國立臺灣大學資訊工程學研究所博士論文

指導教授：吳家麟博士

具語意基礎之電影與運動影片內容分析及組織

Semantics-based Content Analysis and Organization in Movies and Sports Videos

研究生：朱威達 撰

學號：D91922016

中華民國九十五年六月

致謝

經歷了許多考驗，我終於完成博士論文，在人生與研究的道路上更前進一步。這幾年在台大的學習過程中，我不僅學到研究的方法與技能，更深深體會到環境的影響、老師的身教言教、以及同儕之間競合形成的力量。在這個過程當中，我接收到許多鼓勵與指導，也承蒙許多學弟妹的配合，讓我順利完成研究目標。

我首先最感謝的是敬愛的吳家麟教授，其豐富的學養與對學術追求的認真態度讓我在學習期間獲益良多。吳教授除了給予我許多空間得以選擇喜歡的研究主題之外，亦常在關鍵時刻指引研究方向。吳教授的關心與指導是我完成博士論文的主要支持。

其次，我非常感謝論文口試委員對本研究的寶貴意見，他們包括黃肇雄教授、洪一平教授、李琳山教授、許永真教授、郭大維教授、李素瑛教授、杭學鳴教授、陳良弼教授、鍾國亮教授、許聞廉教授、與陳銘憲教授。委員們的建議讓我的論文更完整，也讓本研究更經得起考驗。

我還要感謝通訊與多媒體實驗室中許多老師的鼓勵，他們在實驗室事務上或生活上都給予我相當大的支持與肯定，其中包括歐陽明教授、陳文進教授、周承復教授、陳炳宇教授、以及莊永裕教授。

除此之外，我也要感謝暨南國際大學陳恆佑教授，他帶我初探學術研究之堂奧，也引起我進一步唸博士班的興趣。感謝李家同教授，他指導我學好語文，也在研究的道路上給予我支持。

與實驗室同學朝夕相處共同學習是另一股讓我前進的力量。承蒙許多學長姐的指導與學弟妹的支持，我獲得許多無價的生活體驗。感謝童怡新博士與何嘉強博士讓我見識到銅牆鐵壁般的強者風範；感謝郭晉豪博士在生活上與團隊合作上給予的協助；感謝黃俊翔博士中肯又切中要點的研究指導；感謝黃奕勤博士、莊玉如博士、與林佳緯博士的關懷與鼓勵；感謝葉正聖博士時時都能給予協助的超人作為。

感謝與我共同進行研究的文皇、振修、人豪、萱瑋、致豪、宇皓、駿丞、家偉、昇舫、仲毅、俊彥等。與你們的合作讓我學到很多，也感謝你們的配合。感謝實驗室其他同學的鼓勵與支持，包括育慈學姐、頌文、致仁等博士班同學，廷

建、至豪、錦昕、盛禾、佳盈、義欽、雅婷、奇豪、俊偉、聖凱、嘉豪等碩士班同學等。

感謝新嘉、詩綺、婉鈺、獻良在我到台北求學這段期間時不時的熱情邀約，讓我的生活更加豐富。

最後我非常感謝我的父母與家人，你們的支持讓我無顧之憂得以全心學習。多年來我在外求學少有歸鄉，你們對我的包容與鼓勵是最大的後盾。

本論文相關成果受數個國科會計畫贊助，包括台灣大學卓越延續計畫「多媒體生活環境的數位內容科學」(NSC94-2752-E-002-006-PAE)、產學合作計畫「經驗融合：兼具安全性與延展性之多媒體人本計算」(NSC94-2622-E-002-024, NSC93-2622-E-002-033)與「媒體內容工程：MPEG-4/7 相關技術之研發」(NSC91-2622-E-002-002)。

朱威達 謹誌

九十五年六月

Abstract

Conducting content analysis approaching semantics level is an emerging trend in multimedia researches. Such kind of analysis matches users' needs and facilitates content management and utilization in a more effective and reasonable way. Unlike conventional content-based retrieval or indexing, works on semantics analysis integrate techniques of statistical pattern recognition and machine learning with specific production rules or domain knowledge to bridge the semantic gap between low-level features and high-level semantics.

On the basis of machine learning and pattern recognition technologies, systems that combine analytical results from different classifiers, different features, or different modalities are developed. In this dissertation, we propose a general framework that introduces the idea of mid-level representation between audiovisual features and semantic concepts. Two types of techniques, i.e. statistical pattern recognition and rule-based decision, are combined to facilitate narrowing the semantic gap.

We develop three systems that respectively conduct semantic concept detection in action movies, in broadcasting baseball games, and in sports videos. In action movies, we detect semantic concepts, such as gunplay and car-chasing scenes, through analyzing aural information. Statistical approaches are exploited to characterize concept modeling and to facilitate mapping between different semantic granularities. In baseball games, visual and speech information are combined, and a hybrid method that includes rule-based and statistical techniques is designed for semantic concept detection. Thirteen semantic concepts, such as single, double, homerun, and strikeout, are explicitly detected, and several realistic applications can therefore be built. In general sports videos, we extract the ball trajectory to be a new type of metadata for describing content characteristics. Some novel semantic concepts, such as pitch types in baseball games, can therefore be modeled and detected. These studies are the instances of the proposed general framework and demonstrate the realization of automatic semantic concept detection.

中文摘要

將內容分析技術推向語意層級是近年來在多媒體領域中急速發展的研究課題。此類技術的分析結果較能符合使用者的需求，也讓內容管理與應用變得更加有效率。有別於傳統以內容為基礎的檢索技術，數位內容語意分析結合圖型識別、機器學習的技術與特定製作原則、領域知識來彌合低階特徵值與高階語意之間的鴻溝。

基於機器學習與圖型識別的技術，已有許多系統結合不同分類器、不同特徵值、或不同媒體型態的結果來進行語意分析。在本論文中，我們提出一個通用的架構來進行此類研究。其中，我們引入介於視聽特徵值與語意概念之間的中介資訊來輔助分析。

我們發展了三個不同的系統，在電影、棒球影片、以及一般的運動影片中進行語意概念偵測。在動作電影中，我們透過聲音的資訊來偵測槍戰與飛車追逐等語意概念。我們採用統計方法來描述概念以及對應不同層次的語意。在棒球比賽中，我們基於畫面與語音的資訊，結合了以規則為基礎與以模型為基礎的方法來做語意概念偵測。總計有十三種不同的概念，如一壘安打、二壘安打、全壘打、三振等，可被偵測出來，也藉此我們可發展許多實際的應用。在一般的運動影片中，我們提出可用球的軌跡來輔助內容分析。一些新型態的語意概念，如棒球比賽中投手的球種，可因此被描述與偵測出來。這三大類研究都是基於我們所提的通用架構，也因此證實了此架構對於語意概念偵測的實用性。

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

Large amounts of digital content have been created, stored, and disseminated as a result of the rapid advances in media creation, storage, and compression technologies. Massive data present challenges to users in content browsing and retrieval, thereby diminishing the benefits brought by digital media. Although various content creation and utilization devices/methods are available for many splendid applications, tremendous and disordered multimedia content impede information access and usage.

The information access problem has arisen for many years since large volume of digital data can be stored in disks. In the last decade, the emergence of internet even aggravates this problem because more data can be easily shared and different variations of media, such as image, audio, and video, are easily created to convey much complex information. Recently, text-based information indexing and retrieval have been well solved, and many search engines like Google or Yahoo! are popularly used around the world. However, multimedia information indexing is still an open issue that poses urgent needs in either industry or research communities.

In this dissertation, we investigate how to develop systematic approaches on multimedia content analysis and adaptation to solve this thorny problem, as shown in Figure 1-1. Based on elaborate analysis techniques, we can provide better tools for media management, access, and utilization, with the functionalities that better match users' needs.

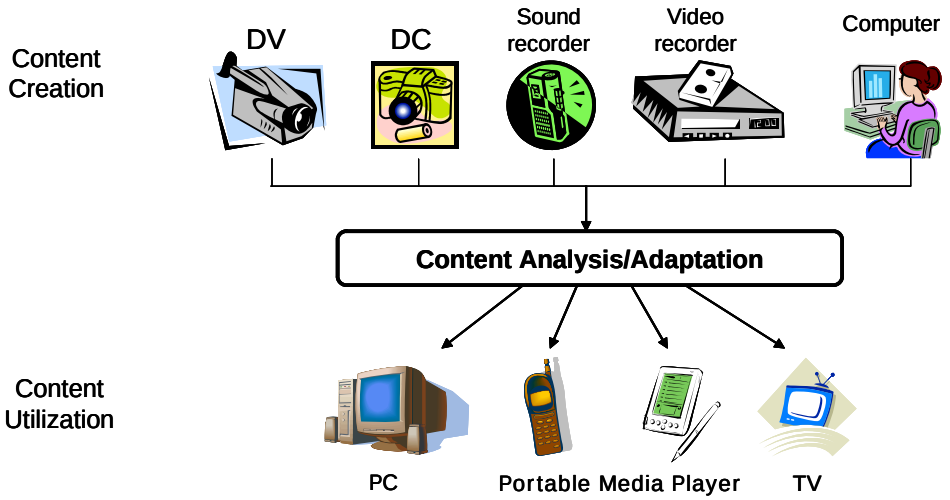


Figure 1-1. Content analysis or adaptation techniques facilitate efficient access and management in heterogeneous content creation and utilization environments.

1.2 Related Works

The existing content analysis techniques can be categorized in several ways, including categorizing by modality, by level of analysis, and by processing methods. Extensive survey on the categorization can be seen in [Naph01]. We generally review related techniques in this section, and more specific surveys will be described in the corresponding chapters.

1.2.1 Categorize by Modality

We can categorize existing techniques by modality, including image, audio, and video analyses.

- Image analysis

Since the prominent QBIC (Query by Image Content) system [Flic95] was developed, content-based image retrieval attracts much attention and brings about lots of research issues. Most systems provide query by example/sketch functionality, and some variations, such as relevance feedback and object-based query, were also proposed to enhance the performance of image retrieval. Examples of image retrieval systems include QBIC [Flic95], Virage image search engine [Bach96], VisualSeek [Smit96], and MARS [Rui98]. Extensive survey can be found in [Smeu00] and [Rui99].

- Audio analysis

Although speech processing/recognition technologies have been studied for a long

time, little progress has been made for nonspeech audio data. Recently, classification and segmentation techniques [Lu02][Zhan98] were proposed to discriminate different types of audio, such as speech, music, noise, and silence. They either perform audio analysis based on trainable models [Lu02] or based on heuristics [Zhan98]. As the increasing needs of music classification and sharing, additional works have been conducted on music genre classification [Tzan02], music snippets [Lu03], audio thumbnailing [Bart05], and structure detection [Madd06].

- Video analysis

Based on the studies of still image analysis, video data analysis that additionally exploits temporal characteristics have also actively advanced in recent years, especially high-quality videos are largely created and disseminated on the internet. The Informedia project [Wact96] presents video analysis systems that pioneer several topics on visual stream segmentation and audio content classification. From the viewpoint of video hierarchy, shot boundary detection algorithms [Yeo95][Hanj02] were developed to segment video clips into shots, each of which presents visual continuity. The keyframes of each shot are then selected to summarize a video clip and are applied to video abstraction [Li01][Pfei96] and content-based retrieval [Dimi02]. On the other hand, techniques for genre classification are also widely studied. Genres of films [Monc03] and TV programs [Liu98] are automatically classified. Various features from audio, video, and text [Wang00] are exploited, and multimodal approaches are proposed to cope with the access and retrieval issues of multimedia content.

1.2.2 Categorize by Level of Analysis

Existing content analysis techniques can be categorized by the level of analysis: analyzing the syntactic structure of media or analyzing the hidden semantics in media. One of the typical examples in former techniques is to parse a video into hierarchical structures, such as scene, shot, and frames [Zhan95]. Another example on image retrieval is to use color, texture, or edges to be the bases of image matching [Smeu00].

