delayed replay is the same as the instant replay. The system will search in subsequent no-event segments for a time slot with $T_{re} - T_{rs}$ in length that meets the following condition:

- no motion;

If such a time slot is found, a delayed replay is inserted. This search continues until a suitable time slot is found for Cat. 1 event, or two delayed replays have been inserted for an Cat. 2 event, or a more important Cat. 2 event occurs.

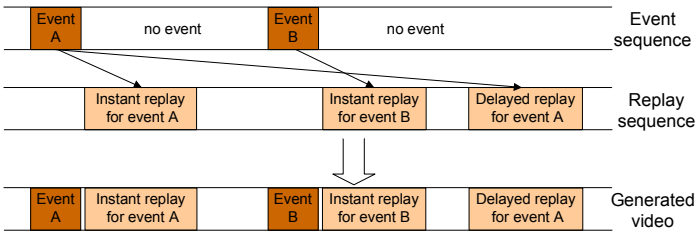


Figure 5.6: Replay structure

Once the proper time slots $[T_{rs}\ T_{re}]$ are found for replay insertion, the video segments of the events from different cameras are ranked by our proposed view selection criterion

to select the camera view for output. Depending on the duration of time slots found for replay insertion, one or several replay scenes sorted by the results of view selection criterion are launched.

5.2.4 Automatic Camera View Selection and Switching

To avoid the presence of long and uninteresting far view shots and provide more detailed game actions, the director often switches between main-camera and sub-camera captures during broadcasting process. The selection of which camera feed to launch into broadcasting is affected by both the video content and the broadcaster's experience. To simulate such operations, one possible solution is to define rules for an automatic system for camera view selection. For example, in a similar problem of composing lecture video from multi-camera input, Rui *et al.* [32] applied rule-based approach to control when and which camera to switch to. In broadcast soccer video composition, one rule may be to switch into a sub-camera view when a ball is near a player such that viewer would then have a close-up view of the action. Such rule based system is however difficult to implement as it implies that the system must be able to extract player and ball's semantics. As our implementation uses features containing no such information, we did not use rule-based approach for the camera selection procedure.

Instead, we consider other factors that determine camera capture selection. For this study, we collected a multi-camera soccer video data set consisting of three raw camera captures of one Singapore Soccer League (S-League) game (Camera 1, 2 and 3, Figure 4.1.e) and one recorded TV broadcasting of the same match, each 90 minutes long. Observation of this data set reveals that segments which are unclear, *e.g.*, those with very high camera pan/zoom motion, are never selected by the director (*e.g.*, Figure 5.7.a), while segments which possess clear game view and suitable converge may be selected (*e.g.*, Figure 5.7.b). Hence our solution for camera view selection and switching is to mimic the professional director's choice of using mainly the main-camera capture and at proper instances to launch sub-camera segment which has the clearest game view among all the other sub-cameras. Since no semantic information is utilized, a selected segment may not present desired content. However, as all the cameras are tracking the game actions, this lack of semantic information to select view may not be critical. The camera view

selection and switching problem is now divided into two sub-problems, specifically the *Suitable/unsuitable sub-camera segment classification*, and the *View switching instance and camera selection*. They are respectively discussed in the following subsections.

(a) (b)

Figure 5.7: Examples from sub-camera

(a) unsuitable segments; (b) suitable segments

5.2.4.1 Suitable/unsuitable Sub-Camera Segment Classification

This subsection describes our module to segment the sub-camera capture into alternate partitions of suitable and unsuitable segments for broadcast composition selection. We define a suitable segments as segments with clear view (*i.e.* not blurred images), and the unsuitable segments as the rest. Similar work has been reported previously, *e.g.*, Hua *et al.* [121] used light exposure and camera motion to identify low quality segments in home video. In our study, as the lighting condition in soccer stadium does not change significantly, we propose to only use camera motion feature for suitable/unsuitable camera capture segment classification.

The HMM based classifier is selected for the task as HMM is good at modeling temporal patterns. Two HMMs are used to classify each sub-camera capture, one for the suitable segments (*suitable HMM*) and one for the unsuitable segments (*unsuitable HMM*). In our implementation, each HMM model consists of 3 states with each state having 10 Gaussian mixtures as this setting generates best results for our data set. The feature used for the HMM classifiers is the R^5 camera motion (F_4, Table 4.2) from the relevant sub-camera video.

Our HMM training database consists of 4 video recordings as mentioned at the beginning of subsection 5.2.4, *i.e.*, the broadcast video, main-camera 1 (Figure 4.1.e) capture, left sub-camera 2 and right sub-camera 3 captures. The first half of these video clips

are used for training, each 45 minutes long. To train the *suitable* model, the original broadcast video of our database is first hand-labeled to mark the camera ID which produced the segment as illustrated in Figure 5.8: To train the *suitable HMM* of sub-camera 2, the training data consists of segments when sub-camera 2 is used for broadcast. To train the *unsuitable HMM* of sub-camera 2, the training data consists of frames from sub-camera 2 that corresponds to those segments that the main-camera (camera 1) was used for broadcast. Note that there are segments in camera 2's capture that are not used for training of either the *suitable* or *unsuitable* model, i.e., when camera 3 is used for broadcast instead.

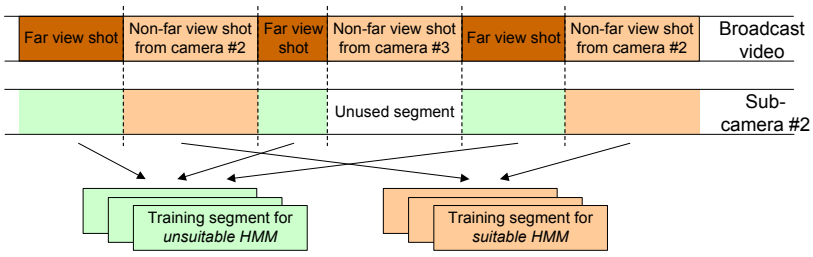


Figure 5.8: Training data for sub-camera 2

The grammar of the HMM classifer is illustrated in Figure 5.9. Every sub-camera will generate a sequence of *suitable* and *unsuitable* segments where each segment is an R^3 vector whose elements are the class label c_i= {*suitable, unsuitable*} of the segment, the absolute ending time t_i of the segment and the corresponding normalized likelihood score p_i for the segment which is computed by

$$p_i = \frac{1}{t_i - t_{i-1}} P(F_4([t_{i-1}+1, t_i])|HMM_{c_i}) \tag{5.4}$$

where $P(F_4([t_{i-1}+1, t_i])|HMM_{c_i})$ represents the likelihood score of camera motion feature F_4 segment $[t_{i-1}+1, t_i]$ for HMM_{c_i}. The obtained sequence for sub-camera j is denoted by F_6^j where

$$F_6^j = \{[c_i^j, t_i^j, p_i^j]\} \quad i = 1..N_j \tag{5.5}$$

where i denotes the i^{th} segment and N_j denotes the total number of segments found by sub-camera j's HMM classifier. Note that N_j may be different from sub-camera to sub-camera.

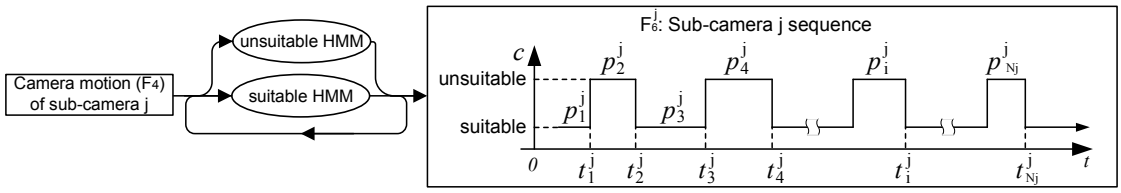


Figure 5.9: HMM for suitable/unsuitable segmentation

5.2.4.2 View Switching Instance and Camera Selection

To introduce our implementation of the view switching strategy, we discuss two scenarios: One for no-event (Figure 5.4.a) and the other for event occurrence (Figure 5.4.b) of a game separately.

For a no-event segment, we first select a suitable duration of the main-camera view and then switch to a sub-camera view, and repeat this alternate switching until the end of this no-event segment. For each switching of view, three parameters are to be evaluated, specifically the duration of the main-camera (denoted by k), the next sub-camera capture to use and the duration of the selected sub-camera view (denoted by l). To select the appropriate sub-camera, our idea is to compare the proposed duration of various sub-cameras and choose the one with the highest probability score for the view switching. To clarify our implementation, let us first assume that there is only one main-camera and one sub-camera (the j^{th} sub-camera), hence the problem is only to find the duration of the main and sub-camera for a particular segment. Fig.5.10 illustrates the search strategy.

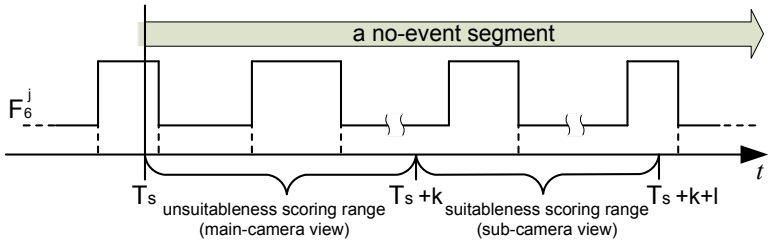


Figure 5.10: Searching for switching instance between a single main/sub-camera pair

To elaborated, let

$$\phi_k^j = \sum_{\substack{c_i^j \in \{unsuitable\} \\ t_i^j \in [T_s, T_s+k]}} p_i^j - \sum_{\substack{c_i^j \in \{suitable\} \\ t_i^j \in [T_s, T_s+k]}} p_i^j \tag{5.6}$$

represents the unsuitable likelihood score of sub-camera j for time duration $[T_s, T_s + k]$ where c_i^j, t_i^j and p_i^j are the three elements of the i^{th} component in the F_6 sequence (Eq.5.5) of sub-camera j, T_s is the starting time of the no-event segment. Empirically we found $[4, 24]$ seconds to be an appropriate range to search for the far view duration value k. Using the set of k values, we further compute the suitable likelihood score of sub-camera j for duration $[T_s + k, T_s + k + l]$

$$\varphi_{k,l}^j = \sum_{\substack{c_i^j \in \{suitable\} \\ t_i^j \in [T_s+k, T_s+k+l]}} p_i^j - \sum_{\substack{c_i^j \in \{unsuitable\} \\ t_i^j \in [T_s+k, T_s+k+l]}} p_i^j \tag{5.7}$$

where the variable $\varphi_{k,l}^j$ represents how much the duration $[T_s+k, T_s+k+l]$ is suitable for sub-camera j's view, and $[1.5, 8]$ seconds were empirically found to be a suitable range to search for the non-far view duration l. In this manner, the unsuitable likelihood score ϕ_k^j can be used to measures how much the duration $[T_s, T_s+k]$ is unsuitable for a sub-camera view, which implies that the main camera view is suitable. To find the optimum k and l values, we evaluate

$$\{k^j, l^j\} = \arg\max_{k,l}(\phi_k^j + \varphi_{k,l}^j) \tag{5.8}$$

To extend the strategy to a multiple sub-camera situation as in our problem, we find

$$j^* = \arg\max_{j}(\phi_{k^j}^j + \varphi_{k^j,l^j}^j) \tag{5.9}$$

to select the most suitable sub-camera j^*. The main-camera view is then selected for time duration $[T_s, T_s+k^{j^*}]$ and sub-camera j^*'s view is used for duration $[T_s+k^{j^*}, T_s+k^{j^*}+l^{j^*}]$ in the composed broadcast video. This operation is restarted from $T_s + k^{j^*} + l^{j^*} + 1$ and is repeated until the whole duration of the stated no-event segment is processed.

For an event segment, replays are to be generated and hence we evaluate the suitable likelihood score of the event segment from each sub-camera by

$$\varphi^j = \sum_{\substack{c_i^j \in \{suitable\} \\ t_i^j \in [Tms-Ds, Tme+De]}} p_i^j - \sum_{\substack{c_i^j \in \{unsuitable\} \\ t_i^j \in [Tms-Ds, Tme+De]}} p_i^j \tag{5.10}$$

where $[T_{ms} - D_s,\ T_{me} + D_e]$ are the starting and ending time of the event (Figure 4.10). We rank this suitable likelihood score φ^j to select one or several sub-camera segments for replay, depending on the duration of the time-slot available.

5.2.5 Experimental Results

5.2.5.1 Automatic Replay Generation

To evaluate the replay insertion performance, the three raw unedited camera recordings described in subsection 5.2.4 were used to compose a full-length broadcast soccer video. The recorded TV broadcasting of the same match is used as the ground truth to measure our replay insertion performance.