The aforementioned studies investigate the syntactic structure of targeted media and facilitate efficient browsing. However, similarity in the low-level feature space is yet far from that in the conceptual space, in which users think. Processing at higher level, i.e. semantics, is therefore the ultimate goal of content analysis techniques. Some examples of recent attempts to semantic content analysis include semantic video indexing [Naph98][Naph02][Chu04][Chu05-3], semantic visual templates generation [Chan98], and many works on sports videos analyses, such as [Chu05-4], [Ekin03], and [Xie04].

Although bridging the semantic gap is yet a very challenging problem, integrating techniques from other disciplines like pattern recognition, machine learning, and computer vision realizes the goal. Developing and advancing semantic content analysis techniques are accordingly the targets of this dissertation.

1.2.3 Categorize by Processing Methods

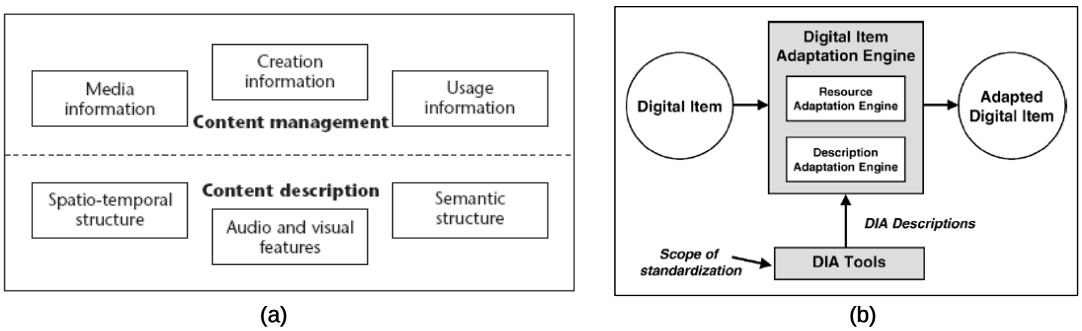
As mentioned in the previous subsection, techniques from different fields are cooperated to achieve better content analysis. In addition, domain knowledge or production rules often significantly help in specific domain media analysis. Rule-based methods or analyzing with heuristics have been proposed for different kinds of media. For example, Zhang and Kuo [Zhan98] proposed a content-based audio classification and retrieval system based on heuristics. For broadcasting baseball videos, Liang et al. [Lian05] exploit official baseball rules to explicitly detect semantic baseball events, such as strikeout, homerun, and double play.

On the other hand, many other types of media don't possess fixed rules or clear domain knowledge, such as movies and home videos. In these cases, researchers appeal to machine learning or statistical pattern recognition techniques to model the hidden characteristics of media. Naphade and Huang [Naph98] [Naph02] propose a probabilistic framework to detect events in generic videos. Haering et al. [Haer00] develop a neural network architecture to detect events in wildlife videos. Extensive survey on statistical pattern recognition used in content analysis can be seen in [Jain00].

Because different media have significantly different characteristics, we usually have to make a choice between rule-based methods or learning-based methods. Moreover, we may have better results by fusing them [Chu05-4]. Collaboration of different methods is also the targeted issue of this dissertation.

1.2.4 Concerns from International Standards

The importance of multimedia content analysis is also reflected in the context of international standards. In MPEG-7, content description and management description tools are defined to describe metadata of multimedia content [Chan01][Mart02-1] [Mart02-2], as shown in Figure 1-2(a). Content description tools present perceptible information, including spatio-temporal structure (e.g. video hierarchy), audio and video features, and semantic description (e.g. objects, events, and relationships). Content management tools let us specify information about media features (e.g. coding format), creation, and usage of multimedia content (e.g. rights and availability) [Mart02-1].



(a) (b)

Figure 1-2. (a) Content description and management description tools in MPEG-7, and (b) digital item adaptation in MPEG-21 (Part7)

In MPEG-21 [Burn03], extended framework has been proposed to perform digital item adaptation [Vetr05]. For universal multimedia access, a digital item that consists of content itself and corresponding description is adapted to different variations, in order to match the requirements or limitations of different usage environments. Figure 1-2(b) shows the idea. Universal and transparent media usage is pursued in this ambitious multimedia framework.

Although some realistic works have been proposed based on the ideas of MPEG-7 and MPEG-21 [Tsen04], how to implement or extract metadata of multimedia content is out of the scope of these standards. They give us the vision of multimedia content analysis and leave the research issues for pursuing. In this dissertation, we would propose several approaches to make the ultimate goal come true.

The aforementioned studies have been conducted on content analysis for efficient indexing, browsing, retrieval, access, and management, but problems do exist in today's applications. The first is the apparent gap between low-level audiovisual features and high-level semantics. Similarities in low-level features do not always match users' perception. The second problem is that, from the viewpoint of end users, scenes/shots are associated due to semantic meaning rather than color layouts or other computational features. Therefore, an approach that bridges the *semantic gap* in a systematic way is urgent to be developed.

In our work, we try to develop systematic approaches to tackle with automatic video indexing, in the sense of semantics rather than low-level features. By exploiting statistical pattern recognition or specific domain knowledge, we narrow the gap between computational audiovisual features and high-level semantic concepts. This work would be the foundation of advanced multimedia information retrieval and digital content/asset management.

1.3 Semantic Concept Detection

To clarify and position the proposed work, we first describe the definitions of *feature*, *event/objects*, *concept*, and *knowledge* and introduce the idea of semantic concept detection. Then, conventional pattern recognition techniques are briefly reviewed and are compared with the techniques in semantic concept detection.

1.3.1 From Feature to Knowledge

The fundamental processes of video indexing are data classification, clustering, or recognition. According to the domain of processing, we distinguish video indexing into several levels, which ranges from audiovisual features that represent data characteristics to knowledge that is conveyed by the implicit relationships between concepts.

- Audiovisual Features

Features are the measurable properties of the phenomena being observed. They can be directly computed from the perspectives of time or frequency domain, aural or visual modality, and statistical or singular characteristics. In the last few decades, most researches proceed content analysis in the domain of features. Some commonly used features include color, edge, and motion features in video sequences, and energy, zero-crossing rate, and mel-frequency cepstral coefficients (MFCC's) in audio streams [Wang00]. Many applications have been developed on the basis of feature matching, clustering, and modeling. One of the typical examples is content-based image retrieval [Smeu00][Flic95].

- Events and Objects

Events and objects are entities that take place or exist in time and space in the world [Beni05]. Events often come up with a specific evolution of features and last for a duration. For example, in the radio broadcasting situation where speech and music would occur alternately, the trend of low zero-crossing rate and low mean of spectrum flux emerges when music starts [Chu05-6]. Objects are often discussed in the domains of still images or video sequences without shot change. They can be living entities such as people, animals, and plants; man-made objects such as vehicles, buildings, and furniture; and natural objects such as mountains, rivers, and stars.

- Semantic Concepts

Concepts are the notion or meaning of world entities. Semantic concepts refer to entities named with words in a language [Beni05]. A concrete semantic concept may

be involved with specific evolutionary patterns or interrelationships between various events and objects. For example, a gunplay scene (semantic concept level) is often constituted by frequent gunshots and explosion audio effects (event level) and poses a compact representation of a meaning. To identify the existence of a concept, context of events and objects should often be examined.

- Knowledge

Knowledge is usually defined as facts about the world and represented as concepts and relationships among the concepts [Beni05]. It would not be a visible or perceptible entity in multimedia content, but an implicit relation among concepts or the convention in a specific domain. One example of knowledge in movie making is how to arrange the layout, lighting, and scene editing so that the movie segment can arouse a specific perception.

Figure 1-3 shows these four levels of content representation. Conventionally, we view the representation of features as the lowest level. Features can be processed efficiently and automatically, but the results based on them are still far from human's cognition. Concepts and knowledge are viewed as higher levels of processing because they match more appropriately with human's cognition or perception. Events or objects are viewed as the intermediate information between features and concepts. In our work, we try to go beyond the scope of conventional content-based (feature-based) content analysis and propose a systematic framework to tackle the issues at event and concept levels. With the aid of reliable event and concept detection, it's possible to automatically discover knowledge implicitly hidden in content.

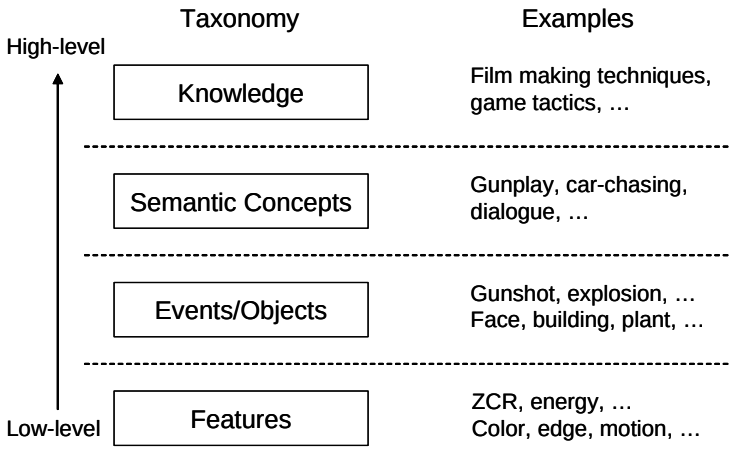


Figure 1-3. From features to knowledge.

1.3.2 Pattern Recognition vs. Semantic Concept Detection

The goal of our work is to perform semantic concept detection in a systematic and effective way. This kind of problem can be formulated as: given digital content that may be in the form of audio or video, identify what events/objects are present or what concepts can be inferred. This task is conceptually similar to pattern recognition problems. Figure 1-4 shows a conventional pattern recognition framework. After feature extraction, techniques such as template matching, statistical classification, syntactic/structural matching, and neural network can be applied based on the statistical characteristics of features [Jain00]. On the other hand, if the targeted events or objects possess explicit patterns and clear production rules, we can achieve effective results through rule-based decision/detection methods. This paradigm has been widely adopted for a few decades, and various approaches focusing on feature extraction, feature selection, statistical modeling, and event/object detection have been devised. However, such kind of event/object detection is not sufficient to achieve efficient and effective multimedia content management and retrieval. Higher level of processing like semantic concept detection should be involved in developing a content management system that really matches human's needs.

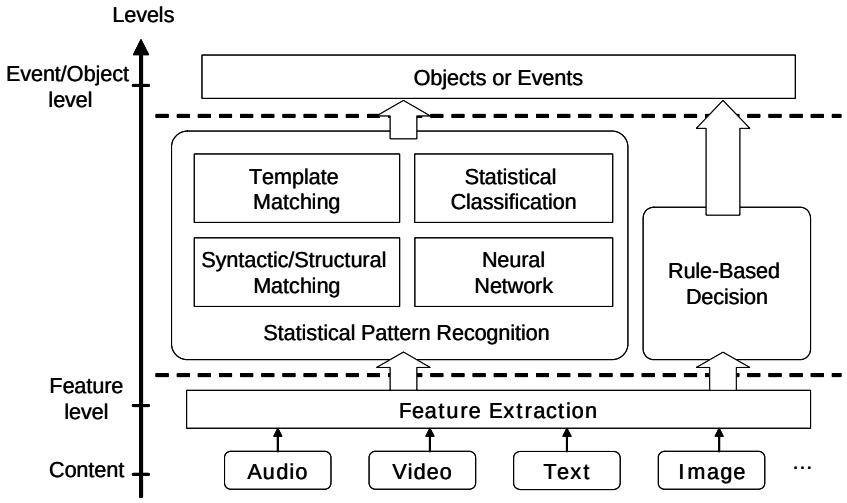


Figure 1-4. A conventional pattern recognition framework.

Recently, the idea of combining the results of event detection to achieve higher level analysis has attracted more and more attention. Duan et al. [Duan03] proposed a mid-level framework that integrates the results of event/object detection to detect semantic concepts in soccer games. Naphade and Huang [Naph02] proposed a dynamic Bayesian network to characterize various detected objects and infer semantic concepts from a probabilistic view.

From these works, we conclude these approaches from the viewpoints of pattern recognition and illustrate the essential framework in Figure 1-5. The philosophy of

this kind of work is to combine the results of individual classifiers and take their outputs to train a meta-classifier or make an integrated decision. Information from different modalities or different classification methods can be integrated to boost the performance of detection or provide extension for other unknown concepts. Actually, this idea has been studied for years in the community of pattern recognition and is justified to be beneficial to efficiency and accuracy in some cases [Kitt98]. This multi-level classification/detection can be applied to model the relationships that are not explicit in the low-level feature space and is viewed as an effective approach to narrow the semantic gap.

The semantic gap, as shown in Figure 1-5, is derived from the mismatch between feature similarity and human cognition. However, in well produced and edited videos such as movies, news, and sports videos, production rules or domain knowledge are often used to present concepts [Dora02][Zett99][Chen04]. The layout of objects or the relationship of events affect human's perception or present specific concepts. On the other hand, object and event detection have been widely studied and various classification and recognition methods have been proposed. Through connecting these two relationships, automating detection of semantic concept will be possible. The construction details of the semantic concept detection framework will be discussed in Chapter 2.

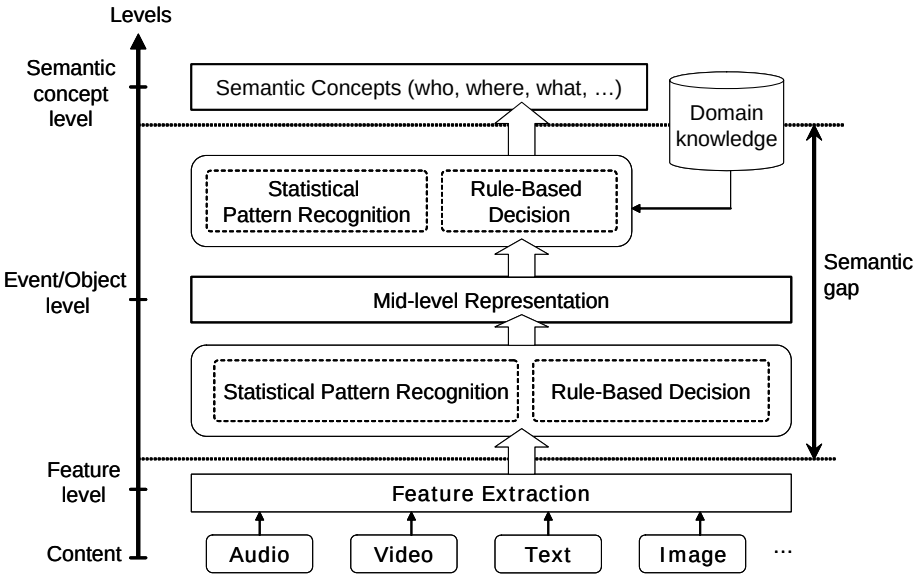


Figure 1-5. The concluded semantic concept detection framework.

1.4 Problem Statement

In this dissertation, we perform semantic concept modeling/detection/indexing for multimedia content. This task can be described as:

Given multimedia content, develop techniques that bridge the gap between features and semantic concepts such that content management and utilization can be proceeded as human does.

To facilitate efficient and effective semantic concept processing, techniques for this goal should be provided with the following characteristics:

1) Classification or detection results from different classifiers (with different features, different classifier parameters, or different datasets) should be seamlessly combined to achieve more reliable and/or accurate performance [Kitt98].
2) Specific production rules [Dora02][Zett99], domain knowledge, and probabilistic models for semantic inference should be cooperated to deal with different types of media and facilitate various applications.
3) Both the concepts in temporal (evolution along with time) and spatial (coexistence in the same space) aspects should be considered and modeled.

1.5 Summary of Contributions

In this dissertation, we investigate multimedia semantic analysis with respect to semantic concept detection and content adaptation. These works are summarized as follows:

1.5.1 Audio Semantic Concept Detection in Movies

On the basis of audio information, we try to detect specific semantic concepts in action movies [Chu05-2, Chu05-3, Chu05-5, Chu04, Chen03]. We propose a two-level framework to narrow the semantic gap: audio event modeling and semantic concept modeling. Four audio events, including gunshot, explosion, engine, and car-braking sounds, are modeled by probabilistic models (say, the hidden Markov model) in the first level. Then, the outputs of event models are concatenated as vectors to describe the interrelationship between different events and characterize two semantic concepts, i.e. gunplay and car-chasing scenes. Context of audio events are taken into account. Therefore, this work is specially called as *semantic context detection* because the *context of semantic concepts* is modeled.

1.5.2 Explicit Baseball Concept Detection

In this work, we propose a hybrid method that exploits rule-based decision and model-based decision to explicitly detect what happened in baseball games [Chu05-1, Chu06-1, Chu05-4, Lian05, Lian04]. Thirteen concepts, including single, double, homerun, strikeout, etc. are detected according to the audiovisual information in various broadcasting games. With the help of key-phrase spotting [Chen98] from speech information, we further develop a fusion scheme to elaborate event detection. The results of detection are applied to automatic box score generation, game summarization, and highlight extraction. With the explicitness of concept detection, realistic applications that match users' needs can be built.

1.5.3 Trajectory-Based Analysis in Baseball Videos

To enrich the viewing experience of baseball games and provide some clues for enhancing pitcher's performance, we propose a Kalman filter-based approach to track ball trajectory from single-view pitching sequences [Chu06-2]. Without setting extraordinary equipments in stadiums or other sensing instruments, this approach robustly extracts ball trajectory for pitching sequences captured from TV channels or downloaded from the Internet. To validate the detected ball trajectories, we investigate the characteristics of ball trajectories on the basis of a baseball physical model. The effectiveness of ball trajectory extraction and ball position detection has been shown. Moreover, based on the extracted trajectory, the pitching type can be automatically recognized, and therefore, a new type of metadata that is never provided in previous researches can be generated.

Rapid advance of content analysis techniques stimulates the development of attractive applications that are across the boundaries of different academic disciplines. The results of content analysis not only facilitate information management but also provide new thoughts for multimedia communications, such as intelligent content adaptation and semantic-based QoS applications. We discuss how content analysis aids in content repurposing for various clients, which have different device capabilities and content requirements. In the discussion section, we bring up some ideas to inject the impacts of content analysis into multimedia communication applications.

1.6 Dissertation Organization

This dissertation contains lots of multimedia content analysis studies from different perspectives. We propose a generic framework that analogizes semantic analysis to language learning in Chapter 2. All other contexts in this dissertation are centralized by this general idea. Firstly, semantic concept detection from audio information is described in Chapter 3. Based on probabilistic modeling, audio events are modeled and detected. The relationships between audio events are further modeled to characterize audio semantic concepts. In Chapter 4, we describe the details of explicit concept detection in broadcasting videos. Collaboration of visual and speech information is comprehensively described, and several realistic applications demonstrate the effectiveness of the proposed methods. On the other hand, we present trajectory extraction and pitching type recognition for baseball videos in Chapter 5. The new type of metadata emerges with a cheap and efficient solution. We describe some discussions on cooperating content analysis and content adaptation with other fields in Chapter 6. Moreover, the original contributions of this dissertation and directions for future researches are also addressed.

In Appendix A, we briefly review the theoretical foundations of hidden Markov model (HMM), which is used in Chapter 3 for modeling audio events and semantic concepts. The probabilistic modeling and training techniques are described. In Appendix B, we describe the idea and training techniques of support vector machine (SVM), which is a discriminative approach to modeling semantic concepts. Appendix C states the computational media aesthetics, which describes and relationship between audiovisual elements and filmmaking.

Chapter 2
A Unified Framework for Multimedia Semantic Analysis

2.1 Content Analysis and Concept Language

To approach multimedia semantic analysis, we first sketch how human beings learn a concept from language. In linguistics, a "grammatical sentence" that presents a complete statement is a string of symbols that conforms to the syntactic rules (or the so-called language grammar) [From97]. We know the meaning of a sentence because of the constituted words and their syntactic relationship. Similarly, but at different levels, a word is pronounced by the constituted phonemes according to phonological rules. Figure 2-1(a) illustrates the multi-granularity relationship in languages. In the last few decades, the linguistic domain knowledge and statistical modeling techniques have facilitated the development of speech recognition systems. The syntactic rules of constituting a sentence are conveyed in statistical language models, and the phonological rules of each isolated word are learnt as acoustic models [Huan01], as shown in Figure 2-1(b). The works on speech recognition demonstrate the feasibility of mapping different granularities of information in systematic ways, with the helps of statistical learning techniques or specific syntactic rules.

Multimedia semantic analysis, apparently, covers large amount of research topics. To simplify the descriptions in the following sections, we use semantic concept detection as the example to portray the ideas. We analogize a semantic concept to a sentence, which is constructed by some mid-level representations, such as specific audio/video elements. Like syntactic rules or language models, audio and video elements can be elaborately arranged to construct specific semantics. Film grammar [Arij91] or the media aesthetics [Zett99] are, therefore, widely applied in making movies and TV programs. Likewise, the audio/video elements are modeled based on the characteristics of low-level features, which are analogous to phonemes in language, as shown in Figure 2-1(c). The idea of mapping semantic concept detection to linguistic structure or speech recognition process demonstrates that we tackle with multimedia semantic analysis in a multi-granularity way.

The common characteristics of these examples are that different semantic

granularities are defined. Some rules or statistical methods are applied to map one semantic level to another, whereas the mapping functions vary in different tasks. As different levels of presentations are linked as a chain, we propose a "content chain framework" to deal with multimedia semantic analysis.

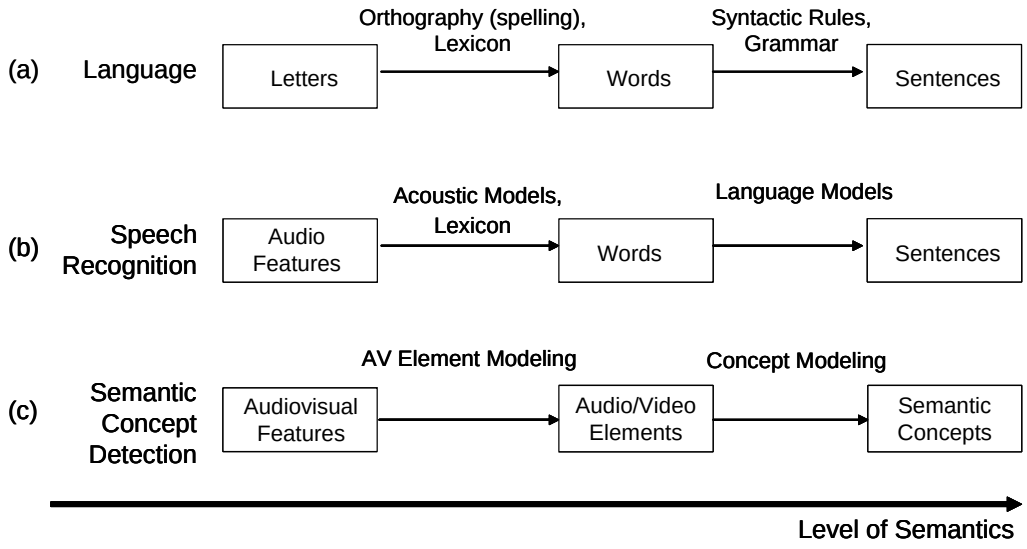


Figure 2-1. Analogies between language, speech recognition, and semantic concept detection.