Table 5.5: Replay insertion performance

video	Our generation	Ground truth
total	61	31
same	26	
missed	5	
recall	83.9%	
precision	42.6%	

We count a generated replay as "same" as that in the broadcast video if replays are launched in both our generated video and the broadcast video for the same event, no necessary that the two replays should appear at exactly the same time. Table 5.5 lists the replay insertion results. Although our replay insertion module achieved 83.9% "Recall" score for replay insertion, it generated significantly more replays than a human director. The reasons for this are presented below:

The first reason is due to the subjective nature of director's choice rather than replays being launched once predefined conditions being met. For example, in our system, a replay would always be launched if a just-missing shoot ("Attack") event is detected and a suitable time slot is available. However, we observed that a human director may either ignore the event or instead choose a long close-up view of the disappointed player. Hence it is obvious that an automated system will generate more replays once the predefined conditions are met.

The second reason is due to the time constraint on the director to react and choose a replay selection. The strict time limit set to generate a replay means that a good replay segment selection might be missed. Hence, with the assistance of an automatic system, more replays may be generated.

The third reason for the excess in replay generation is due to occurrence of false event detection. Incorrect event detection may lead to unnecessary replays being carried out. Currently automatic systems are not able to detect events as accurately as humans. However, with minor human intervention this problem can be minimized.

5.2.5.2 Subjective User Study on Video Composition Quality

A subjective user study [145] is conducted to evaluate the overall quality of our automatic broadcast soccer video composition. The following 4 attributes are used in this study: *Necessary* to measure whether the inserted replay scenes are necessary. *Clarity* to measure whether the selected video segments for replay are clear and comprehensive, *e.g.*, the selected replay scenes are closely related to the event, the selected sub-camera shots are appropriate, etc. *Smoothness* to measure whether the view switching is smooth, and *Acceptance* to measure how much the viewer can accept the broadcast video.

The procedure of the user study is as follows: Firstly 19 segments (73 seconds to 161 seconds long) are randomly selected from the generated video composition of subsection 5.2.5.1 for viewer's scoring. Each viewer will watch at least 5 segments from the 19 chosen to score the first three attributes, *i.e. Necessary*, *Clarity* and *Smoothness* at five scale with score (5) corresponding to strongly accept, (4) accept, (3) marginally accept, (2) reject and (1) strongly reject. After the scoring of these three attributes for each segment, the TV broadcast video of the same time instance is shown to allow viewers to compare the quality of the automatic composition to professional production. This is scored in the *Acceptance* attribute. Note that we have specifically let viewers score the first three attributes prior to watching the professional production to remove potential bias.

In the first phase, 20 people were invited to this user study. These people are all research staffs or students of computer science and engineering, aged between 21-59, 18 male and 2 female. All of the 20 people watch soccer TV broadcasting, 17 are soccer fans but only 2 actually follow the two S-League teams in our generated video. In the second phase of the user study, we set up an on-line testing web-site [146] and 21 anonymous users have participated in the test. The average scores of these 41 people are listed in Table 5.6.

Table 5.6: Broadcast soccer composition quality

Necessary	Clarity	Smoothness	Acceptance
4.25	3.92	3.47	4.12

It is found from the user study that although our system generates significantly more replays than the human director's choice, the participants were satisfied with these replays as indicated by the high *Necessary* and *Acceptance* score in Table 5.6. We however observed that the *Clarity* and *Smoothness* scores in Table 5.6 are relatively low. This is caused by two reasons: First, the broadcast director could choose from 8 camera inputs (Figure 4.1.e) for composition while in our system, we have only 3 camera inputs available. The lack of camera selection limits our video composition possibility; Second, our proposed approach for view selection and switching is mainly based on the camera motion (F_4 in Table 4.2) features from which the semantics such as who the player is or what he is doing is not extracted, hence the selected sub-camera segments might not always contain the desired content of the event. This module can be improved by applying more advanced computer vision techniques such as face detection [147]/recognition [148], motion/gesture analysis [149], etc to generate semantically rich features for the camera selection module. The *Acceptance* score 4.12 (Table 5.6) demonstrates that our automatically generated broadcast soccer video composition is comparable to that produced by professionals.

5.2.5.3 Discussions

To evaluate other possibilities of our system, we examined the use of different broadcast rules to generate composition in the experiments. For example, we generated composition in which we switched to the replay scene right after the event was identified. This is different from TV broadcast video where a short duration of medium/close-up view shots are normally launched before the replay scenes. By immediately launching into replay, the generated video composition is also more likely to switch back to live game earlier than human director's selection, hence the potential loss of interesting live game actions could be minimized. After showing these clips to the viewers, it was observed that such replay scheme was however disagreeable among different people: Some opposed

our scheme because they preferred to have the medium/close-up shots which provide necessary information about the players reactions, while the others supported the scheme because the live game action is more interesting to them. In any case, as it is easy for our system to change the way the replay scenes are launched, different viewer could customize his/her favorable broadcast sports video composition by providing personalized profiles.

The above-mentioned video clips and the subjective user study documents can be found in the "Subjective test on automatic broadcasting" folder on the DVD attached to this thesis.

5.2.6 Conclusion

This section presented a novel system to automatically compose broadcast sports video using raw camera captures. Our experimental results showed that the broadcast sports video generated by our system is acceptable in the user study. The research of automatic broadcast sports video composition is valuable because for professional directors, an intelligent composition interface to collate and synchronize various video feeds will greatly simplify the broadcast video generation task and streamline the production process. In addition, this system will also benefit private subscribers who can customize the broadcast video content with respect to their personal preference and/or transmission channel characteristics. Furthermore we predict that the proliferation of customized sports segments will offer a potential market of game viewership.

5.3 Music Sports Video Composition

The advancement in computer hardware and networking has opened up new perspectives in the distribution of visual/audio content to public and personal subscribers. Home users in the near-future can expect the possibility to re-edit existing videos to produce customized sports video segments for themselves. However, current production of such personalized sports video remains difficult for both professional and amateur users for two reasons: First, the selection of interesting portions from voluminous sports video documents is time-consuming and labor-intensive; Second, editing video and music clips requires suitable tools, professional skills, experience and artistic talent, which the general

users do not usually possess. Hence, by introducing tools that can automate the sports video editing process, production efficiency can be greatly improved.

This section introduces our automatic MSV composition system. MSV is commonly seen in the prelude/postlude of a sports TV program, sports highlight, summary, MTV, etc. It also serves as a good example to demonstrate the functionality of an automatic sports video composition system because generating MSV involves video content analysis, music content analysis, video/music alignment, etc, operations that are typical in video composition.

To automatically compose MSV, two major research issues should be addressed, specifically the *Semantic sports video content extraction* and the *Automatic music video composition*. In chapter 3 we have proposed techniques to automatically extract interesting events from broadcast sports video. Hence in this section we only focus on the *Automatic music video composition* issue for MSV generation. The following subsections present an overview of our proposed system and describe our implementation of automatic MV composition.

5.3.1 System Overview

Figure 5.11 illustrates our two-level MSV generation system. Specifically, the low-level "semantic sports video content extraction" block uses techniques presented in chapter 3 to identify events/player/team from the sports video input for high-level composition. In the high-level "automatic music video composition" block, the "Music analysis" module analyzes the input music clips to obtain "beat", "lyric" and "semantic music structure" boundaries, and the "Content selection/matching" module uses Dynamic Programming (DP) based algorithms to select suitable video and music content to generate video-centric and music-centric MSVs.

5.3.2 Semantic Music Analysis

Analyzing the music content is a necessary task for MV composition. For example, previous work [30, 121] extracted the music "beat" to segment the music into sub-clips and matched the video segment to each music sub-clip. In our system, in addition to the music beat boundary, the lyric sentence boundary and the semantic music structure

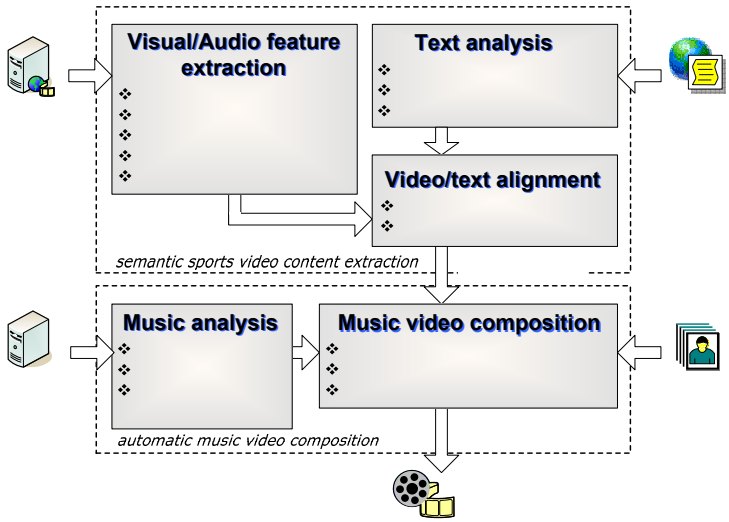


Figure 5.11: Personalized MSV generation system

boundary are also used to align the music to the video. The lyric sentence boundary is the start of each lyric sentence, and the semantic music structure specifies the boundary of successive semantic music partitions such as Intro, Verse, Chorus, Bridge, Instrumental and Outro [150]. By using these three boundary information, we are able to precisely specify different video contents for different music portions to enhance the artistic attribute of our generated MVs. The music structure analysis method by the I^2R [134] media research group [150] is used to extract the beat and semantic structure boundary. The "lyric" is obtained by collecting lyric with time-stamp from the Internet.

5.3.3 MSV Composition Scheme

To mimic the composition scenarios in professional MSV productions, a quantitative study using 35 professional soccer MSVs and 9 basketball MSVs (totally 298 minutes) was conducted. The database can be broadly classified into two MSV composition schemes:

Video-centric scheme The video-centric MSV is driven by the video content, and the music clips play a minor role in the video composition. Examples of the video-centric MSV include prelude and postlude of sports TV program, game league/season/series highlight, summary, etc. In the video-centric MSV composition process, the director's focus is to communicate the video content, *e.g.* a selection of events or players;

Music-centric scheme The music-centric MSV is driven by both video and music input. In addition, the artistic style to synchronize the video and music content is the focus in this scheme. One example of the music-centric MSV is the MVs using sports video content. To compose music-centric MSV, the director first selects the music and then selects suitable video segments to enrich the artistic attraction of the music to compose a music-centric MSV.

The following subsections discuss our implementations for the music-centric MSV composition and video-centric MSV composition respectively.

5.3.4 The Music-centric MSV Composition Scheme

To generate high quality MSV, we aim to mimic the current practices in professional MSV production. From our observation of the MSV database described in subsection 5.3.3, we created six rules for music-centric MSV composition:

Rule 1 The visual part of the MSV should consist of multiple scenes, and each scene will be made up by several shots; The musical part contains several semantic music structure [150], each structure includes several lyric sentences and each sentence covers several bars (1 bar equals to several beats, usually 4 or 8, depending on the music).

Rule 2 Selected shots from the same scene (event) should be presented chronologically. The music is played continuously from the beginning to the end without stoppages/cuts.

Rule 3 Shots are not repeated.

Rule 4 If a scene is to be used in the MSV, then more shots should be selected from the scene so that it is easier for the audience to understand the video presentation.

Rule 5 The video sequence and the music phrases should be synchronized, *i.e.*, the video shots transit only at the music phrases boundary (lyric sentence or bar boundary), and the shot motion and the music tempo are similar. To achieve this, the selected shots can be cut to match the duration of a music phrase. However, one should

present as much content of the original shot as possible to have better visual representation.

Rule 6 Personalization: Users should be able to exclude certain video contents from selection or allow to specify certain video contents to be launched at a specified instance. Users could also specify different video contents for different semantic music portions. For example, using landscapes scene for the "Intro" and use excited player or coach scenes for the "Chorus".

To perform the tempo matching between the music and video, a Greedy algorithm was introduced in our previous work [124]. The Greedy algorithm works in the following manner: Starting from the first music phrase, the video scene that obtains the highest matching score with the music phrase is picked out in each iteration. This process is repeated until all the music boundaries have been matched. Then the selected video segments are integrated together with the music to produce music-centric MSV.