2.2 Content Chain Framework

2.2.1 Framework Overview

Multimedia semantic analysis can be proceeded from different perspectives, at different levels, or by different methods according to the targeted applications. Conceptually, we often face this problem in many fields: given some types of observations, find the symbol that most likely presents according to the characteristics of observations. As illustrated in the 2-level chain in Figure 2-2, the nodes denote set of entities in each level, such as phonemes at the lower level and words at the higher level. The edge between two levels denotes a generative function f, which maps lower-level entities to higher-level ones. We take the isolated word recognition as an example and formulize the generative function as follows:

Given a word domain $W=\{w_1,w_2,\ldots,w_N\}$, where N is the number of words in a specific application. Denote $P=\{p_1,p_2,\ldots,p_n\}$ as the set of observed phonemes and $R=\{r_1,r_2,\ldots,r_m\}$ as the relations between the elements in P. A word w_i can be presented as $w_i=f(P_i,R_i)$, where P_i (R_i) is a set of elements in P (R), and f is the generative function that generates the word w_i by giving the phonemes P_i and corresponding relationship R_i. For example, the function f can be a phonetic

dictionary that maps the phoneme sequence "B UH K" to the word "book" [CMU06]. On the other hand, in the automatic isolated word recognition paradigm, the function f is elaborately implemented by the hidden Markov model [Rabi89], which maps audio features to words on the basis of statistical characteristics.

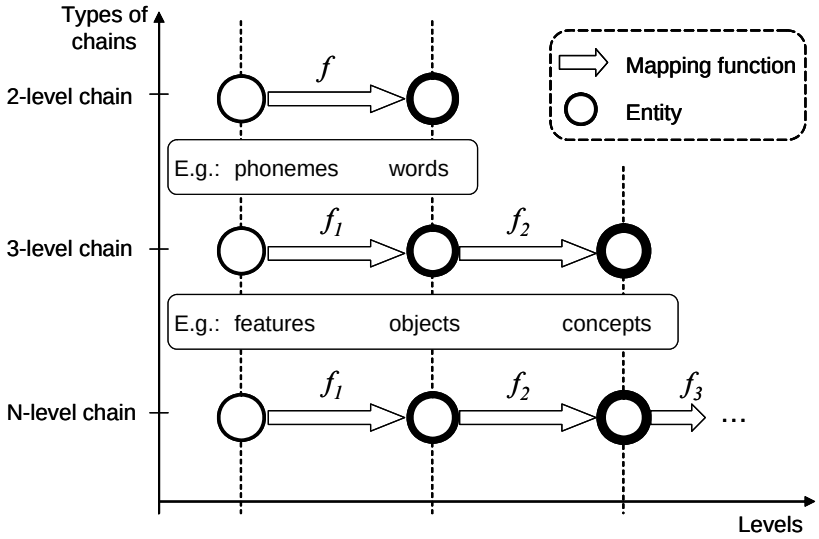


Figure 2-2. Illustrations of different levels of content chains.

The aforementioned paradigm is only involved in two levels. However, not every problem can be simplified as two levels. The semantic concept detection, for example, is one of the complex problems that should be further divided. In semantic concept detection, we can hardly develop a generative function that directly maps audio/video features, such as motion and color, to semantic concepts, such as homerun in baseball games and gunplay scenes in action movies. That's the notorious "semantic gap problem." For this problem, some literature has mentioned that mid-level representations [Duan03] could be introduced to bridge the gap. As shown in the 3-level chain of Figure 2-2, the process of semantic concept detection steps from audio/video features to objects, and then steps to concepts. This paradigm shows that the process of semantic content analysis behaves like the extension of conventional pattern recognition problems.

For a complex problem, it may be a feasible manner to divide it into finer sub-problems and conquer them sequentially, like the N-level chain in Figure 2-2. The generative function(s) between two levels should be carried out in accordance with the characteristics of two ends. For semantic concept detection formulized as a 3-level chain, the generative functions can be described as follows [Lay06]:

Given a concept domain $C=\{c_1,c_2,\ldots,c_N\}$, where N is the number of concepts in the targeted domain. A concept c_i is represented as $c_i=f_2(O_i,O_i^R)$, where $O=\{o_1,o_2,\ldots,o_n\}$

are the mid-level representations, with the corresponding relationship O_i^R. Similarly, but at a different level, a mid-level element $o_j=f_1(V_j,V_i^R)$, where $V=\{v_1,v_2,\ldots,v_m\}$ are the audio/video features, with the corresponding relationship V_i^R.

From the aforementioned examples, we figure out that the generative function can be determinately built like a dictionary or a set of rules, or can be nondeterminately built by using statistical learning techniques. To combat the targeted problem, we should take different types of generative functions. According to the determination characteristics, we discuss the choices of using generative functions in two phases: deterministic mapping function and nondeterministic mapping function.

2.2.2 Deterministic Mapping Function

For those problems which present clear and definite relationships between essential elements, we are able to construct the mapping function by exhaustively listing the generative rules or heuristically defining some thresholds for decision making. One example of deterministic mapping functions is rule-based baseball concept detection module [Chu05-4]. A "double" concept, for example, can be characterized as the batter reaching the second base while nobody is out. Because baseball rules are well defined and have tight relationships with baseball concepts, we can get promising results in applying them as the mapping function. Other examples like audio classification [Zhan98] can also be heuristically performed by using deterministic methods.

2.2.3 Nondeterministic Mapping Function

For those problems which cannot be resolved by definitely exploiting rules or no clear definition on elements, we can appeal to statistical pattern recognition techniques. Because we don't exactly know the generative rules of a specific entity, supervised or unsupervised learning approaches are adopted to statistically characterize the relationship between observed features and the targeted entity. For example, in speech recognition, given a sequence of observed feature vectors, $O=o_1,o_2,\ldots,o_T$, the probability of a word w is $P(w|O)$, which is *a posteriori* probability calculated from the hidden Markov modeling. Analogous to this formulation, Bach et al. [Bach05] propose a statistical modeling method for baseball concept detection.

2.2.4 Generality of the Content Chain Framework

Works on multimedia semantic analysis are engaged in developing a specific detection or classification technique to bridge two semantic granularities. Deterministic methods, such as rule-based detection, or nondeterministic methods, such as statistical learning techniques, are widely studied. However, it's often the case

that different levels of sub-problems possess significantly different characteristics. Simply using one type of method cannot succeed.

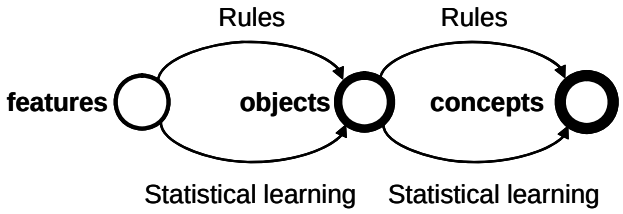


Figure 2-3. Implementations of generative functions.

In the proposed chain framework, we suggest that any mapping function between two semantic granularities can be implemented by deterministic methods, nondeterministic methods, or hybrids of them. Therefore, in the semantic concept detection example shown in Figure 2-3, there are at least four combinations of mapping methods. Different combinations are selected according to targeted granularity and the characteristics of content. For example, in baseball concept detection paradigm, concepts like “strikeout” and “double play” are detected by checking the caption information changes according to baseball rules. We can develop a rule-based method to bridge the object domain and the concept domain. From features to objects, template matching based on statistical information is used to automatically map visual features to caption information.

Overall, we describe the generality of the proposed framework from the following perspectives:

(1) Extensibility: The content chain is extensible according to how we divide the targeted problem into smaller ones. The guidelines for dividing a problem into smaller ones often derive from domain knowledge or empiricism. Moreover, different branches may be extended on the basis of the same information. For example, we can extend the baseball analysis chain with a concept selection module to generate game abstraction, or extend it with a sequential mining module to mine the offense tactics or conventions for a specific team.

(2) Flexibility: The mapping function between two levels should be determined according to content characteristics. In the case that production rules are clear and can be explicitly exploited, deterministic mapping function can be built. Otherwise, statistical learning (either in supervised or unsupervised manners) techniques can be applied to find the nondeterministic mapping functions.

Unfortunately, the gaps between different semantic granularities are not always able to be bridged via computational methods. In these cases, user intervention may

play an important role for bridging the gap. Content-based image retrieval with relevance feedback is, therefore, invented to improve the retrieval performance. The proposed content chain framework describes a general process of semantic content analysis, including the computational and artificial manners.

2.3 Framework Correspondence

In this section, we show the correspondence between the proposed framework and the works we have done in this dissertation. We address this issue in terms of the representation of nodes and edges.

2.3.1 Semantic Concept Detection in Movies

Given a movie clip, we try to detect semantic concepts via audio information such that the highlighted parts can be automatically indexed. The targeted semantic concepts include gunplay and car-chasing scenes in action movies.

The semantic granularities of this task are illustrated as the Figure 2-4. Nodes representing different semantic granularities are audio features, audio events, and semantic concepts. The following items are corresponding to that in Figure 2-4.

(1) The audio features we extracted include timbre and perceptual features, which can be directly computed from audio signals.

(2) The audio events we modeled include gunshot, explosion, engine, and car-chasing sound effects. They are indicative elements for the highlighted parts of an action movie.

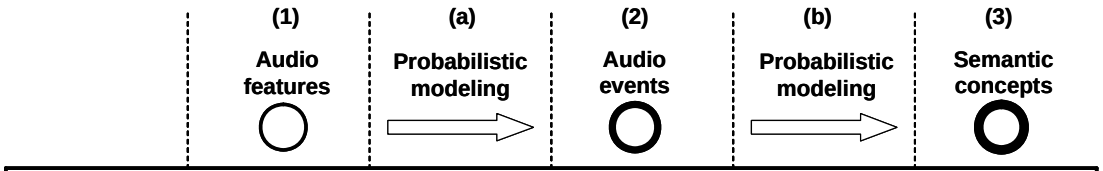
(3) Since aural information plays an important role for presenting highlighted scenes in action movies, we focus on the semantic concepts that demonstrate apparent aural effects. The semantic concepts we detected are gunplay scenes and car-chasing scenes.

The edges representing mapping functions between two semantic granularities are stated as follows:

(1) From features to events, hidden Markov models (HMM) are used to characterize the temporal variations/transition of audio features. Unlike clear and definite phonological rules in word construction, no artificial rules or conventions exist between audio features and audio events. Therefore, probabilistic techniques are used in this mapping.

(2) Likewise, although we know that there would be many gunshot and/or explosion sounds in gunplay scenes, there is no definite rule between these two semantic granularities. We again appeal to probabilistic techniques from

two perspectives: generative modeling (HMM) and discriminative modeling (support vector machine).



Analogy to language	Phones	Phonological rules	Words	Syntactic rules	Sentences
Nodes	Zero-crossing rate, short-time energy, MFCC, ...		Gunshot, explosion, engine, car-braking		Gunplay and car-chasing
Edges		Hidden Markov model		Hidden Markov model or support vector machine	

Figure 2-4. Correspondence between audio semantic concept detection and the content chain framework.

2.3.2 Semantic Concept Detection in Baseball Videos

The goal of this task is to automatically detect all semantic concepts, such as double play and homerun, in broadcasting baseball videos. On the basis of detected concepts, more elaborate applications can be built, or hidden knowledge between concepts can be obtained by cooperating with a mining module.

Figure 2-5 shows the semantic granularities of this task. Nodes representing different semantic granularities are visual features, caption information, baseball concepts, and game abstraction/knowledge.

(1) To detect where the caption information is, we extract color and edge information in video frames.

(2) At the level of caption information, we recognize the number of out, number of score, and base-occupation situation.

(3) Thirteen baseball concepts, which are commonly used to index baseball games, are automatically detected.

(4) Baseball concepts occurred in a game can be elaborately selected to construct a game abstraction, in the types of highlights or summary. Moreover, hidden knowledge, such as game tactics or offense conventions, can be found through applying a mining process to a pool of concepts.

The mapping functions between semantic granularities are stated as following:

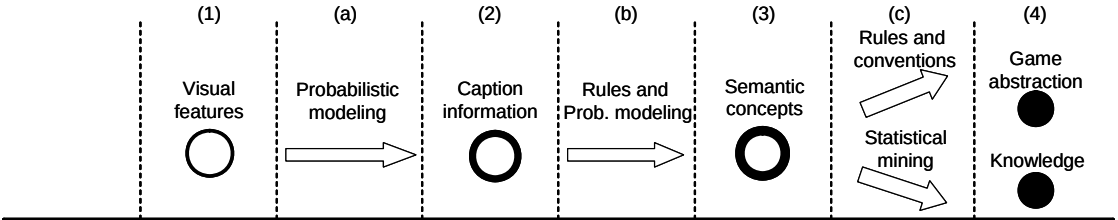
(1) Mapping between visual features and caption information is actually the problem of character recognition. We accomplish this task by using a

template matching technique.

(2) From caption information to semantic concepts, official baseball rules significantly help in detecting most concepts. For those concepts that cannot be discriminated by simply using rules, classifiers constructed based on visual information and speech are combined to make an explicit decision.

(3) From semantic concepts to game abstraction, some production rules or broadcasting conventions can be applied. On the other hand, a statistical mining method can be utilized to discover some hidden patterns or subtle knowledge based on large volume of game results.

The whole process goes through the feature space, the object space, the concept space, and the knowledge space.



Analogy to language	Phones	Phonological rules	Words	Syntactic rules	Sentences	Writing guidelines	Paragraphs
Nodes	Color, edge, …		Number of out, number of score, base-occupation		Single, double, homerun, strikeout, …		• Summary or highlight • Mining results
Edges		Template matching		Rule-based decision, kNN modeling, and classifiers fusion		• Conventions, user preference, or rules • Sequential mining	

Figure 2-5. Correspondence between baseball concept detection and the content chain framework.

2.3.3 Trajectory-based Analysis in Sports Videos

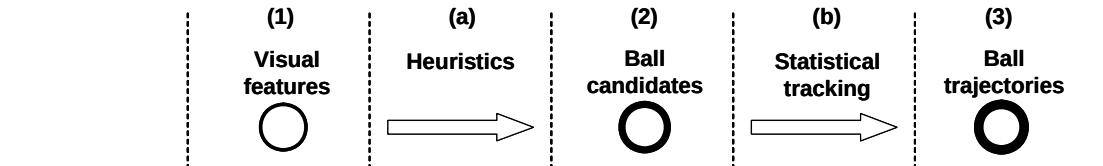
Ball trajectory is unique and important information in sports videos. Given a video clip, we want to extract the ball trajectory to facilitate semantic analysis.

Figure 2-6 shows the semantic granularities of this task. Nodes representing different semantic granularities are visual features, ball candidates in each frame, and the extracted ball trajectory.

(1) The primary clues for detecting the ball are color and shape.

(2) Ball-like objects are detected in each frame. Note that the real ball object may be occluded by other objects or is merged into the background.

(3) After tracking, we extract a reasonable ball trajectory that is concatenated by ball objects in each frame.



Analogy to language	Phones	Phonological rules	Words	Syntactic rules	Sentences
Nodes	Color, shape		Ball objects		Ball trajectories
Edges		Heuristics-based classifiers		Kalman filter-based tracking and physics-based validation	

Figure 2-6. Correspondence between ball trajectory extraction and the content chain framework.

The mapping functions between semantic granularities are stated as following:

(1) We can apply several heuristics, such as color, shape, and size, to detect ball candidates from each frame.

(2) From ball candidates to trajectory, Kalman filter-based tracking techniques are used to find a reasonable trajectory. Moreover, like the syntactic rules that constitute a grammatical sentence from words, a reasonable trajectory should conform to aerodynamics. We devise a physics-based method to validate the extracted ball trajectory.

2.4 Summary

In this chapter, we describe a content chain framework based on the idea of learning semantics from language. We analogize multimedia semantic analysis as a process for bridging different semantic granularities. In the proposed content chain framework, nodes represent semantic representation and edges represent the generative functions that generate higher-level entities based on lower-level entities. The extensibility and flexibility of this framework are discussed. On the basis of the developments of deterministic or nondeterministic functions, this framework describes the process of semantic analysis and can be the foundation of the following chapters. Accordingly, we give the correspondence between the content chain framework and the works we will describe in this dissertation. Variety and generality of the proposed framework are shown in the context.

Chapter 3
Semantic Analysis in Movies through Audio Information

3.1 Introduction

As the rapid advance in media creation, storage, and compression technologies, large amounts of multimedia content have been created and disseminated by various ways. Massive multimedia data challenge users in content browsing and retrieving, thereby motivating the urging needs of information mining technologies.

To facilitate effective or efficient multimedia document indexing, many research issues have been investigated. However, they pose many problems in today's applications. The semantic gap between low-level features and high-level concepts degrades the performance of multimedia content management systems. Similarities in low-level features don't certainly match with user's perception. Scenes or shots are associated due to semantics rather than physical features like color layouts and motion trajectories. Therefore, it would be more reasonable to discover information from meaningful events or objects rather than physical features.

To diminish the differences between analysis results and user's expectation, two research directions are emerged. The first is to detect attractive parts of movies or TV programs by exploiting the domain knowledge and production rules. According to media aesthetics [Zett99], which includes the study and analysis of media elements commonly applied, the related studies attempt to uncover the semantic and semiotic information by computational frameworks. Preliminary results have been reported on film tempo analysis [Dora02] and scare scene detection in horror movies [Monc03].

Semantic indexing is another emerging study that identifies objects and events in audiovisual streams and facilitates semantic retrieval or information mining. The major challenge of this work is to bridge the gaps between physical features and semantic concepts. Studies on semantic indexing can be separated into two levels: isolated audio/video event detection and semantics identification. Former studies [Cai03][Naph98] took advantage of HMM-based approaches to tackle event detection. Audio events such as applause, laughter, and cheer are modeled. However, in today's applications, detecting isolated audio/video events is not quite intuitive to users. For

example, rather than identifying individual gunshots in an action movie, we are more likely to recognize a scene of gunplay, which may consist of a series of gunshots, explosions, sounds of jeeps and screams from soldiers. Such a scene conveys a solid semantic meaning and is at a reasonable granularity for semantic retrieval. For modeling visual semantics, some approaches based on Bayesian network [Naph02] and support vector machine [Smit03] have been proposed to fuse the information of visual events and to infer some semantic concepts, such as "outdoor" or "beach" concepts. However, few studies are reported to perform audio-based semantic concept detection. In some types of videos, such as action movies, audio information plays more important role than visual ones. For example, a gunplay scene may occur in a rainforest or a downtown street, at day or night, which have significant variations in vision. On the contrary, aural information remains similar in different gunplay scenes, and some typical audio events (e.g. gunshot and explosion sounds in gunplay scenes) significantly provide the clues for detecting semantic concepts.

Due to rapid shot changes and dazzling visual variations in action movies, our studies focus on analyzing audio tracks and accomplish semantic indexing via aural clues. In this chapter, an integrated hierarchical framework is proposed to detect two semantic concepts, i.e. "gunplay" and "car-chasing," in action movies. To characterize these two semantic concepts by event fusion, "gunshot" and "explosion" sound effects are detected for "gunplay" scenes, and "car-braking" and "engine" sounds are detected for "car-chasing" scenes. For audio event modeling, HMM-based approaches that have been applied in visual event modeling [Naph98] are used. Then gunplay scenes and car-chasing scenes are modeled based on the statistical information from audio event detection. For semantic concept modeling, generative (hidden Markov model) and discriminative (support vector machine) approaches are investigated. We view semantic concept detection as a problem of pattern recognition, and similar feature values (detection results of audio events) would be fused to represent a semantic concept. For example, gunplay scenes may have similar gunshot and explosion occurrence patterns and are distinguished from other scenes by pattern recognition techniques. We discuss how the fusion approaches work and show the effectiveness of this event fusion framework. The results of semantic concept detection can be applied to multimedia indexing and facilitate efficient media access.

3.2 Hierarchical Audio Models

The semantic indexing process is performed in a hierarchical manner. At the audio event level, the characteristics of each audio event are modeled by an HMM in terms of the extracted audio features. At the semantic concept level, the results of audio

event detection are fused by using generative (HMM) or discriminative (SVM) schemes.

3.2.1 Audio Event and Semantic Concept

Audio events are defined as short audio clips which represent the sound of an object or an event. On the basis of elaborately selected audio features, fully connected (ergodic) HMMs are used to characterize audio events, with Gaussian mixtures modeling for each state. Four audio events, including gunshot, explosion, engine, and car-braking, are considered in this work.

In this study, we aim at indexing multimedia documents by detecting semantic concepts. A semantic concept may be derived from the association of various events. Therefore, we introduce the idea of modeling a semantic concept via the context of relevant events. To characterize a semantic concept, the information of specific audio events, which are highly relevant to some semantic concepts, are collected and modeled. In action movies, the occurrences of gunshot and explosion events are used to characterize gunplay scenes. The occurrences of engine and car-braking events are used to characterize car-chasing scenes.

For a semantic concept, there may be no specific evolution pattern along the time axis. For example, in a gunplay scene, we cannot expect that explosions always occur after gunshots. Moreover, there may be some silence segments which contain no relevant audio events, but they are viewed as parts of the same gunplay scene in human's sense. Figure 3-1 illustrates examples of "gunplay" semantic concepts. The audio clip from t1 to t2 is a typical gunplay scene which contains mixed relevant audio events. In contrast to this case, no relevant event exists from t4 to t5 and from t6 to t7. However, the whole audio clip from t3 to t8 is viewed as the same scene in user's sense, as long as the duration of the "irrelevant clip" doesn't exceed users' tolerance. Therefore, to model the characteristics of semantic concepts, we develop an approach that takes a series of events along the time axis into account rather than just the information at a time instant.

Note that multiple audio events may occur simultaneously, as shown in the duration from t1 to t2 in Figure 3-1. Some studies have been conducted to separate mixed audio signals in speech and music domains, by using independent component analysis [Hyva01]. The reported works are mainly performed on synthetically mixed audio signals or sounds recorded at simple acoustic conditions. However, separating mixed audio effects recorded in complicated real-world situations is not widely studied. In this work, when multiple audio events are mixed, we simply select two representative events to describe the characteristics of the corresponding audio clip. Although separating mixed audio effects is possible, elaborate studies on this issue are

beyond the scope of this dissertation.

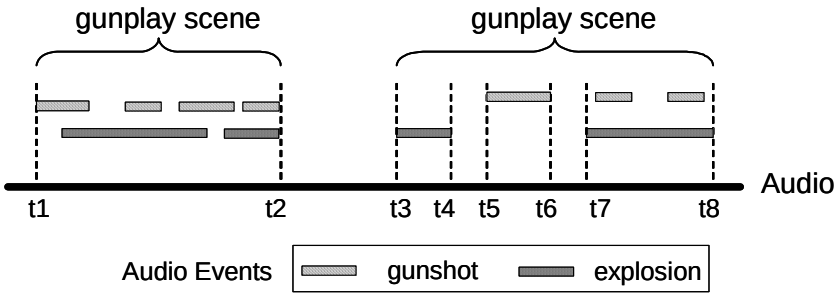


Figure 3-1. Examples of audio semantic concepts.

3.2.2 Hierarchical Framework

The proposed framework consists of audio event and semantic concept modeling. Some essential audio features from training corpus are first extracted and modeled by HMMs, as shown in Figure 3-2(a). After constructing each audio event model, the likelihood of a test audio segment with respect to each audio event can be computed through the Forward algorithm [Rabi89]. To determine how a segment is close to an audio event, a confidence metric based on the likelihood ratio test [Duda01] is defined. We say that the segments with higher confidence scores from the gunshot model, for example, imply higher probability of the occurrence of gunshot sounds

In the stage of semantic concept modeling/detection, the confidence values from event detection constitute the cues for characterizing high-level semantic concepts. The *pseudo-semantic features* that indicate the occurrences of events are constructed to represent the association of audio clips. We call them pseudo-semantic features because they represent the interrelationship of several audio events, which are grounds for users to realize what the clip presents. With these features, two approaches based on generative and discriminative strategies are investigated to model semantic concepts, as shown in Figure 3-2(b). As the usage in pattern recognition and data classification, HMM and SVM shed lights on clustering these pseudo-semantic features and facilitate detection processes.