To select the most suitable matching unit, a scoring scheme is defined to find the scene whose shot durations could perfectly match the length of respective music structures to reduce cutting. In addition, the motion of each shot is also computed and matched with that of the music such that the speed of video and music are similar. This idea is explained as follows:

Assume a video content pool $\boldsymbol{S} = \{\boldsymbol{S}^1, \boldsymbol{S}^2, ...\boldsymbol{S}^L\}$ contains L video scenes, and $\boldsymbol{S}^l = [\boldsymbol{s}^l_1, \boldsymbol{s}^l_2, ..., \boldsymbol{s}^l_{M_l}]$ consists of M_l shots in chronological order. To select a suitable scene, a matching score v_l for event $\boldsymbol{S}_l$ is computed as follows,

$$v_l = \frac{1}{M_l} \sum_{i=1}^{M_l} \bar{p}(\mathbf{X}|\Theta) \cdot r_{li} \cdot v_{li} \tag{5.11}$$

where $\mathbf{X}$ is the A/V features extracted from the search range to find event (scene) $\boldsymbol{S}_l$ (section 3.3.2) and $\bar{p}(\mathbf{X}|\Theta)$ represents the normalized probability score (Eq.3.12) which is used as a confidence for $\boldsymbol{S}_l$. The variable r_{li} is a binary value. It's set to 1 if the current scene $\boldsymbol{S}_l$ and current shot $\boldsymbol{s}_{li}$ contain the required semantic event, otherwise 0. The variable v_{li} measures the distance between a single shot $\boldsymbol{s}_{li}$ in the event $\boldsymbol{S}_l$ to the music phrase. Specifically, v_{li} is computed by

$$v_{li} = e^{\frac{-||\mathbf{T}_{li}-\mathbf{T}_m||^2}{2\mathbf{\Sigma}_m^2}} \tag{5.12}$$

where $\mathbf{T}_{li} = [D_{li}\ M_{li}] \in R^2$ represents the characteristic of the shot $\boldsymbol{s}_{li}$, and $\boldsymbol{T}_m = [D_m\ M_m] \in R^2$ represents the music phrase characteristic. The variable D_{li} is the duration (in seconds) of the shot $\boldsymbol{s}_{li}$ and M_{li} is the average motion intensity computed for the shot. The variable D_m is the duration (in seconds) of the music boundary, and M_m is a measure of music intensity. Although M_{li} and M_m are measures of different features, we use them in this form to correlate between visual features and music tempo. The variance $\Sigma_m^2 \in R^{2\times 2}$ is a diagonal matrix with its diagonal values empirically set to 10% of $\mathbf{T}_m$. This scoring scheme is applied to all the events in the video content pool. Each time the event that obtains the highest score is picked out. This process is repeated until all the music boundaries have been matched.

However, the Greedy algorithm cannot guarantee to find the global optimal solution. In addition, it is computationally expensive, and the semi-automatic operations are implemented using complicated if-else rules. In the following subsections, we propose a novel approach based on Dynamic Programming (DP) [151] algorithm to align the video to music instead. Our approach uses the DP to perform global optimization by locally optimizing a sub-problem and is different from the previous greedy algorithm in that it uses an additional data structure (table) with each entry in the table representing the optimal value for a particular sub-problem. The DP strategy has been successfully used in tasks such as string matching [151]. The string matching problem is very similar to our problem, where letters (the video shots) from a long source string (the video content pool) are selected to match a destination string (the music). In addition, our method allows user interventions to be inserted to control the shot selection/matching process in music-centric MSV composition. Thus the proposed system can operate in both the fully and semi-automatic mode [122].

5.3.4.1 Problem Formation

The algorithm to align video scenes to music phrases for music-centric MSV composition is presented in this subsection. Let a video content pool be represented by

$\boldsymbol{S} = [\boldsymbol{S}^1, \boldsymbol{S}^2, ... \ \boldsymbol{S}^L]$ with L scenes in random order, where scene $\boldsymbol{S}^l = [\boldsymbol{s}_1^l, \boldsymbol{s}_2^l, ..., \boldsymbol{s}_{M_l}^l]$ consists of M_l shots in chronological order. Hence

$$\boldsymbol{S} = [[\boldsymbol{s}_1^1, ..., \boldsymbol{s}_{M_1}^1], ..., [\boldsymbol{s}_1^L, ..., \boldsymbol{s}_{M_L}^L]] \tag{5.13}$$

In our implementation, we use the techniques described in Chapter 3 to extract event from sports video. Each event is a scene which consists of several shots. Let M denote the total number of shot in $\boldsymbol{S}$ and can be evaluated by

$$M = \sum_{l=1}^{L} M_l \tag{5.14}$$

If we ignore the scene information, Eq.5.13 can be rewritten as

$$\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_M] \tag{5.15}$$

In our approach, each $\boldsymbol{s}_i$, $i = 1, ..., M$ records a set of attributes of the shot. Currently the attributes used are

$$\boldsymbol{s}_i = \{\frac{d_i}{\max\{\{d_j\}_{j=1}^{M}, \{l_j\}_{j=1}^{N}\}}, \frac{m_i}{\max(m_j)_{j=1}^{M}}\} \tag{5.16}$$

where d_i and m_i are the duration (in seconds) and motion magnitude of shot $\boldsymbol{s}_i$, respectively. m_i is calculated by Eq.3.5. The variable l_j is the length of the music phrases as explained later.

Similarly, let a sequence of music $\boldsymbol{B}$ with N boundaries be denoted by

$$\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_N] \tag{5.17}$$

where $\boldsymbol{b}_i$ represents a single music phrase covering one lyric sentence duration or several music bars. Each $\boldsymbol{b}_i$, $i = 1, ..., N$ records a set of attributes of the music phrase. Currently the attributes adopted are

$$\boldsymbol{b}_i = \{\frac{l_i}{\max\{\{d_j\}_{j=1}^{M}, \{l_j\}_{j=1}^{N}\}}, \frac{t_i}{\max(t_j)_{j=1}^{N}}\} \tag{5.18}$$

where l_i and t_i are the duration (in seconds) and tempo [150] of phrase $\boldsymbol{b}_i$, respectively. The tempo of a music segment is computed by our semantic music analysis module.

The music-centric video composition scheme is to find N shots from $\boldsymbol{S}$ that best match the N music phrases in $\boldsymbol{B}$.

5.3.4.2 DP Algorithm

The DP-based algorithm is applied to find the best shots in $\boldsymbol{S}$ for the given music $\boldsymbol{B}$. Our method operates in the following manner: To find the globally optimum solution, the DP algorithm locally optimizes a sub-problem and uses a grid matrix $\boldsymbol{G}$ of M (row) $\times$ N (column) to record the intermediate matching score. To depict the searching process, let each component of $\boldsymbol{G}$ be denoted as $\boldsymbol{g}_i^j$, $i = 1, ..., M$, $j = 1, ..., N$ where

$$\boldsymbol{g}_i^j = \{d_i^j, r_i^j\} \tag{5.19}$$

consists of 2 components. The value d_i^j is the locally accumulated mismatch value between shot $\boldsymbol{s}_i$ and music phrase $\boldsymbol{b}_j$, and the value r_i^j is the row value of $\boldsymbol{g}_i^j$'s predecessor in $\boldsymbol{G}$. To minimize the accumulated mismatch value, r_i^j is found by

$$r_i^j = \arg \min_{k=1,...,M} (d_k^{j-1} + T(\boldsymbol{g}_k^{j-1}, \boldsymbol{g}_i^j)) \tag{5.20}$$

where $T(\boldsymbol{g}_k^{j-1}, \boldsymbol{g}_i^j)$ is the transition cost from grid $\boldsymbol{g}_k^{j-1}$ to grid $\boldsymbol{g}_i^j$. The form of $T()$ will be discussed later. The locally accumulated mismatch d_i^j (Eq.5.19) is computed by

$$d_i^j = d_{r_i^j}^{j-1} + T(\boldsymbol{g}_{r_i^j}^{j-1}, \boldsymbol{g}_i^j) + ||\boldsymbol{s}_i - \boldsymbol{b}_j||^2 + \tilde{d}_i^j \tag{5.21}$$

where $\tilde{d}_i^j$ is initialized to 0 by default for fully-automatic operation. In the semi-automatic working mode, $\tilde{d}_i^j$ can be set to ∞ to inhibit matching between shot $\boldsymbol{s}_i$ and music phrase $\boldsymbol{b}_j$. This is discussed later. The term $||\boldsymbol{s}_i - \boldsymbol{b}_j||^2$ is the Euclidean distance between shot $\boldsymbol{s}_i$ and music phrase $\boldsymbol{b}_j$ which measures the dissimilarity between the two.

To build $\boldsymbol{G}$, the DP algorithm starts from the left-most column of $\boldsymbol{G}$, *i.e.*, $\{\boldsymbol{g}_1^1, \boldsymbol{g}_2^1, ..., \boldsymbol{g}_M^1\}$. As the grids in this column have no predecessor, $r_i^1 = -1$, and therefore $d_i^1 = ||\boldsymbol{s}_i - \boldsymbol{b}_1||^2$. So we have

$$\boldsymbol{g}_i^1 = \{d_i^0, r_i^0\} = \{||\boldsymbol{s}_i - \boldsymbol{b}_1||^2, -1\} \quad i = 1, .., M \tag{5.22}$$

Then j is increased to 2 to compute grids $\{\boldsymbol{g}_1^2, \boldsymbol{g}_2^2, ..., \boldsymbol{g}_M^2\}$ using Eq.5.20 and Eq.5.21. This process is reiterated until $j = N$. The grid with the smallest d_i^M value in the last column indicates the optimal minimum accumulated mismatch. By tracing the chain of this grid's predecessors till the first column of $\boldsymbol{G}$, we can find the optimum alignment between the shots and music phrases.

5.3.4.3 Music-centric MSV Composition Example

To satisfy the six music-centric MSV composition rules, we constrain the search range for predecessors (Eq.5.20), the form of transition function $T()$ (Eq.5.21) and the initialization parameter $\tilde{d}_i^j$ (Eq.5.21) of DP. We give a typical video/music matching example as shown in Figure 5.13 to illustrate our algorithm. In this example, the user wants to select 8 shots from 7 soccer scenes to align with 8 music phrases. These 7 scene contain $\{5, 3, 3, 3, 4, 5, 4\}$ shots respectively and are separated by the horizontal dark lines as shown in Figure 5.13.a. The 8 music phrases belong to 3 semantic music structures, and each structure contains $\{3, 4, 1\}$ phrases respectively. These 3 structures are marked using vertical dark lines as illustrated in Figure 5.13.a.

The user has imposed the following restrictions in this example: shots for semantic music structure 1 ($\boldsymbol{b}_1$ to $\boldsymbol{b}_3$) can only be selected from scene 1, 2, 3 and 4, shots for semantic music structure 2 ($\boldsymbol{b}_4$ to $\boldsymbol{b}_7$) can only be selected from soccer scene 5, 6, 7, shots for semantic music structure 3 ($\boldsymbol{b}_8$) can be selected from all 7 soccer scenes, shot 23 ($\boldsymbol{s}_{23}$) should not be used, and shot 4 ($\boldsymbol{s}_4$) must be placed at music phrase 2.

In this example, as the user has specified shots inclusion/exclusion and placement requirement, the system is operating under the semi-automatic working mode. We first create the empty 27×8 grid matrix $\boldsymbol{G}$ as shown in Figure 5.13.a. The following paragraphs describe how we constrain the DP algorithm to meet all the six music-centric MSV composition rules.

Rule 1: This requirement is always satisfied because our algorithm represents the video/music using a hierarchical structure where "shot"/"bar" is the basic unit. (Eq.5.15 and Eq.5.17).

Rule 6: To exclude $\boldsymbol{s}_{23}$ from selection in Eq.5.21, we set

$$\tilde{d}_i^j = \infty \quad i = 23,\ j = 1, ..., 8$$

and hence DP will never select $\boldsymbol{s}_{23}$ for composition. Similarly, we set

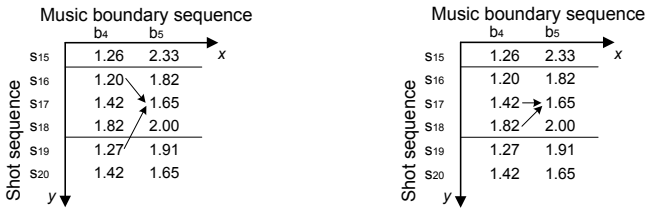
$$\tilde{d}_i^j = \infty \quad \begin{array}{l} i = 1, ..., 3, 5, ..., 27,\ j = 2 \\ and\ i = 4,\ j = 1, 3, ..., 8 \end{array} \tag{5.23}$$

to fix $\boldsymbol{s}_4$ to be placed at phrase $\boldsymbol{b}_2$, and set

$$\tilde{d}_i^j = \infty \quad \begin{array}{l} i = 1, ..., 14,\ j = 4, ..., 7 \\ and\ i = 15, ..., 27,\ j = 1, ..., 3 \end{array} \tag{5.24}$$

such that shots for music structure 1 ($\boldsymbol{b}_1$ to $\boldsymbol{b}_3$) are selected from soccer scene 1, 2, 3 and 4, and shots for music structure 2 ($\boldsymbol{b}_4$ to $\boldsymbol{b}_7$) are from soccer scene 5, 6, 7. The rest $\tilde{d}_i^j$ uses the default 0 value. Applying these initialization values for the semi-automatic working mode, the value of $\tilde{d}_i^j$ for each grid in $\boldsymbol{G}$ is initialized as illustrated in Figure 5.13.b.