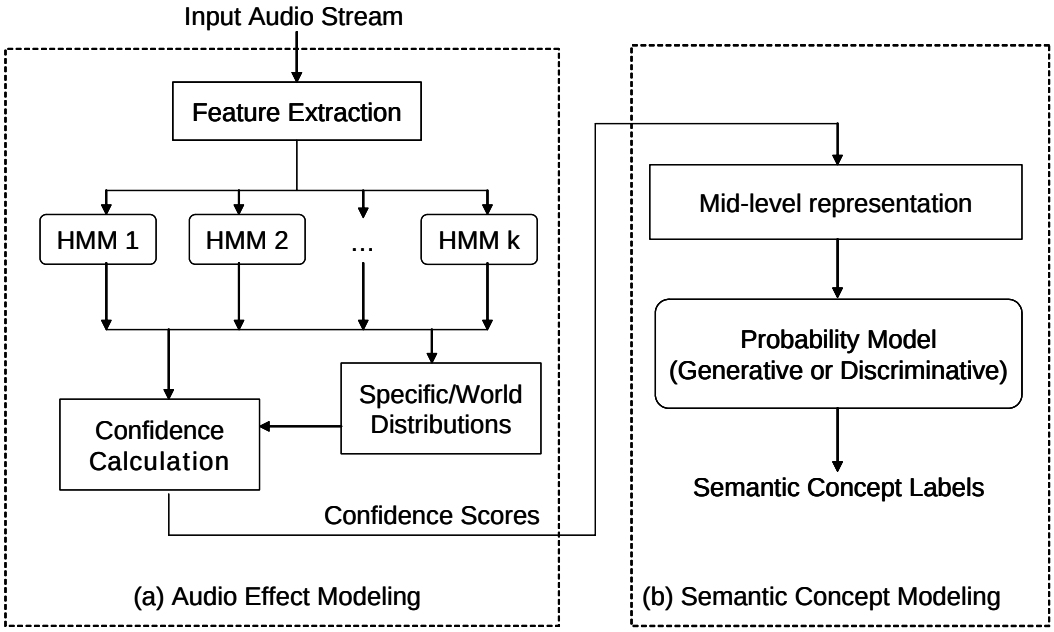


Figure 3-2. The proposed hierarchical framework contains (a) audio event and (b) semantic concept modeling.

3.3 Audio Feature Extraction

One important factor for pattern recognition is the selection of suitable features that characterize original data adequately. To analyze audio sequences, several audio features from time-domain amplitude and frequency-domain spectrogram are extracted and utilized. In our experiments, all audio streams are down-sampled to the 16 KHz, 16 bits and mono-channel format. Each audio frame is of 25 ms, with 50% overlaps. Two types of features, i.e. perceptual features and Mel-frequency Cepstral Coefficients (MFCC), are extracted from each audio frame. The perceptual features include short-time energy, band energy ratio, zero-crossing rate, frequency centroid, and bandwidth [Wang00]. These features are shown to be beneficial for audio analysis and are widely adopted [Lu02, Zhan98, Tzan02, Lu03, Cai03].

3.3.1 Short-Time Energy

Short-time energy (STE) is the total spectrum power of an audio signal at a given time and is also referred to loudness or volume in the literature. It provides a convenient representation of the amplitude variations over time. This feature is useful for detecting silence or distinguishing speech from non-speech signals. The STE of frame n is calculated by

$$v(n) = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} s_n^2(i)}\ , \qquad (3\text{-}1)$$

where $s_n(i)$ is the ith sample in the nth frame audio signal and N is the frame length. Note that the STE of an audio clip is device-dependent. Audio clips may be produced from or recorded on different devices under different circumstances, so the basis or mean volume of each audio clip may not be identical. To reduce the clip level fluctuation of volume mean, we normalize the volume of a frame based on the maximum volume of the corresponding audio clip.

3.3.2 Band Energy Ratio

The distributions of energy in various kinds of audio signals, such as music and speech, are important spectral characteristics and can be used to model the spectrum power more accurately. The band energy (BE) is defined as the energy content of a signal, for a given frame, in a band of frequencies:

$$BE_i = \int_{\omega_L}^{\omega_U} |SF(\omega)|^2 \, d\omega, \tag{3-2}$$

where $SF(\omega)$ denotes the short-time Fourier transform coefficients, and ω_U and ω_L are the upper and the lower bound frequencies of the sub-band i, respectively.

In order to model the characteristics of spectral distribution more accurately, the band energy ratio is considered in this work. The frequency spectrum is divided into four sub-bands with equal frequency intervals, then the band energy ratios are computed as:

$$BER_i = \frac{BE_i}{\sum_i BE_i}, \; 1 \leqq i \leqq 4. \tag{3-3}$$

3.3.3 Zero-Crossing Rate

Zero-crossing rate (ZCR) is defined as the average number of signal sign changes in an audio frame. It gives a rough estimate of frequency content and has been extensively used in many audio processing applications, such as voiced and unvoiced components discrimination, endpoint detection, and audio classification. Moreover, by combining ZCR with volume, we avoid misclassifying low volume unvoiced speech components as silence.

Average ZCR in discrete case is calculated as:

$$ZCR = \frac{1}{2N} \sum_{i=1}^{N-1} |sign(s(i)) - sign(s(i-1))|, \tag{3-4}$$

where $sign(s(i)) = \begin{cases} 1, & s(i) \geq 0, \\ -1, & s(i) < 0. \end{cases}$

3.3.4 Frequency Centroid

Frequency centroid (FC) is the first-order statistics of the spectrogram, which represents the power-weighted median frequency of the spectrum in a frame. It has been shown that the frequency centroid is related to human's aural perceptions, so it is also referred to brightness in some literatures. FC is formulated as:

$$FC = \frac{\int_0^\infty \omega \,|\, SF(\omega) \,|^2 \, d\omega}{\int_0^\infty |\, SF(\omega) \,|^2 \, d\omega}. \tag{3-5}$$

3.3.5 Bandwidth

Bandwidth (BW) is the second-order statistics of the spectrogram, which represents the power-weighted standard deviation of the spectrum in a frame. The definition of BW is:

$$BW = \sqrt{\frac{\int_0^\infty (\omega - FC)^2 \,|\, SF(\omega) \,|^2 \, d\omega}{\int_0^\infty |\, SF(\omega) \,|^2 \, d\omega}}. \tag{3-6}$$

Frequency centroid and bandwidth are usually combined to describe statistical characteristics of the spectrum in a frame. They respectively represent the 'center of gravity' and variances of the spectrogram, and their reliability and effectiveness have been demonstrated in previous research [Wang00].

3.3.6 Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients (MFCCs) are the most widely used features in speech recognition and other audio applications. They are computed as the inverse Fourier transform of the logarithmic spectrum in a frame. MFCCs effectively represent human perception because the non-linear scale property of frequencies in human hearing system is considered. Based on the temporal variation of MFCC, an audio sequence can be effectively discriminated as speech or music. In this work, based on the suggestion in [Li00], 8-order MFCCs are computed from each frame.

The extracted features from each audio frame are concatenated as a 16-dimensional (1(STE)+4(BER)+1(ZCR)+1(FC)+1(BW)+8(MFCC)) feature vector. Details of the audio feature extraction processes can be found in [Wang00]. Note that the temporal variations of the adopted features are also considered. That is, the differences of the features between two adjacent frames are calculated. Therefore, by concatenating the feature vector of the *i*-th frame and the differences between the *i*-th and the (*i*+*1*)-th frames, a 32-dimensional (32-D) vector is finally generated for each audio frame.

3.4 Audio Event Modeling

Detecting specific events in audio streams is crucial, which will benefit the higher-level analysis of multimedia documents and facilitate the modeling of human attention and perception more accurately. This section addresses some issues of audio event modeling, including the determination of model size, model training process, and the construction of pseudo-semantic features for semantics modeling.

3.4.1 Model Size Estimation

We use HMMs to describe the characteristics of audio events. The 32-D feature vectors from a type of audio event are grouped into several sets. Each set denotes one kind of timbre, and is modeled later by one state of an HMM. Determining a proper model size is crucial in applying HMMs. The state number should be large enough to describe the variations of features, while it should also be compact when we consider computational cost of model training process. In this work, adaptive sample set construction technique [Bow02] is adopted to estimate a reasonable model size of each audio event. The algorithm is described in the box of the next page.

The thresholds t_1, t_2, and ρ are heuristically designated such that different clusters (states) have distinct differences. In this work, ρ is set as 0.1 to guarantee more than ninety percent of data are clustered. The initial values of t_1 and t_2 could be empirically set, as their initial values just affect the number of iterations for convergency, but not the final results that indicate the number of major clusters. The distance measure $d_i(\mathbf{v}, \mathbf{z}_i)$ we used is Euclidean distance. As Gaussian mixtures are able to handle the slight differences within each state, we tend to keep the number of states less than ten by considering the effectiveness and efficiency of the training process.

A professional sound effects library is used to be the training corpus [Soun06]. Through the above process, the estimated state numbers for car-braking and engine are two and four, and both the state numbers for gunshot and explosion are six. These results make sense because, for each audio event, various kinds of sounds are collected in this sound library, and these numbers represent the degree of variations of each audio event. For example, the sounds of rifle and hand/machine gun are all collected as the gunshots. They vary significantly and should be represented by more state numbers than simple sounds, such as the sharp but simple car-braking sounds.

1. Define two thresholds: t_1 and t_2, with $t_1 > t_2$.
2. Take the first sample $\mathbf{v}_1$ as the representative of the first cluster: $\mathbf{z}_1 = \mathbf{v}_1$, where $\mathbf{z}_1$ is the center of the first cluster.
3. Take the next sample $\mathbf{v}$ and compute its distance $d_i(\mathbf{v}, \mathbf{z}_i)$ to all the existing clusters, and choose the minimum of d_i: $\min\{d_i\}$.
   (a) If $\min\{d_i\} \leqq t_2$, assign $\mathbf{v}$ to cluster i and update the center of this cluster: $\mathbf{z}_i$.
   (b) If $\min\{d_i\} > t_1$, a new cluster with center $\mathbf{v}$ is created.
   (c) If $t_2 < \min\{d_i\} \leqq t_1$, no decision will be made as the sample $\mathbf{v}$ is in the intermediate region.
4. Repeat *step 3* until all samples have been checked once. Calculate the variances of all clusters.
5. If the current variance is the same as that of the last iteration, the clustering process has converged, go to *step 6*. Otherwise, return to *step 3* for further iteration.
6. If the number of unassigned samples is larger than a certain percentage ρ ($0 \leqq \rho \leqq 1$), increase t_1 or decrease t_2 while remaining $t_2 > 2t_1$ and start with *step 2* again. Otherwise, assign the unassigned samples to the nearest clusters and end the process.

3.4.2 Model Training

For modeling gunplay and car-chasing scenes in action movies, the audio events we modeled are gunshot, explosion, engine, and car-braking. For each audio event, 100 short audio clips each with length 3-10 seconds are selected from the SoundIdeas sound effects library as the training data. In the training stage, the training audio streams are segmented into overlapped frames, and the features described in Section 3 are extracted. Based on these features, a complete specification of HMM, which

includes two model parameters (model size and number of mixtures in each state) and three sets of probabilities (initial probability, observation probability, and transition probability), are determined. The model size and initial probability could be decided by the clustering algorithm described in the previous subsection, and the number of mixtures in each state is empirically set as four because it's insensitive to the system performance according to our experiments. The Baum-Welch algorithm is then applied to estimate the transition probabilities between states and the observation probabilities within each state. Finally, four HMMs are constructed for the audio events we concern. Details of the HMM training process can be found in Appendix A and the eminent tutorial [Rabi89].

3.4.3 Specific and World Distribution

After audio event modeling, for a given audio clip, the log-likelihood values with respect to each event model are calculated by the Forward algorithm. Because a sound effect often lasts more than one second, the basic units we analyze for event detection are 1-sec audio segments (called *analysis window* in this work), with 50% overlapping with adjacency segments. In event detection, the most important issue is how to decide whether an event occurs. According to the definition of HMM's evolution problem, the solution of Forward algorithm scores how well a given model matches a given observation sequence. However, unlike audio classification or speech recognition, we cannot simply classify an audio segment as a specific event even if it has the largest log-likelihood value. It may just present general environmental sound and doesn't belong to any predefined audio event. Therefore, to evaluate how likely an audio segment belongs to a specific audio event, a log-likelihood based decision method motivated from the speaker and world models in speaker verification [Zilc01] is proposed.

For each type of audio event, two distributions are constructed from the log-likelihood values. The first distribution represents the distribution of the log-likelihood values obtained from a specific event model i with respect to the corresponding audio sounds. For example, from the "engine" model with the set of engine sounds as inputs, the resulting log-likelihood values are gathered to form the distribution. Figure 3-3(a) illustrates this construction process, and we call this distribution the *specific distribution,* $p(x|\theta_1)$*,* of the engine model. In contrast, the second distribution represents the distribution of the log-likelihood values obtained from a specific audio event model with respect to other audio sounds. As shown in Figure 3-3(b), the *world distribution,* $p(x|\theta_0)$*,* of the engine model is constructed from the log-likelihood values gathered from the engine model with the sets of gun, explosion, and car-braking sounds as inputs. Overall, engine model's specific

distribution describes how the engine HMM evaluates engine sounds, while its world distribution describes how the engine HMM evaluates other kinds of sounds. These two distributions show how log-likelihood values vary with respect to a specific audio event and help us discriminate a specific audio event from others.

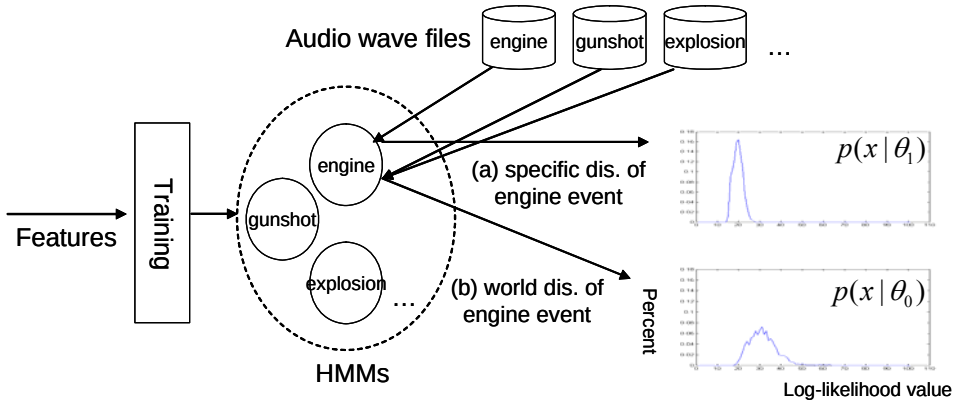


Figure 3-3. Construction of (a) specific distribution $p(x|\theta_1)$ and (b) world distribution $p(x|\theta_0)$ for engine events.

3.4.4 Pseudo-Semantic Features

Based on specific distributions and world distributions, we can evaluate how likely an audio segment (as the unit of analysis window) belongs to a specific audio event and compute a confidence score. The audio segments with low average short-time energy and zero-crossing rate are first marked as silence, and the corresponding confidence scores with respect to all audio events are set as zero. For non-silence segments, the extracted feature vectors are input to the four HMMs. For a given audio segment, assume that the log-likelihood value from an event model is x, the confidence score with respect to audio event i is defined as:

$$c_i = \frac{p_i(x|\theta_1)}{p_i(x|\theta_0)}, \tag{3-7}$$

where $p_i(x|\theta_1)$ and $p_i(x|\theta_0)$ respectively denote the magnitudes of log-likelihood value x with respect to the specific and world distributions of event i. The value c_i represents the confidence score of the audio segment belonging to event i. Note that if the testing audio segment is out of the pre-defined set, both log-likelihood values with respect to the specific and world distributions are very likely to be zeros. We heuristically set the value c_i as zero for rejection in this case.

By the definition in Section 5.2.1, a semantic concept often lasts for at least a period of time, and not all the relevant audio events exist at every time instant. Therefore, the confidence scores of several consecutive audio segments are considered integrally to capture the temporal characteristics in a time series [Tzan02].

We define a *texture window* (c.f. Figure 3-4(b)) of 5-sec long, with 2.5-sec overlaps, to go through the confidence scores of *analysis windows*.

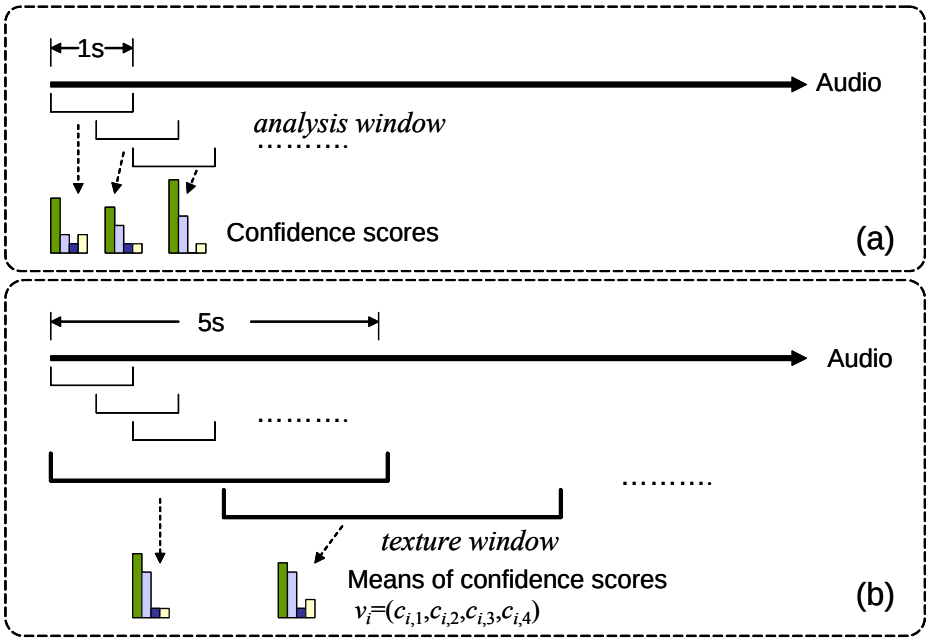


Figure 3-4. Pseudo-semantic features calculation for semantic concepts modeling. (a) Analysis windows and (b) texture windows.

For describing the semantic concepts of audio streams, *pseudo-semantic features* that are constructed from the results of event detection are proposed. Based on the idea of event fusion, the pseudo-semantic features for each texture window are constructed as follows.

1. For each texture window, the mean values of confidence scores are calculated:
   $m_i = mean(c_{i,1}, c_{i,2}, \ldots, c_{i,N})$, $i = 1, 2, 3, 4$,
   where $c_{i,j}$ denotes the confidence score of the j-th analysis window with respect to event i, and N denotes the total number of analysis windows in a texture window.
   By the settings described above, nine analysis windows ($N = 9$), with 50% overlapping, construct a texture window. The corresponding sound effects to events 1 to 4 are "gunshot," "explosion," "engine," and "car-braking."
2. Let b_i be a binary variable describing the occurrence situation of event i. The pseudo-semantic feature vector v_t for the t-th texture window is defined as:
   $v_t = [b_1, b_2, b_3, b_4]$,
   $b_i=1$ and $b_j=1$ if the corresponding m_i and m_j are the first and the second maximums of (m_1, m_2, m_3, m_4). Otherwise, $b_k = 0$.

3. The total pseudo-semantic features V is represented as:

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_T \end{bmatrix},$$

where T is the total number of texture windows in the audio clip.

Calculating running mean values of confidence scores is to describe the characteristics over a number of analysis windows. We did also consider running variances in pseudo-semantic features construction, but the final detection performance doesn't change significantly. The process of binarization is to emphasize the differences between confidence values with respect to different events. If a sound effect is more apparent than others, larger confidence score will be obtained. Therefore, we prompt the events with the first and the second largest confidence values and suppress those with smaller confidence values.

We call the features pseudo-semantic features because they represent the intermediate characteristics between low-level physical features and high-level semantic concepts. The audio segments with higher confidence scores in the audio events relevant to a concept are more likely to convey this concept. For example, the audio segments with higher confidence scores in gunshot and explosion events somehow drop hints on the occurrence of gunplay scenes. To accomplish fusing information from different events, we investigate generative and discriminative approaches to model the pseudo-semantic features. HMM is selected to be the instance of generative approach, and SVM is treated as the instance of discriminative approach.

3.5 Generative Modeling for Semantic Concept

For describing a sophisticated semantic concept, a general model, e.g. Gaussian mixture model, that only covers the event data distributions may not be enough. It is preferable to explicitly model the time duration density by including the concept of state transition. The appearance of relevant events doesn't remain the same at every time instant. There would be some segments with low confidence scores because the sound effect is unapparent or is influenced by other environmental sounds. On the other hand, some segments may pose higher confidence because the audio events raise or explosively emerge. A model with more descriptive capability should take the temporal variations into consideration.

HMM is widely applied in speech recognition to model the spectral variations of acoustic features. It captures the time variation and state transition duration from training data. In speech-related applications, the left-right HMMs, which only allow state index increasing (or staying the same) as time goes by, are considered to be suitable. But in the case of semantic concept modeling, there is no specific consequence formally representing the time evolution. Therefore, ergodic HMMs, or the so-called fully connected HMMs, are used in this work.

3.5.1 Model Training

To perform model training, ten gunplay and car-chasing scenes, each with length 3-5 minutes, are manually selected from several Hollywood action movies as the training corpus. Based on user's sense, the movie clips that completely present gunplay or car-chasing scenes are selected, no matter how many gunshots, engine, or other relevant audio events occur. In model training, audio events are first detected and the pseudo-semantic features are constructed based on the results of event detection. The pseudo-semantic features from each semantic concept are then modeled by an HMM again. For each HMM, the state number is estimated as two and the characteristics of each state are described by one Gaussian mixture. The obtained HMMs elaborately characterize the densities of time-variant features and present the structures of sophisticated semantic concepts.

3.5.2 Semantic Concept Detection

The semantic concept detection process is conducted following the same idea as that of the audio event detection. For every 5-sec audio segment (a texture window), the log-likelihood calculated by the Forward algorithm represents how the semantic concept models match the given pseudo-semantic features. The binary indicator $\alpha_{s,t}$ is defined to show the appearance of semantic concept s at the t-th texture window, $s = 1$ and 2 respectively for gunplay and car-chasing scenes. That is,

$$\text{If } \sigma_s > \varepsilon, \alpha_{s,t} = 1. \text{ Otherwise, } \alpha_{s,t} = 0, \tag{3-8}$$

where σ_s is the log-likelihood value under semantic concept model s, and ε is a pre-defined threshold for filtering out those texture windows with too small values. The threshold can be adjusted by the user to tradeoff the precision and recall of semantic concept detection.

3.6 Discriminative Modeling for Semantic Concept

Support vector machine (SVM) has been shown to be a powerful discriminative technique [Vapn98]. It focuses on structural risk minimization by maximizing the

decision margin. The goal of SVM is to produce a model which predicts target value of data instances in the testing set. In our work, we view the detection process as classifying testing feature vectors (pseudo-semantic features) into one of the predefined classes (semantic concept). Thus we exploit SVM classifiers to distinguish the textures of "gunplay," "car-chasing," and "others" scenes. Although the features obtained from the same semantic concept may disperse variably in the feature space (which is caused by the various patterns of the same semantic concept), the SVM classifier which maps features into a higher dimensional space and finds a linear hyperplane with the maximal margin can effectively distinguish one semantic concept from others.