Rule 2 and **Rule 3**: To satisfy the requirement that shots in the same scene (event) are presented chronologically, and a shot is never repeated, only the immediate prior shot of the same scene, or the 1^{st} shot of another scene can transit into the shot. Figure 5.12.b illustrated the forbidden transition. Therefore the selection of any grid's predecessor (Eq.5.20) has the following constrains:



(a) allowed transition path (b) invalid transition path

Figure 5.12: Transition path examples

- If shot $\boldsymbol{s}_i$ and $\boldsymbol{s}_r$ belong to the same scene, then

$$r_i^j \in R_1 = \{r | r < i,\ \boldsymbol{s}_r \in \boldsymbol{S}_l \ and\ \boldsymbol{s}_i \in \boldsymbol{S}_l\} \tag{5.25}$$

- If $\boldsymbol{s}_i$ and $\boldsymbol{s}_r$ belong to different events, then

$$r_i^j \in R_2 = \{r | r \neq i,\ \boldsymbol{s}_r \in \boldsymbol{S}_l,\ \boldsymbol{s}_i \in \boldsymbol{S}_m\ and\ l \neq m\} \tag{5.26}$$

Hence Eq.5.20 becomes

$$r_i^j = \arg \min_{r \in (R_1 \cap R_2)} (d_r^{j-1} + T(\boldsymbol{g}_r^{j-1}, \boldsymbol{g}_i^j)) \tag{5.27}$$

The requirement in **Rule 2** that the music is played continuously is always satisfied as any grid's predecessor is from its left neighboring column, and hence the selected music phrases are continuous.

Rule 4: To have more shots selected from the same scene, the system should transit from one shot to another shot of the same scene rather than to shots of another scene.

To achieve this, the transition cost function $T()$ (Eq.5.20) is used to penalize transitions from different scene. In our implementation, $T()$ is given the following form:

$$T(\boldsymbol{g}_r^{j-1}, \boldsymbol{g}_i^j) = F(D(r,i)) \tag{5.28}$$

where $D(r,i)$ is the distance between shot $\boldsymbol{s}_r$ and $\boldsymbol{s}_i$. Let shot $\boldsymbol{s}_r$ be the $\hat{r}^{th}$ shot of scene l, and $\boldsymbol{s}_i$ be the $\hat{i}^{th}$ shot of scene m, $D(r,i)$ is calculated by

$$D(r,i) = \begin{cases} (\hat{i} - \hat{r})/M^* & if\ l == m,\ same\ scene \\ (M_l - \hat{r} + \hat{i})/M^* & if\ l \neq m,\ different\ scene \end{cases} \tag{5.29}$$

where M_l is the total number of shot in scene l (Eq.5.13), $M^* = max\{M_1, ..., M_L\}$ is to normalize the distance, and hence $D(r,i) \in [0,1]$. $F()$ (Eq.5.28) should satisfy $F(0) = 0$ and $F(1) = 1$. We select $F(x)$ to be

$$F(x) = Log_2(x+1) \tag{5.30}$$

as the logarithm function has a convex shaped curve such that only transition between nearby grids can result in small cost, and thus more shots from the same scene can be selected. Now Eq.5.28 becomes

$$T(\boldsymbol{g}_r^{j-1}, \boldsymbol{g}_i^j) = \begin{cases} Log_2((\hat{i} - \hat{r})/M^* + 1) & l == m \\ Log_2((M_l - \hat{r} + \hat{i})/M^* + 1) & l \neq m \end{cases} \tag{5.31}$$

Rule 5: This requirement is always satisfied as the mismatch between any video shot $\boldsymbol{s}_i$ and music phrase $\boldsymbol{b}_j$ is computed as $||\boldsymbol{s}_i - \boldsymbol{b}_j||^2$ (Eq.5.20) and then accumulated by Eq.5.21. The selected video shots by DP algorithm minimize the global accumulated mismatch.

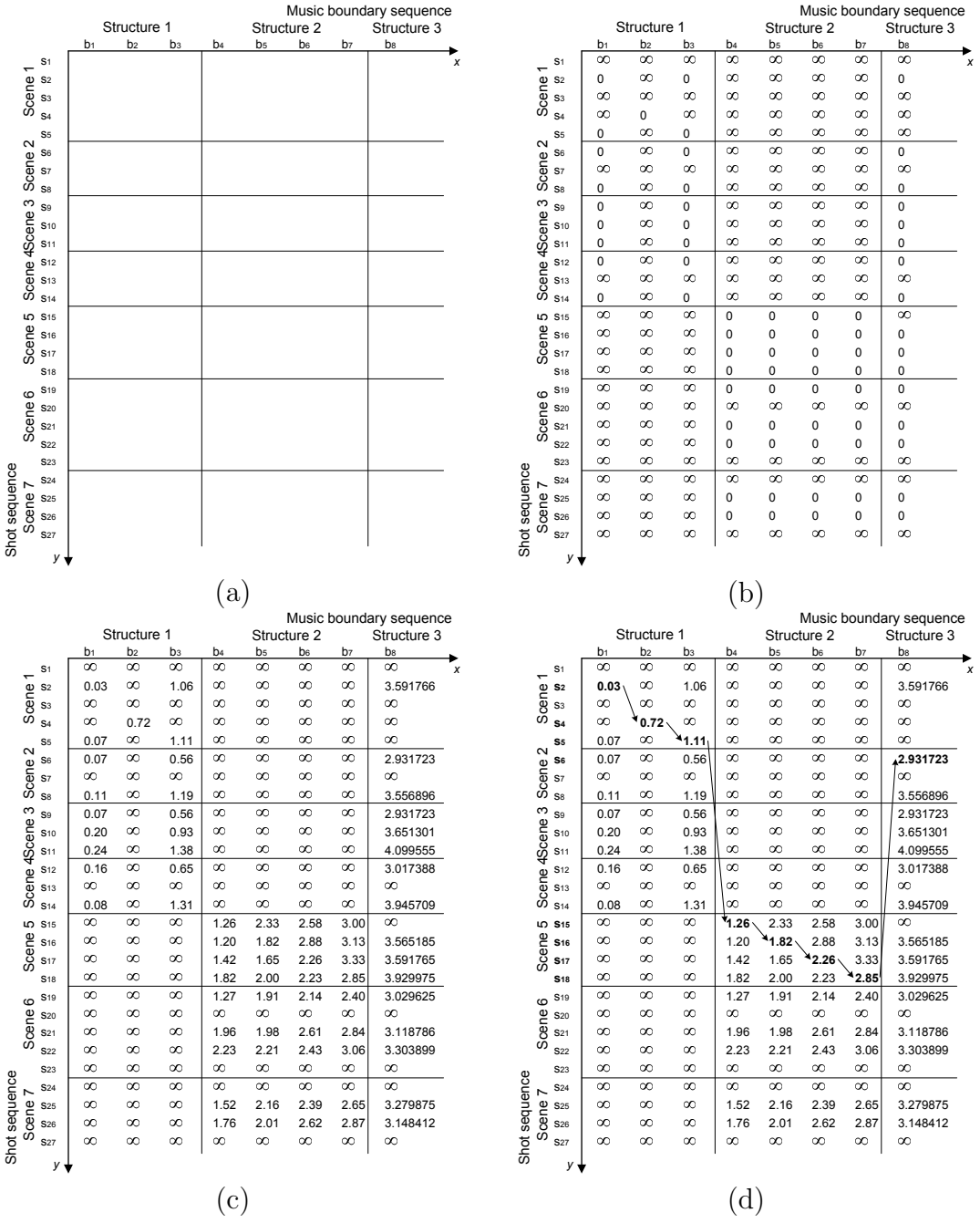


Figure 5.13: A shot selection and video/music matching example

(a) The empty Grid Matrix $\boldsymbol{G}$*; (b) The initial value* $\tilde{d}_i^j$ *of each grid for the semi-automatic working mode; (c) The local accumulated mismatch value* d_i^j *of each grid after Dynamic-Programming process; (d) The global optimum path*

With these constrains, the DP algorithm computes the value of each grid in $\boldsymbol{G}$ as shown in Figure 5.13.c. The minimal accumulated mismatch value is $d_6^8 = 2.931723$ at $\boldsymbol{g}_6^8$. Tracing back from $\boldsymbol{g}_6^8$, our algorithm finds that the best MSV composition result is to use the shots $\{\boldsymbol{s}_2, \boldsymbol{s}_4, \boldsymbol{s}_5, \boldsymbol{s}_{15}, \boldsymbol{s}_{16}, \boldsymbol{s}_{17}, \boldsymbol{s}_{18}, \boldsymbol{s}_6\}$ to match with the 8 respective music phrases. Figure 5.13.d highlights the best matching path.

5.3.5 The Video-centric MSV Composition Scheme

The video-centric MSV composition is simpler than composing a music-centric MSV. In a typical video-centric MSV composition process, the director selects video scenes with desired semantics from database, places these segments along the video track chronologically, and includes suitable music clips as the background music to complete a video-centric MSV production. The key difference is that it is not necessary to align the video to the music as that in music-centric MSV. To mimic such operation, our video-centric MSV composition module's primary task is to use the detected semantic of each event for video content selection with the time-stamp of each scene to sort shots for the composition. Background music is mixed into the video directly without further analysis.

Users may want the music phrases and the video shots to be aligned in order to improve the artistic attraction of the MSV. To align between video shots and music segments, the music cannot be trimmed otherwise the music will sound discontinuously. On the other hand, the video-centric MSV requires focus to be primary on the video, hence, it is not proper to discard any mismatched shot for alignment purpose as we did in the music-centric MSV composition. Our solution is to choose the music clips (or continues music portions) that best align with all the selected video shots and then trim the shots for alignment. This is similar to music-centric MSV composition but the difference is that no shot is discarded and the focus is to find music clips to match with video shots. This is easily achieved by modifying our above algorithm as explained in the following paragraphs.

5.3.5.1 Problem Formation

Given a sequence of chronologically placed video scenes $\boldsymbol{S}$ with totally N shots, we have

$$\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_N]$$

which is very similar to Eq.5.15.

Given L music clips $\boldsymbol{B} = [\boldsymbol{B}^1, \boldsymbol{B}^2, ... \ \boldsymbol{B}^L]$, each music clip $\boldsymbol{B}^l = [\boldsymbol{b}_1^l, \boldsymbol{b}_2^l, ..., \boldsymbol{b}_{M_l}^l]$, $l = 1, .., L$ containing M_l phrases. We have

$$\boldsymbol{B} = [[\boldsymbol{b}_1^1, ..., \boldsymbol{b}_{M_1}^1], ..., [\boldsymbol{b}_1^L, ..., \boldsymbol{b}_{M_L}^L]] = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_M] \tag{5.32}$$

where M is the total number of phrases. The attributes adopted for the shot and phrases are the same as that used in Eq.5.16 and Eq.5.18. The problem is to find N continuous music phrases that best match with the N shots.

5.3.5.2 Video-centric MSV Composition Example

To help understand our algorithm, we give another example where 6 music phrases are to be selected from 3 music clips to match 6 shots.

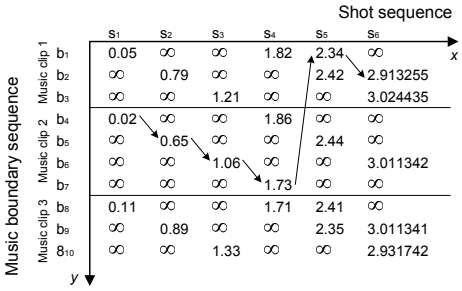


Figure 5.14: A music selection and video/music matching example

Each grid shows the accumulated mismatch value d_i^j. The global optimum path after Dynamic-Programming process is highlighted

Since it's now required to match the music phrases to the video shots, in contrast to the music-centric scheme, we put $\boldsymbol{S}$ as the top row and $\boldsymbol{B}$ as the left column as shown in Figure 5.14. The algorithm will find one or several music clips, N phrases in total, to match the N shots. The construction of the 2-D grid matrix $\boldsymbol{G}$ is similar to that used in music-centric MSV composition scheme, except that for any grid $\boldsymbol{g}_i^j = [d_i^j, r_i^j]$ (Eq.5.20), its predecessor's row r_i^j must satisfy:

- If phrase $\boldsymbol{b}_i$ is not the first phrase in music $\boldsymbol{B}^l$, $l = 1, ...L$,

$$r_i^j = i - 1 \tag{5.33}$$

- If phrase $\boldsymbol{b}_i$ is the first phrase in music $\boldsymbol{B}^l$ then

$$r_i^j = \arg\min_{r \in R_3}(d_r^{j-1} + T(\boldsymbol{g}_r^{j-1}, \boldsymbol{g}_i^j)) \tag{5.34}$$

where $R_3 = \{r | \boldsymbol{b}_r \textit{ is the last phrase in } \boldsymbol{B}^{\boldsymbol{k}}, k \neq l\}$ (Eq.5.32). And, $r_i^1 = -1$.

Such constraints guarantee that the predecessor of any grid (phrase) is either its left neighboring phrase or the ending phrase of another music clip, such that all the selected music clips can be played continuously. Note that, in such a manner, the transition cost $T(\boldsymbol{g}_r^{j-1}, \boldsymbol{g}_i^j)$ in Eq.5.34 is no longer effective as the distance between any grid $\boldsymbol{g}_i^j$ and its predecessor $\boldsymbol{g}_{r_i^j}^{j-1}$ is always $1/M^*$ (Eq.5.29).

The result found by our algorithm for the example is illustrated in Figure 5.14: music clip 2 ($\{\boldsymbol{b}_4, \boldsymbol{b}_5, \boldsymbol{b}_6, \boldsymbol{b}_7\}$) and the first two phrases of music clip 1 ($\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$).

5.3.6 Performance of MSV Composition

5.3.6.1 Objective Measure

To evaluate the performance of video/music matching, researchers have proposed various objective criteria. For example, in [121] the *Distribution Uniformity* attribute is introduced to measure the uniformity of the selected shots' distribution in home video composition. In our study, three objective measures are used, specifically, *Time* which measures the computation cost for a solution, *Inequality* which measures the average mismatch between shots and phrases, and *Integrality* which measures the integrality of the selected scenes. Particularity, *Inequality* and *Integrality* are computed as:

$$Inequality = \frac{1}{N}\sum_{i=1}^{N} ||\hat{\boldsymbol{s}}_i - \boldsymbol{b}_i||^2 \tag{5.35}$$

where $\hat{\boldsymbol{s}}_i$ is the i^{th} selected shot (Eq.5.16), $\boldsymbol{b}_i$ is the i^{th} music phrase (Eq.5.18) and N is the total number of phrases (Eq.5.17). The *Inequality* value ranges between 0 to 1 and smaller *Inequality* score indicates better performance.

$$Integrality = \frac{1}{K}\sum_{i=1}^{K} \frac{\hat{M}_i}{M_i} \tag{5.36}$$

where $\hat{M}_i$ and M_i are the number of selected shots and the number of total shots in scene S^i respectively, and K is the number of scenes with at least one shot selected. The *Integrality* value ranges between 0 to 1 and greater *Integrality* score indicates better performance.

We generated seven music-centric MSVs using our DP based algorithm to make objective comparison with our previous result [124] which is based on Greedy algorithm. We used the events from Euro-Cup 2004 and UEFA 2005 game videos to match the music songs. Figure 5.15 illustrates the comparison results. It shows that our approach in this current system achieves similar *Inequality* and *Integrality* scores but with much less processing time than [124]. Another advantage of our algorithm is that in [124] users could only amend the content of automatically generated MSV by using complicated *if-else* rules, while in this system such option is achieved using customized initialization parameters, hence production efficiency is improved.

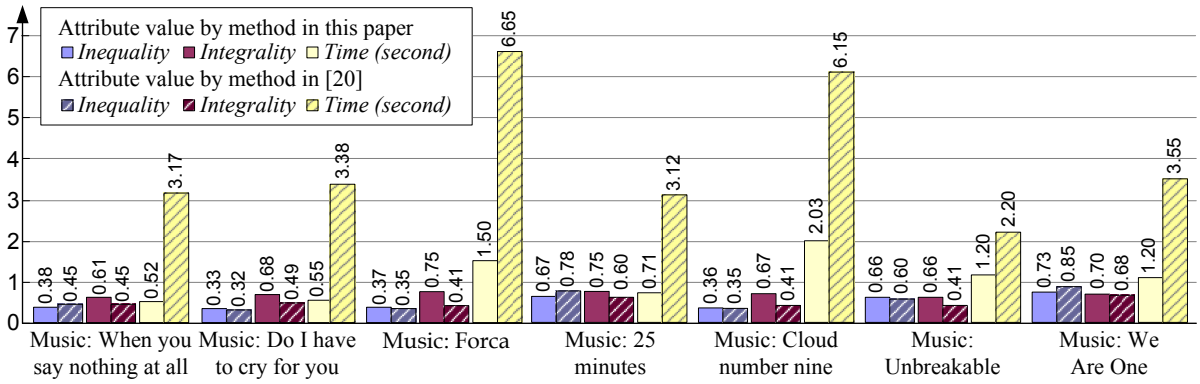


Figure 5.15: Performance of our proposed method

5.3.6.2 Subjective Measure

We employed a subjective user study to score the performance of our automatic MSV generation method against two methods. There are various attributes to evaluate a MV, and in this subjective MV quality test we adopted the *Clarity*, *Conciseness*, *Coherence* and *Overall Quality* [124] measures. 11 people were asked to view the generated videos and gave each clip a score for these four measures at five scale corresponding to strongly accept (5), accept (4), margin (3), reject (2) and strongly reject (1). These participants

are university graduate students or research staffs. To remove the potential biased evaluation results, we presented the MSVs generated by different methods in a random order, and the participants were not aware of the techniques used to generate each video clip. The average score of a MSV from all subjects is the final score of this MSV.

We first used the proposed music-centric method to generate seven soccer MSVs as described in subsection 5.3.6.1. In order to make relative comparison, we also used the muvee [31] software and our previous method to generate MSVs with the same music and video inputs. The results are listed in Table 5.7.

Table 5.7: Music-centric MSV quality

Music	Clarity/Conciseness/Coherence/Overall Qlty.		
	Our method	Previous method in [124]	muvee
M_1	4.66/4.33/4/4.66	4.17/3.8/4.25/4.5	3.15/3.5/3.5/3
M_2	4/4/4.33/4	3.75/3.5/4.25/4.25	3.25/3.75/3.5/3.75
M_3	4.33/4.33/4.33/4.33	3.8/4.25/3.75/4.25	3.33/3.65/3.75/3.8
M_4	4.5/4.25/4.25/4.25	4.5/4.25/4/4	3.25/3.25/2.75/3
M_5	4/3.75/3.75/3.75	3.5/3.5/3.75/3.75	2.75/3.25/3/3
M_6	3.75/4/3.75/3.75	3.75/3.75/3.75/3.75	2.75/3/3/3
M_7	4.25/4/4/4	3.75/3.75/4/3.75	3.25/3.5/3.5/3.25

M_1: "When you say nothing at all"; M_2: "Do I have to cry for you"; M_3: "Forca"; M_4: "25 minutes"; M_5: "Cloud number nine"; M_6: "Unbreakable"; M_7: "We Are One"

From the test the participants indicated that the MVs generated by our system showed good tempo matching between video and music, the visual presentation was better than MVs generated by muvee, and the overall quality was acceptable. One weakness with both the MVs by our system and by muvee software is the occurrence of abrupt changes in video content which results in the lower *Clarity* and *Conciseness* scores (Table 5.7). This phenomenon is especially notable in the MVs generated by muvee because the video shots selected by the software did not present semantic relationship with neighboring shots. The possible reasons that lead to our low *Clarity* and *Conciseness* score are: 1) Our MSV contains too many event types which made it difficult to understand and thus lowering the *Clarity* score. To improve the system, less types of events should be used; 2) Because of the requirement to match the music boundary, shots within an event was sometimes discarded. This resulted in incomplete event being used for the production,

and the *Clarity* and the *Conciseness* performances were affected. This limitation can be reduced by using larger video data sets and minor user intervention.

We next used our proposed video-centric method to generate seven MSV summaries as listed in Table 5.8 column one. Seven music clips were used to match the video contents. To make comparison, soccer MV summaries generated by professionals were also collected.

Table 5.8: Video-centric MSV quality

Content	Clarity/Conciseness/Coherence/Overall Qlty.
V_1	4.6/4.8/4.5/4.7
V_2	4.75/4.5/4.5/4.5
V_3	4.1/4.15/4.5/4.3
V_4	4/4/4.25/4
V_5	4.25/4/4.25/4.25
V_6	4.25/4.25/4.25/4.25
V_7	4.4/4.3/4.3/4.5
V_8	4.8/5/4.3/4.7

V_1: Card event from Euro-Cup 2004; V_2: Shot event from Chelsea-vs-Liverpool in UEFA 2005; V_3: C. Ronaldo from Euro-Cup 2004; V_4: F.Lampard in UEFA 2005; V_5: Team Portugal in Euro-Cup 2004; V_6: Club Bayernmunich (against Chelsea) in UEFA 2005; V_7: Team Germany in World-Cup 2002; V_8: A manually generated summary

Results from Table 5.8 shows that our generated video-centric MSV by event (V_1 and V_2) is comparable to the professional generated one (V_8). The *Conciseness* score of the manually generated clip V_8 is higher than V_1 and V_2 due to the fact that the professional producer can recognize and delete the undesired shots from the selection to make the production more concise. We also observed that the *Clarity* of video-centric MSV by player or team ($V_3, ..., V_7$) gets lower score compared with MSV by event (either manual or automatic generation). This is mainly because our current system is unable to identify if every single shot in an event is related to the required player/team or not. Hence the generated video sometimes contains shots of unexpected players or teams which affect the *Clarity* of the summary. However, as the unexpected video shots only take up a small portion of the generated video, the *Overall Qlty.* score of our automatic generated MSVs by player and team is only slightly lower than manually generated ones.

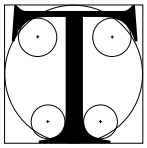
The above-mentioned MSV clips and the subjective user study documents can be found in the "Subjective test on music sports video" folder on the DVD attached to this thesis.

5.3.7 Conclusion

This section presents a novel application for fully/semi-automatic MSV composition. The experimental results show that the generated music-centric MSVs are better than previous approach [124] or automatic general video editing system [31], and the generated video-centric MSVs are comparable to professionals' production. This has obvious importance to reduce manual processing, and furthermore enable the generation of personalized MSVs. The generated video material can be used for customized soccer video summarization, soccer highlight generation, sports MTV production, etc. In addition, the MSV also brings positive effect on enjoyment of and motivation for performing physical exercise [152]. Besides, the proposed semantic soccer video content extraction approach can also be used for *Content Based Video Retrieval*.

Chapter 6

Conclusions and Future Work



HIS chapter summarizes the research topics and goals of the thesis and lists our major contributions. In addition, ideas for further research are discussed.

6.1 Research Topics and Goals

The objective of the research presented in this dissertation is semantic understanding and composition of multimedia content. We particularly focus on sports video to make use of domain-specific knowledge to improve the accuracy and robustness of our proposed techniques. The following challenges in content-based sports video analysis are addressed in our proposed sports video analysis technique:

- Generality: generic across several sports domains;
- Accuracy: satisfactory content analysis performance;
- Complexity: low computational cost for time-critical situations;
- Accessibility: automatic sports video composition/editing to provide personalized access to sports video content

6.2 Major Contributions

The main contributions of the thesis are our proposed framework to automatically analyze sports video contents and perform personalized sports video composition. The following paragraphs reiterate our achievements.

In our semantic sports video analysis framework, we classified the existing sports video into broadcast and non-broadcast categories and proposed different methods to analyze these two types of videos:

- To analyze broadcast sports video content, we proposed to utilize textual information to assist current state-of-art visual/audio feature based sports video analysis. Particularly, we mined web-casting text for sports event detection and applied text/video alignment to obtain accurate video event boundary. The advantages of our proposed techniques are as follows: First, the method is generic across several sports domains as web-casting text is available for many types of sports game. Second, the incorporation of web-casting text significantly improves the sports video event detection accuracy due to the exact text keywords of event. And third, analyzing web-casting text helps to extract event semantics such as players' names, the development of the event, etc.

- In addition to existing work which mainly focus on broadcast sports video, we also examined raw single main-camera capture for event detection. In particular, we proposed algorithms to generate several mid-level visual/audio representations and statistically model these mid-level classifications to detect high-level event and video event boundary from non-broadcast sports video.

With the extracted features and semantic information, we examined three novel sports video composition/editing applications:

- We introduced a live soccer highlight generation system to detect events from live broadcast soccer game, extract suitable video segment boundary for each detected event and generate customized highlight. The low computational cost of the system allows for real-time processing. We evaluated the prototype system using the most recent English Premiere League (EPL) and World-Cup 2006 games with promising results.

- We developed an automatic broadcast soccer video composition generation system to analyze multiple raw unedited soccer video inputs to compose broadcast soccer video. We used raw multi-camera Singapore Soccer League (S-League) videos to produce one broadcast soccer video composition, and the subjective user test indicated that the automatically generated broadcast sports video composition was comparable to that produced by professionals. Our system demonstrates the possibility of automating the broadcast sports video composition operation, and hence introduces new avenues of producing personalized sports video broadcasting medias.

- We presented a Music Sports Video (MSV) editing system that uses a scene-based video content selection and video/music matching algorithm to generate high quality MSV clips. The proposed system is capable of working in both fully and semi-automatic mode. By semi-automatic we mean that the user can control the inclusion/exclusion of certain video content, location of desired video content, etc typical video editing operations. The objective and subjective user study showed that our system meets user's expectations.

6.3 Future Work

There are many possible directions for further researches. The following paragraphs discuss some ideas.

6.3.1 Availability of Textual Information

Our proposed broadcast sports video analysis technique relies on analyzing web-casting text for sports video event detection. However, web-casting text or other textual information is not universally available. Hence a more generic content-based sports video analysis approach is necessary to extract textual information. One possibility is to perform speech recognition on the commentary speech [87] or to extract the caption text [40, 89] by Optical Character Recognition (OCR). We will examine this problem in future works.

If textual information is available, an interesting issue is to examine the fusion of textual feature with visual/audio features. One possibility is to perform text analysis and

visual/audio analysis together and combine the two to improve the overall event detection accuracy [97, 93]. In our system, an alternative approach is proposed which detects events using only text analysis and the visual/audio analysis is applied to locate video event boundary. The above-mentioned strategies achieved better performance than using just visual/audio analysis. However, in both cases, the accuracy drops when text analysis, visual/audio analysis or text/video alignment errors occur. Further investigation of more suitable information fusion method and better visual/audio analysis is necessary.

6.3.2 Generality of Non-Broadcast Sports Video Analysis

Since the set of events in different sports games are different, it is extremely difficult to define a universal set of mid-level visual/audio representations across various sports domains for analysis. In addition, even with similar representations defined, the way to generate these representations can be very different across different sports domains. For example, although the broadcast basketball video and soccer video can both be classified into far view, medium view and close-up view, the green pixel count measure used for soccer video (Chapter 3) won't apply for basketball game. In chapter 4, to detect events from non-broadcast sports video, our module used game-specific middle-level features for modeling. This however compromises the generality of the system to achieve good semantic analysis results. There are two possible ways to reduce this problem: First, discovering the set of features that are generic across different sports domains using huge amount of training data, or second, recognizing the sports video's genre and then applying different feature extraction and modeling. We will examine the effectiveness of these possibilities in future work.

6.3.3 Computational Performance

One important use of content-based sports video analysis system is to automatically recognize interesting content from voluminous video data, and to assist time-critical operation. One good example of such time-critical application is the broadcast soccer video composition system described in section 5.2. The system process multiple raw soccer video captures to compose a broadcast soccer video. As such system aims to

operate online, the algorithms need to be improved so that the system can operate in real-time.

One possible method to reduce the computational expense for sports video processing systems is to optimize the algorithms using domain-specific knowledge for a particular problem. For example, in section 5.1 we describe a live soccer highlight generation system using text analysis from professional web-casting text of BBC [100] and ESPN [98]. This allows us to reduce the number of features and enable us to develop a faster video event boundary modeling method which achieved almost real-time performance. Following this optimization mode in the future, we believe that the computational cost of sports video analysis system can be further reduced.

6.3.4 Personalized Sports Video Presentation

Personalized sports video presentation involves the selection and re-combination of sports video semantics according to individual user's profile, e.g. viewer's preference or transmission channel characteristic. To provide personalized sports content, an automatic system should be able to identify a large set of sports video semantics and provide a comprehensive and interactive interface for the users' selection. In this thesis, our focus is on the first issue. For the second issue, we have only conducted a few subjective user study to reveal certain patterns in viewers' selection or their favorite content and to evaluate the quality of our generated sports video segments (subsection 5.2.5.1, subsection 5.3.6.2, etc). We will look more into the interface design problem in future work.

6.3.5 Distribution of Sports Video Material

The delivery of sports video segments over new media channels such as 3G is an attractive option for both service-providers and consumers. Most of the existing commercial offerings are of three types: live SMS updates, live video streaming of the game over 3G, and post-game 3G short highlight video clips. Currently live 3G short video segment updates is not wide spread due to three reasons: concerns of broadcast rights, accuracy/acceptability of video, and cost. We will not discuss the broadcast right issue as it is not a research interest. As for the accuracy and acceptability of the video, we argue that this is a mindset issue. Traditionally, video highlights creation is part of

post-production. Extensive video editing is required to put together different interesting segments and an automatic system would therefore not be as good. However if the target audience is more keen on timeliness, then our proposed techniques to generate personalized video materials may suffice for an instant video alert market. As for cost concerns, since our prototype system uses ordinary computer equipment and only require minimal operator assistance, we expect the cost to be minimal.

Lastly, we foresee a significant growth in consumer client-based applications. With the advent of pervasive broadband/UMTS connectivity, IPTV, home media centers and plummeting cost of set-top OEM, we envision a great demand for both time-shifting and place-shifting video services. In the place-shifting setup, relevant video can be detected from a broadcast, segmented and transcoded over an IP channel to a mobile device. We believe that our proposed techniques can be improved to fit into these scenarios. The demand for mobile digital media to the mass market is expected to grow significantly in the near future. This research area of media analysis is a key enabler technology for the successful development of new applications. In short, interesting and exciting time lies ahead for researchers of this area.

Appendix A: Publications

The research that forms the major part of this thesis has led to the following publications:

A.1 Journal

[2] J. Wang, E. S. Chng, C. Xu, H. Lu and Q. Tian, "Generation of Personalized Music Sports Video using Multimodal Cues", *IEEE Trans. on MultiMedia*, vol. 9, issue 3, pp 576-588, April 2007

[1] J. Wang, C. Xu, E. S. Chng, H. Lu and Q. Tian , "Automatic Composition of Broadcast Sports Video", *ACM MultiMedia System Journal (To appear)*

A.2 Conference Paper

[12] C. Xu, J. Wang, K. Wan, Y. Li and L. Duan, "Live Sports Event Detection Based on Broadcast Video and Web-casting Text", *Proc. of ACM MultiMedia'06*, pp 221-230, USA, October, 2006

[11] J. Wang, C. Xu and E. S. Chng, "Automatic Sports Video Genre Classification using Pseudo-2D-HMM", *Proc. of IEEE ICPR'06*, pp 778-781, China, August, 2006

[10] J. Wang, E. S. Chng, C. Xu, H. Lu and X. Tong, "Identify Sports Video Shots With "Happy" or "Sad" Emotions" *Proc. of IEEE ICME'06*, Toronto, Canada, July, 2006

[9] J. Wang, E. S. Chng and C. Xu, "Fully and Semi-Automatic Music Sports Video Composition" *Proc. of IEEE ICME'06*, Toronto, Canada, July, 2006

[8] C. Xu, J. Wang, Q. Tian and H. Lu, "Sports Video Personalization for Consumer Products", *Proc. of IEEE ICCE'06*, Las Vegas, USA, 2006.

[7] J. Wang, C. Xu, E. S. Chng, L. Duan, K. Wan and Q. Tian, "Automatic Generation of Personalized Music Sports Video", *Proc. of ACM MultiMedia'05*, pp 735-744, Singapore, 2005

[6] X. Tong, L. Duan, C. Xu, H. Lu, J. Wang and J. Jin, "Periodicity Detection of Local Motion", *Proc. of IEEE ICME'05*, Amsterdam, Netherlands, July, 2005

[5] J. Wang, E. S. Chng and C. Xu, "Soccer Replay Detection using Scene Transition Structure Analysis", *Proc. of IEEE ICASSP'05*, vol. 2, pp. 433-436, Pennsylvania, USA, March 2005

[4] K. Wan, J. Wang, C. Xu and Q. Tian, "Automatic Sports Highlights Extraction with Content Augmentation", *Proc. of IEEE PCM'04*, vol. 3332, pp. 19-26. Tokyo, Japan, Dec. 2004

[3] J. Wang, C. Xu, E. S. Chng, K. Wah and Q. Tian, "Automatic Replay Generation for Soccer Video Broadcasting", *Proc. of ACM MultiMedia'04*, pp 32-39, New York, USA, 2004

[2] J. Wang, C. Xu, E. S. Chng, X. Yu and Q. Tian, "Event Detection Based on Non-Broadcast Sports Video", *Proc. of IEEE ICIP'04*, pp. 1637-1640, Singapore, 2004

[1] J. Wang, C. Xu, E. S. Chng and Q. Tian, "Sports Highlight Detection from Keyword Sequences using HMM", *Proc. of IEEE ICME'04*, pp. 599-602, Taipei, China, June 2004

References

[1] "http://www.nielsen-netratings.com/news.jsp?section=dat_gi,"

[2] "http://www.altavista.com/about/default/,"

[3] "http://www.youtube.com/t/about/,"

[4] "http://video.google.com/video_about.html,"

[5] D. Tjondronegoro, Y. Chen, and B. Pham, "Content-based video indexing for sports applications using multi-modal approach," *Proc. of ACM MultiMedia'05, Doctoral Symposium*, pp. 1035–1036, 2005.

[6] A. Demiris, M. Traka, E. Reusens, K. Walczak, C. Garcia, K. Klein, C. Malerczyk, P. Kerbiriou, C. Bouville, E. Boyle, and N. Ioannidis, "Enhanced sports broadcasting by means of augmented reality in mpeg-4," *Proc. of EURO IMAGE ICAV3D'01*, pp. 10–13, 2001.

[7] S. Smoliar and H. Zhang, "Content-based video indexing and retrieval," *IEEE Trans. on MultiMedia*, vol. 2, no. 1, pp. 63–75, 1994.

[8] S. F. Chang, "The holy grail of content-based media analysis," *IEEE Trans. on MultiMedia*, vol. 9, pp. 6–10, 2002.

[9] R. Leonardi, P. Migliorati, and M. Prandini, "Semantic indexing of soccer audio-visual sequences:a multimodal approach based on controlled markov chains," *IEEE Trans. on Circuits And Systems for Video Technology (CSVT)*, vol. 14, no. 5, pp. 34–43, 2004.

[10] N. Babaguchi, Y. Kawai, and T. Kitahashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Trans. on MultiMedia*, vol. 4, pp. 68–75, 2002.

[11] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," *Proc. of ACM MultiMedia'02*, pp. 105–115, 2002.

[12] K. Peker, R. Cabasson, and A. Divakaran, "Rapid generation of sports video highlights using the mpeg-7 motion activity descriptor," *Proc. of SPIE Conf. on Storage and Retrieval for Media Databases*, vol. 4676, pp. 318–323, 2000.

[13] Z. Xiong, R. Radhakrishnan, and A. Divakaran, "Generation of sports highlights using motion activity in combination with a common audio feature extraction framework," *Proc. of IEEE ICIP'03*, vol. 1, pp. 5–8, 2003.

[14] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden markov modelsl," *Proc. of IEEE ICIP'02*, pp. 609–612, 2002.

[15] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Robust dominant color region detection with applications to sports video analysis," *Proc. of IEEE ICIP'03*, vol. 1, pp. 21–24, 2003.

[16] P. Xu, L. Xie, S. Chang, A. Divakaran, A. Vetro, and H. Sun, "Algorithms and system for segmentation and structure analysis in soccer video," *Proc. of IEEE ICME'01*, pp. 928–931, Aug. 2001.

[17] 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. 24, 2003.

[18] K. Wan, X. Yan, X. Yu, and C. Xu, "Robust goal-mouth detection for virtual content insertion," *Proc of ACM MultiMedia'03*, pp. 468–469, 2003.

[19] K. Wan, J. Wang, C. Xu, and Q. Tian, "Automatic sports highlights extraction with content augmentation," *Proc. of IEEE PCM'04*, pp. 19–26, 2004.

[20] C. Xu, K. Wan, S. Bui, and Q. Tian, "Implanting virtual advertisement into broadcast soccer video," *Proc of IEEE PCM'04*, pp. 264–271, 2004.

[21] G. Pingali, Y. Jean, A. Opalach, and I. Carlbom, "Lucentvision:converting real world events into multimedia experiences," *Proc. of IEEE ICME'00*, 2000.

[22] N. Adami, R. Leonardi, and P. Migliorati, "An overview of multi-modal techniques for the characterization of sport programmes," *Proc. of SPIE VCIP'03*, pp. 1296–1306, 2003.

[23] A. Hauptmann and M. Smith, "Text, speech, and vision for video segmentation:the informedia project," *In AAAI Symposium on Computational Models for Integrating Language and Vision*, pp. 10–12, 1995.

[24] M. Christel, M. Smith, C. Taylor, and D. Winkler, "Evolving video skims into useful multimedia abstractions," *Proc. of CHI 98, ACM*, pp. 171–178, 1998.

[25] N. Babaguchi, Y. Kawai, T. Ogura, and T.Kitahashi, "Personalized abstraction of broadcasted american football video by highlight selection," *IEEE Trans. on MultiMedia*, pp. 575–586, Aug. 2004.

[26] C. Juan, A. Long, B. Myers, R. Bhatnagar, S. Stevens, L. Dabbish, D. Yocum, and A. Corbett, "Simplifying video editng using metadata," *Symposium on Designing Interactive Systems 2002*, pp. 157–166, 2002.

[27] E. Andrade, J. Woods, E. Khan, and M. Ghanbari, "Region-based analysis and retrieval for tracking of semantic objects and provision of augmented information in interactive sport scenes," *IEEE Trans. on MultiMedia*, pp. 1084–1096, Dec. 2005.

[28] M. Davis, "Editing out video editing," *IEEE Trans. on MultiMedia*, vol. 10, no. 2, pp. 54–64, 2003.

[29] X. Hua, L. Lu, and H. Zhang, "Automatic music video generation based on temporal pattern analysis," *Proc. of ACM MultiMedia'04*, pp. 472–475, 2004.

[30] J. Foote, M. Cooper, and A. Girgensohn, "Creating music videos using automatic media analysis," *Proc. of ACM MultiMedia'02*, pp. 553–560, 2002.

[31] MuVee Technologies Pte. Ltd, "Muvee^{TM}," 2000.

[32] Y. Rui, A. Gupta, J. Grudin, and L. He, "Automating lecture capture and broadcast: Technology and videography," *Multimedia Systems*, pp. 3–15, 2004.

[33] Y. Gong, H. Chua, and T. Lim, "An automatic video parser for tv soccer games," *Proc. of Asian Conf. on Computer Vision'95*, vol. 2, pp. 509–513, 1995.

[34] J. Assfalg, M. Bertini, C. Colombo, A. Bimbo, and W. Nunziati, "Semantic annotation of soccer videos:automatic highlights identification," *Computer Vision and Image Understanding (CVIU)*, vol. 92, no. 2-3, pp. 285–305, 2003.

[35] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Trans. on Image Processing*, vol. 12:7, no. 5, pp. 796–807, 2003.

[36] H. Miyamori and S. Iisaku, "Video annotation for content-based retrieval using human behavior analysis and domain knowledge," *Proc. of IEEE ICAFG'00*, pp. 320–325, 2000.

[37] D. Zhong and S. Chang, "Structure analysis of sports video using domain models," *Proc. of ICME'01*, 2001.

[38] W. Zhou, A. Vellaikal, and J. Kuo, "Rule-based video classification system for basketball video indexing," *Proc. of ACM MultiMedia'00*, pp. 213–216, 2000.

[39] M. Petrovic, V. Mihajlovic, W. Jonker, and S. Djordievic-Kajan, "Multi-modal extraction of highlights from tv formula 1 programs," *Proc. of IEEE ICME'02*, 2002.

[40] V. Mihajlovic and M. Petrovic, "Automatic annotation of formula 1 races for content-based video retrieval," *Technical report, TR-CTIT-01-41*, 2001.

[41] H. Denman, N. Rea, and A. Kokaram, "Content based analysis for video from snooker broadcasts," *Proc. of CIVR'02*, pp. 198–205, 2002.

[42] N. Benjamas, N. Cooharojananone, and C. Jaruskulchai, "Flashlight and player detection in fighting sport for video summarization," *Proc. of IEEE Symposium on Communications and Information Technology*, pp. 441–444, 12-14 Oct. 2005.

[43] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, pp. 679–698, 1986.

[44] H. Tamura, S. Mori, and T. Yamawaki, "Texture features corresponding to visual perception," *IEEE Trans. on System Man and Cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.

[45] L. Wang, B. Zeng, S. Lin, G. Xu, and H.-Y. Shum, "Automatic extraction of semantic colors in sports video," *Proc. of IEEE ICASSP'04*, pp. iii–617–20, May 2004.

[46] L. Duan, M. Xu, T. Chua, Q. Tian, and C. Xu, "A mid-level representation framework for semantic sports video analysi," *Proc. of ACM MultiMedia'03*, pp. 33–44, 2003.

[47] H. Pan, B. Li, and M. Sezan, "Detection of slow-motion replay segments in sports video for highlights generation," *Proc. of IEEE ICASSP'01*, pp. 1649–1652, 2001.

[48] V. Kobla, D. DeMenthon, and D. Doermann, "Identifying sports videos using replay, text, and camera motion features," *Proc. of SPIE Conf. on Storage and Retrieval for Media Databases*, vol. 3972, pp. 332–343, 2000.

[49] V. Kobla, D. DeMenthon, and D. Doermann, "Detection of slow-motion replay sequences for identifying sports videos," *Proc. of IEEE Workshop on Multimedia Signal Processing*, 1999.

[50] N. Nitta and N. Babaguchi, "Automatic story segmentation of closed-caption text for semantic content analysis of broadcasted sports video," *Proc. of International Workshop on Multimedia Information Systems'02*, pp. 110–116, 2002.

[51] N. Babaguchi and N. Nitta, "Intermodal collaboration:a strategy for semantic content analysis for broadcasted sports video," *Proc. of IEEE ICIP'03*, vol. 1, pp. 13–16, 2003.

[52] H. Pan, B. Li, and M. I. Sezan, "Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions," *Proc. of IEEE ICASSP'02*, pp. 3385–3388, 2002.

[53] X. Tong, H. Lu, Q. Liu, and H. Jin, "Replay detection in broadcasting sports videos," *Proc. of ICIG'04*, 2004.

[54] N. Babaguchi, Y. Kawai, Y. Yasugi, and T. Kitahashi, "Linking live and replay scenes in broadcasted sports video," *Proc. of ACM MultiMedia'00 workshop on Multimedia Information Retrieval (MIR)*, pp. 205–208, 2000.

[55] S. Choi, Y. Seo, H. Kim, and K. Hong, "Where are the ball and players? :soccer game analysis with color-based tracking and image mosaick," *Proc. of ICIAP'97*, pp. 196–203, 1997.

[56] X. Yu, C. Xu, H. Leong, Q. Tian, Q. Tang, and K. Wan, "Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video," *Proc. of ACM MultiMedia' 03*, pp. 11–20, 2003.

[57] T. Bebie and H. Bieri, "Reconstructing soccer game from video sequence," *Proc. of IEEE ICIP'98*, pp. 898–902, 1998.

[58] Y. Ohno, J. Miura, and Y. Shirai, "Tracking players and estimation of the 3d position of a ball in soccer games," *Proc. of IEEE ICPR'00*, vol. 1, pp. 145–148, 2000.

[59] K. Matsumoto, S. Sudo, H. Saito, and S. Ozawa, "Optimized camera viewpoint determination system for soccer game broadcasting," *Proc. of IAPR Workshop on Machine Vision Applications (MVA'00)*, pp. 115–118, 2000.

[60] T. Orazio, N. Ancona, G. Cicirelli, and M. Nitti, "A ball detection algorithm for real soccer image sequences," *Proc. of IEEE ICPR'02*, pp. 210–213, 2002.

[61] H. Saito, N. Inamoto, and S. Iwase, "Sports scene analysis and visualization from multiple-view video," *Proc. of IEEE ICME'04*, pp. 1395–1398, 27-30 June 2004.

[62] M. Takahashi, T. Misu, M. Tadenuma, and N. Yagi, "Real-time ball trajectory visualization using object extraction," *Proc. of the 2nd IEE European Conference on Visual Media Prodcution*, 2005.

[63] "http://www.hawkeyeinnovations.co.uk/,"

[64] F. Yan, W. Christmas, and J. Kittler, "A maximum a posteriori probability viterbi data association algorithm for ball tracking in sports video," *Proc. of IEEE ICPR'06*, pp. 279–282, 2006.

[65] J. Wang and P. Nandan, "Detecting tactics patterns for archiving tennis video clips," *Proc. of the 6th IEEE Symposium on Multimedia Software Engineering*, pp. 186–192, 13-15 Dec. 2004.

[66] M. Xu, J. Orwell, L. Lowey, and D.Thirde, "Architecture and algorithms for tracking football players with multiple cameras," *Proc. of IEE Vision, Image and Signal Processing*, pp. 232–241, 8 April 2005.

[67] J. Wang, C. Xu, E. S. Chng, K. Wan, and Q. Tian, "Automatic replay generation for soccer video broadcasting," *Proc of ACM MultiMedia'04*, pp. 31–38, 2004.

[68] B. Li and M. Sezan, "Event detection and summarization in american football," *Proc. of SPIE Conf. on Storage and Retrieval for Media Databases*, pp. 202–213, 2002.

[69] A. Demiris, G. Diamantakos, K. Walczak, E. Reusens, P. Kerbiriou, K. Klein, C. Garcia, I. Marchal, J. Wingbermuhle, E. Boyle, W. Cellary, and N. Ioannidis, "Piste:mixed reality for sports tv," *Proc. of International Workshop on Very Low Bitrate Video Coding'01*, 2001.

[70] G. Sudhir, J. Lee, and A. Jain, "Automatic classification of tennis video for high-level content-based retrieval," *Proc. of IEEE International Workshop on Content Based Access of Image and Video Database*, pp. 81–90, 1998.

[71] Y. Kang, J. Lim, Q. Tian, and M. Kankanhallis, "Soccer video event detection with visual keywords," *Proc. of IEEE PCM'03*, 2003.

[72] J. Hayet, J. Piater, and J. Verly, "Fast 2d model-to-image registration using vanishing points for sports video analysis," *Proc. of IEEE ICIP'05*, pp. III–417–20, 11-14 Sept. 2005.

[73] Y. Tan, D. Saur, S. Kulkami, and P. Ramadge, "Rapid estimation of camera motion from compressed video with application to video annotation," *IEEE Trans. on Circuits and Systems for Video Technology (CSVT)*, vol. 10.1, pp. 133–146, 2000.

[74] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, no. 1-3, pp. 185–203, 1981.

[75] J. Huang, Z. Liu, Y. Wang, Y. Chen, and E. Wong, "Integration of multimodal features for video scene classification based on hmm," *Proc. of IEEE Workshop on Multimedia Signal Processing*, 1999.

[76] Y. Ma and H. Zhang, "Motion pattern based video classification using support vector machines," *Proc. of IEEE International Symposium on Circuits and Systems, Theme:Circuits and Systems for Ubiquitous Computing (ISCAS'02)*, 2002.

[77] K. Wan, J. Lim, C. Xu, and X. Yu, "Real-time camera field-view tracking in soccer video," *Proc. of IEEE ICASSP'03*, vol. 3, pp. 6–10, 2003.

[78] K. Wan and C. Xu, "Efficient multimodal features for automatic soccer highlight generation," *Proc. of IEEE ICPR'04*, pp. 973–976, 2004.

[79] K. Wan and C. Xu, "Robust soccer highlight generation with a novel dominant-speech feature extractor," *Proc. of IEEE ICME'04*, 2004.

[80] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, "Comparing mfcc and mpeg-7 audio features for feature extraction, maximum likelihood hmm and entropic prior hmm for sports audio classification," *Proc. of IEEE ICASSP'03*, vol. 1, 2003.

[81] D. Zhang and D. Ellis, "Detecting sound events in basketball video archive," *http://www.ctr.columbia.edu/ dpwe/courses/e6820-2001-01/projects/dqzhang.pdf.*

[82] M. Xu, N. Maddage, C. Xu, M. Kankanhalli, and Q. Tian, "Creating audio keywords for event detection in soccer video," *Proc. of IEEE ICME'03*, vol. 2, pp. 281–284, 2003.

[83] S. Takao, T. Haru, and Y. Ariki, "Summarization of news speech with unknown topic boundary," *Proc. of IEEE ICME'01*, Aug. 2001.

[84] W. Greiff, A. Morgan, R. Fish, M. Richards, and A. Kundu, "Fine-grained hidden markov modeling for broadcast-news story segmentation," *Proc. of ACM MultiMedia'01*, pp. 1–5, 2001.

[85] "http://en.wikipedia.org/wiki/closed_captioning/,"

[86] Y. Chang, W. Zeng, I. Kamel, and R. Alonso, "Integrated image and speech analysis for content-based video indexing," *Proc. of IEEE International Conf. on Multimedia Computing and Systems*, pp. 306–313, 1996.

[87] Y. Ariki, T. Shigemori, T. Kaneko, J. Ogata, and M. Fujimoto, "Live speech recognition in sports games by adaptation of acoustic model and language model," *Proc. of EURO Speech'03*, pp. 1453–1456, 2003.

[88] "http://viplab.dsi.unifi.it/assavid/,"

[89] D. Chen, K. Shearer, and H. Bourlard, "Video ocr for sport video annotation and retrieval," *Proc. of IEEE International Conf. on Mechatronics and Machine Vision in Practice*, no. 28, pp. 57–62, 2001.

[90] D. Zhang and S.-F. Chang, "Event detection in baseball video using superimposed caption recognition," *Proc. of ACM MultiMedia'02*, pp. 315–318, 2002.

[91] Y. Li, C. Xu, K. W. Wan, X. Yan, and X. Yu, "Reliable video clock time recognition," *Proc. of IEEE ICPR'06*, 2006.

[92] H.-C. Shih and C.-L. Huang, "A robust superimposed caption box content understanding for sports videos," *Proc. of the 8th IEEE International Symposium on Multimedia*, pp. 867–872, Dec. 2006.

[93] B. Li, J. Errico, H. Pan, and M. I. Sezan, "Bridging the semantic gap in sports," *Proc. of SPIE Conf. on Storage and Retrieval for Media Databases*, pp. 314–326, 2003.

[94] B. Li and I. Sezan, "Semantic sports video analysis:approaches and new applications," *Proc. of IEEE ICIP'03*, pp. 17–20, 2003.

[95] B. Li, J. Errico, H. Pan, and I. Sezan, "Bridging the semantic gap in sports video retrieval and summarization," *Journal of Visual Communication and Image Representation*, pp. 393–424, Sept. 2004.

[96] B. Li, H. Pan, and I. Sezan, "A general framework for sports video summarization with its application to soccer," *Proc. of IEEE ICASSP'03*, pp. 169–172, Apl. 2003.

[97] H. Xu and T. Chua, "The fusion of audio-visual features and external knowledge for event detection in team sports video," *Proc. of ACM MultiMedia'04 workshop on Multimedia Information Retrieval (MIR)*, 2004.

[98] "http://www.soccernet.com/,"

[99] "http://fifaworldcup.yahoo.com/,"

[100] "http://news.bbc.co.uk/sport1/hi/football/teams/,"

[101] "http://sports.espn.go.com/nba/,"

[102] W. Adams, G. Iyengar, C. Lin, M. Naphade, C. Neti, H. Nock, and J. Smith, "Semantic indexing of multimedia content using visual, audio and text cues," *Eurasip Journal on Applied Signal Processing*, vol. 2, pp. 170–185, 2003.

[103] N. Rea, R. Dahyot, and A. Kokaram, "Modeling high level structure in sports with motion driven hmms," *Proc. of IEEE ICASSP'04*, pp. iii–621–4, May 2004.

[104] J. Assfalg, M. Bertini, A. D. Bimbo, W. Nunziati, and P. Pala, "Soccer highlights detection and recognition using hmms," *Proc. of IEEE ICME'02*, 2002.

[105] G. Xu, Y.-F. Ma, H.-J. Zhang, and S. Yang, "Motion based event recognition using hmm," *Proc. of IEEE ICPR'02*, pp. 831–834, 2002.

[106] G. Xu, Y. Ma, H. Zhang, and S. Yang, "A hmm based semantic analysis framework for sport game event detection," *Proc. of IEEE ICIP'03*, vol. 1, pp. 25–28, 2003.

[107] L. Xie, S. Chang, A. Divakaran, and H. Sun, "Unsupervised discovery of multi-level statistical video structures using hierarchical hidden markov models," *Proc. of IEEE ICME'03*, 2003.

[108] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Feature selection for unsupervised discovery of statistical temporal structures in video," *Proc. of IEEE ICIP'03*, vol. 1, pp. 29–32, 2003.

[109] V. Vapnik, "The nature of statistical learning theory," *Springer*, 2000.

[110] R. Leonardi, P. Migliorati., and M. Prandini, "A markov chain model for semantic indexing of sport program sequences," *Proc. of International Workshop on Image Analysis for Multimedia Interactive Services 2003*, 2003.

[111] M. Han, W. Hua, W. Xu, and Y. Gong, "An integrated baseball digest system using maximum entropy method," *Proc. of ACM MultiMedia'02*, pp. 347–350, 2002.

[112] J. Assfalg, M. Bertini, C. Colombo, A. D. Bimbo, and W. Nunziati, "Automatic extraction and annotation of soccer video highlights," *Proc. of IEEE ICIP'2003*, vol. 2, pp. 527–530, 2003.

[113] H. Shih and C. Huang, "A semantic network modeling for understanding baseball video," *Proc. of IEEE ICASSP'03*.

[114] A. Ekin, A. Tekalp, and R. Mehrotra, "Extraction of semantic description of event using bayseian network," *Proc. of IEEE ICIP'01*, pp. 1585–1588, 2001.

[115] H. Michael, A. Thorsten, and OhmRainer, "Application of mpeg-7 descriptors for content-based indexing of sports videos," *Proc of SPIE VCIP'03*, vol. 5150, pp. 1317–1328, 2003.

[116] A. Hanjalic, "Adaptive extraction of highlights from a sport video based on excitement modeling," *IEEE Trans. on MultiMedia*, pp. 1114–1122, Dec. 2005.

[117] D. Tjondronegoro, Y.-P. Chen, and B. Pham, "Highlights for more complete sports video summarization," *IEEE Trans. on Multimedia*, pp. 22–37, Oct.-Dec. 2004.

[118] C. Snoek and M. Worring, "Multimodal video indexing:a review of the state-of-the-art," *ISIS Technical Report Series, Intelligent Sensory Information Systems Group, University of Amsterdam*, vol. 2001-20, 2001.

[119] T. Koyama, I. Kitahara, and Y. Ohta, "Live mixed-reality 3d video in soccer stadium," *Proc. of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality ISMAR'03*, pp. 178–187, 2003.

[120] X. Qiu, Z. Wang, S. Xia, and Y. Wu, "Virtual-real comparison technique used on sport simulation and analysis," *Proc. of the 7th International Conference on Signal Processing*, pp. 1296–1300, 31 Aug. 2004.

[121] X. Hua, L. Lu, and H. Zhang, "Ave:automated home video editing," *Proc. of ACM MultiMedia'03*, pp. 490–497, 2003.

[122] A. Girgensohn, J. Boreczky, P. Chiu, J. Doherty, J. Foote, G. Golovchinsky, S. Uchihashi, and L. Wilcox, "A semi-automatic approach to home video editing," *Proc. of UIST'00, ACM Press*, pp. 81–89, 2000.

[123] Y. Ariki, S. Kubota, and M. Kumano, "Automatic production system of soccer sports video by digital camera work based on situation recognition," *Proc. of the 8th IEEE International Symposium on Multimedia*, pp. 851–860, Dec. 2006.

[124] J. Wang, C. Xu, E. S. Chng, L. Duan, K. Wan, and Q. Tian, "Automatic generation of personalized music sports video," *Proc of ACM MultiMedia'05*, pp. 31–38, 2005.

[125] J. Wang, E. S. Chng, C. Xu, H. Lu, and Q. Tian, "Generation of personalized music sports video using multimodal cues," *to appear on IEEE Trans. on MultiMedia*.

[126] N. Nitta, N. Babaguchi, and T. Kitahashi, "Generating semantic descriptions of broadcasted sports video based on structure of sports game," *Multimedia Tools and Applications*, vol. 25, no. 1, pp. 59–83, 2005.

[127] "http://www.thefa.com/euro2004/newsandfeatures/postings/2004/07/minbymin_portgreecefinal.htm,"

[128] dtSearch Corp, "dtsearch 6.50 (6608)," 1991-2005.

[129] A. Amir, M. Berg, G. Iyengar, C.-Y. Lin, C. Dorai, M. Naphade, A. Natsev, C. Neti, H. Nock, I. Sachdev, J. Smith, Y. Wu, B. Tseng, and D. Zhang, "Ibm research trecvid-2003 video retrieval system," *Proc. of NIST TRECVID'03*, 2003.

[130] A. Amir, "The ibm shot boundary detection system at trecvid 2003," *Proc. of NIST TRECVID 2003*, 2003.

[131] J. Wang, E. S. Chng, and C. Xu, "Soccer replay detection using scene transition structure analysis," *Proc. of IEEE ICASSP'05*, 2005.

[132] M. Pilu, "On using raw mpeg motion vectors to determine global camera motion," *Proc. SPIE VCIP'98*, vol. 3309, pp. 448–459, 1998.

[133] M. Bertini, R. Cucchiara, A. D. Bimbo, and A. Prati, "Object andevent detection for semantic annotation and transcoding," *Proc. of IEEE ICME'03*, pp. 421–424, 2003.

[134] "http://www.i2r.a-star.edu.sg,"

[135] S. Takagi, S. Hattori, K. Yokoyama, A. Kodate, and H. Tominaga, "Sports video categorizing method using camera motion parameters," *in Proc. of ICME'2003*, July 2003.

[136] J. Wang, C. Xu, E. S. Chng, and Q. Tian, "Sports highlight detection from keyword sequences using hmm," *Proc. of IEEE ICME'04*, 2004.

[137] V. Petrushin, "Hidden markov models:fundamentals and applications," *Online Symposium for Electronics Engineer*, 2000.

[138] "Hidden markov model toolkit," *http://htk.eng.cam.ac.uk/*.

[139] J. Parker, "Algorithms for image processing and computer vision," *New York:John Wiley and Sons, Inc.*, pp. 23–29, 1997.

[140] P. Hough, "Method and means for recognizing complex patterns," *US Patent 3069654*, 1962.

[141] International Football Association Board, "Law of The Game," *Federation International de Football Association, 11 hitzigweg, 8030 Zurich, Switzerland*, 2003.

[142] G. Xu, Y. Ma, H. Zhang, and S. Yang, "An hmm-based framework for video semantic analysis," *IEEE Trans. on Circuits and Systems for Video Technology (CSVT)*, pp. 1422–1433, Nov. 2005.

[143] X. Gibert, H. Li, and D. Doermann, "Sports video classification using hmms," *Proc. of ICME'2003*, 2003.

[144] T. Hayden, "Empowering sports fans with technology," *Computer*, pp. 106–107, Sept. 2004.

[145] J. Chin, V. Diehl, and K. Norman, "Development of an instrument measuring user satisfaction of the human-computer interface," *Proc. of SIGCHI on Human Factors in Computing System*, pp. 213–218, 1998.

[146] "http://www.ntu.edu.sg/home5/y020002/research/broadcasting/test.htm,"

[147] M. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images:a survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, pp. 34–58, 2002.

[148] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition:a literature survey," *ACM Computing Surveys*, vol. 35, pp. 399–458, 2003.

[149] D. M. Gavrila, "The visual analysis of human movement:a survey," *Computer Vision and Image Understanding (CVIU)*, vol. 73, no. 1, pp. 82–98, 1999.

[150] N. Maddage, C. Xu, M. Kankanhalli, and X. Shao, "Content-based music structure analysis with applications to music semantics understanding," *Proc. of ACM MultiMedia'04*, pp. 112–119, 2004.

[151] E. Ukkonen, "On approximate string matching," *Proc. of Data Compression Conference (DCC'93)*, pp. 148–157, 1993.

[152] G. Wijnalda, S. Pauws, F. Vignoli, and H. Stuckenschmidt, "A personalized music system for motivation in sport performance," *IEEE Pervasive Computing*, pp. 26–32, July-Sept. 2005.