Note that SVMs were originally designed for binary classification. In this work, we should classify a segment into three scenes, thus the SVM classifiers should be extended to handle multiclass classification both in training and testing processes.

3.6.1 Model Training

Recently, a few researches are conducted to reduce a multiclass SVM into several binary SVM classifiers [Plat00]. According to the performance analysis of multiclass SVM classifiers [Hsu02], we adopt the 'one-against-one' strategy to model these three scenes. Three SVM models are constructed, i.e. "gunplay vs. car-chasing," "gunplay vs. others," and "car-chasing vs. others." For training each classifier, feature vectors are collected and their labels are manually determined to construct instance-label pairs (x_i, y_i), where $x_i \in R^n$ and $y_i \in \{1, -1\}$. An SVM finds an optimal solution of data separation by mapping the training data x_i to a higher dimensional space by a kernel function ϕ up to a penalty parameter C of the error term. In model training, the kernel function we used is the radial basis function (RBF), which has been suggested in many SVM-based researches. That is, our kernel function is

$$K(x, y) = \exp(-\gamma \left\| x_i - x_j \right\|^2), \quad \gamma > 0. \tag{3-9}$$

It is crucial to find the right parameters C and γ in RBF. Therefore, we apply five-fold cross validation with a grid search of varying (C, γ) on the training set to find the best parameters achieving the highest classification accuracy.

For training SVM classifiers, the pseudo-semantic features obtained from four audio events are labeled manually based on the unit of a texture window. Then all labeled texture windows are collected together to produce the training vectors. Three binary SVM classifiers will be combined later to identify which semantic concept a texture window belongs to. Details of the idea of SVM and training process are addressed in Appendix B.

3.6.2 Semantic Concept Detection

In semantic concept detection, the Decision Directed Acyclic Graph SVM algorithm (DAGSVM) [Plat00] is applied to combine the results of one-against-one SVMs. The DAGSVM algorithm has been shown to be superior to existing multiclass SVM algorithms in both training and evaluation speeds. Figure 3-5 illustrates one example of the detection procedure. Initially, the test vectors are viewed as the candidates for all three concepts. In the first step of detection, the test vectors are input to the root SVM classifier, i.e. "car-chasing vs. others" classifier. After this evaluation, the process branches to left if more vectors are predicted as the "others" category, and the "car-chasing" concept is removed from the candidate list. The "gunplay vs. others" classifier is then used to re-evaluate the test vectors. After these two steps, the vectors representing the characteristics of texture windows are labeled as "gunplay" or "others."

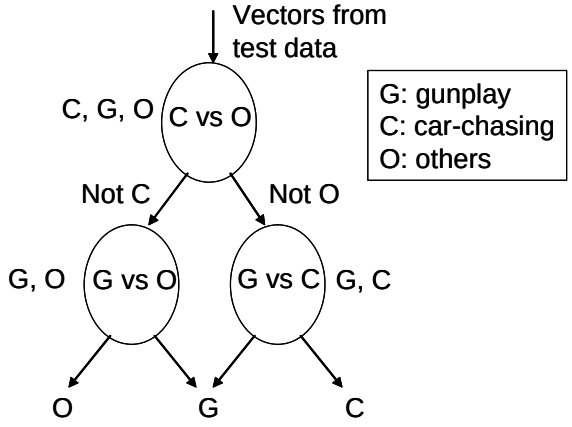


Figure 3-5. The testing procedure of DAGSVM.

The DAGSVM separates the individual classes with large margins. It is safe to discard the losing class at each one-against-one decision because, for the hard margin case, all of the examples of the losing class are far away from the decision surface. Hence, the choice of the class order in detection procedure is arbitrary.

3.7 Performance Evaluation

We may first describe the characteristics of sound effects in movies before preparing the evaluation data. According to our observations, although the acoustic conditions may vary differently in different movies, the sound effects indicating a specific semantic concept fall into several fixed types. The reasons for this phenomenon

include: 1) there have been some conventions to construct a concept in movie making, and 2) the sound effects are often added or embellished after shooting according to commonly used techniques. For example, in a gunplay scene, the sounds of gunshots can be often categories as several canonical types: hand gun, rifle, machine gun, and ricochet. Therefore, very huge amount of training data are not necessarily required.

For each audio event, 100 short audio clips each with length 3-10 sec are selected. For semantic concept modeling, because there is no standard corpus for audio semantic concepts, the evaluation data are manually selected from Hollywood movies. Thirty movie clips each with length 3-5 minutes are selected and labeled for each semantic concept. Twenty-four clips of them are used as the dataset for model training, while the rest are used for model testing. Note that the criteria of selecting training data for audio events and semantic concepts are different. For semantic concept modeling, we collected the "gunplay" and "car-chasing" scenes based on the experienced user's subjective judgments, no matter how many relevant audio events exist in the scene. On the contrary, the training data for audio event modeling are short audio segments that are exactly the audio events.

We evaluate the performance for both audio event detection and semantic concept detection. Moreover, the effectiveness of this later fusion approach is compared with that in the baseline approach, which only exploits low-level audio features and works in an early fusion manner.

3.7.1 Evaluation of Audio Event Detection

In audio event detection, audio streams are segmented into audio clips through analysis windows, as illustrated in Figure 3-4(a), and the log-likelihood values of audio clips in each analysis window with respect to four audio events are evaluated. The audio clip in an analysis window is correctly detected as the event i if its corresponding confidence score is larger than a predefined threshold and is the maximum value with respect to all events. That is,

$$C = max(c_1, c_2, c_3, c_4) \text{ and } C > \delta, \tag{3-10}$$

where c_i (i = 1, …, 4), calculated from eq. (3-7), is the confidence score with respect to event i, and δ is determined by the Bayesian optimal decision rule [Duda01] on the basis of specific and world distributions. We decide that the analysis window with confidence score x belongs to a specific event (category θ_1) if

$$\lambda_{01} P(\theta_1 \mid x) > \lambda_{10} P(\theta_0 \mid x), \tag{3-11}$$

where λ_{ij} is the cost incurred for deciding θ_i when the true state of nature is θ_j.

By employing Bayes formula, we can replace the posterior probabilities by the prior probabilities and conditional densities. Then we decide θ_1 if

$$\lambda_{01} p(x \mid \theta_1) P(\theta_1) > \lambda_{10} p(x \mid \theta_0) P(\theta_0), \tag{3-12}$$

and otherwise decide θ_0.

Then we alternatively rewrite eq. (3-12) and decide θ_1 if

$$C = \frac{p(x|\theta_1)}{p(x|\theta_0)} > \frac{\lambda_{10}}{\lambda_{01}} \frac{P(\theta_0)}{P(\theta_1)} = \delta . \quad (3\text{-}13)$$

The prior probabilities are estimated based on our training data. The costs λ_{10} and λ_{01} could be adjusted to vary the value of threshold such that higher precision or recall could be achieved in the detection stage.

3.7.1.1 Overall Performance

The overall detection performance is listed in Table 3-1. The average recall is over 70% and the average precision is about 85%. Although the detection accuracy is often sequence-dependent and is affected by confused audio effects, the reported performances support the applicability and superiority of the event modeling. In addition, different audio events have different evaluation results. Because the car-braking sounds are often very short in time (less than one second, which is the length of one basic analysis unit defined in our work) and are mixed with other environment sounds, the detection accuracy is particularly worse than others. This situation is different from gunshot sounds because there is often a continuity of gunshots (the sounds of a machine gun or successive handgun/rifle shoots) in a gunplay scene.

Table 3-1. Overall performance of audio event detection.

Audio Event	Recall	Precision
Gun	0.938	0.95
Explosion	0.786	0.917
Car-Braking	0.327	0.571
Engine	0.890	0.951
Average	0.735	0.847

The detection performance is more encouraging if we neglect the particular case in car-braking detection. For other audio events, the average recall is 87% and the average precision is 94%. On the other hand, because the car-braking sound is a representative audio cue of car-chasing scenes, we still take the detection results of car-braking sounds into account in car-chasing concept modeling.

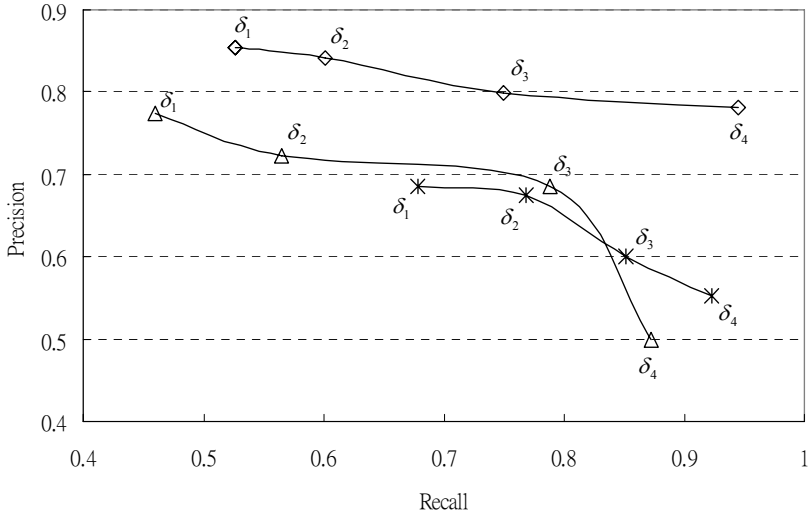


Figure 3-6. Three examples of detection performance with different thresholds ($\delta_1 > \delta_2 > \delta_3 > \delta_4$).

We also briefly investigate how different thresholds in eq. (3-10) affect the detection performance. When we penalize misclassifying θ_0 as θ_1 (false alarm) more than the converse (i.e. $\lambda_{10} > \lambda_{01}$), we get larger threshold δ, and hence higher precision but lower recall are expected. Figure 3-6 shows detection performance with four different thresholds ($\delta_1 > \delta_2 > \delta_3 > \delta_4$) from three different test sequences. Note that the trend of detection performance conforms to the general principle of pattern classification, while detection results are sequence-dependent.

3.7.1.2 Performance Comparison

To compare the performance of video retrieval/indexing between various approaches, some institutes such as TREC Video Retrieval Evaluation [TREC06] developed corpus for video event evaluation. However, few standard datasets are designed for audio event detection. Most works of audio event detection (including our work) use privately collected datasets. Direct comparison between different approaches, which use different datasets and model different events, is not plausible. However, in order to show that the proposed approach achieves promising performances in detecting various audio events, we refer to other works that focused on audio events in sports games [Wang04-3], TV shows [Cai03], and movies [Naph01].

Because not all referred works report precision and recall values, we only list the detection accuracy (precision) in Table 3-2 for fair comparison. In [Wang04-3], four audio events including “acclaim,” “whistle,” “commentator speech,” and “silence” are detected in soccer videos, while speech and silence generally are not viewed as

special sound effects. More than 90% of detection accuracy is achieved. In [Cai03], the events of "laughter," "applause," and "cheer" are detected in TV shows. For each event, average precision values from three test sequences are listed. The most similar work to ours is [Naph01]. It also introduces a variation of HMM to model audiovisual features of explosion events. More than 86% of explosion events are correctly detected, while we achieve 91.7% of precision. From these results, we can see that the proposed audio event detection module works at least as well as other reported approaches and is capable of being a robust basis for higher level modeling.

Table 3-2. Detection accuracy of different approaches.

	[Wang04-3]		[Cai03]		[Naph01]		Our approach	
Audio events	acclaim	98%	laughter	82.3%	explosion	86.8%	explosion	91.7%
	whistle	97.3%	applause	87.4%			gun	95%
	commentator speech	92.6%	cheer	92.6%			brake	57.1%
	silence	91.1%					engine	95.1%

3.7.2 Evaluation of Semantic Concept Detection

In semantic concept detection, the models based on HMM and SVM are evaluated respectively. As the basic analysis unit is one texture window, the metrics of recall and precision are calculated to show the detection performance, as shown in Table 3-3. We tested movie clips from "We Were Soldiers," "Windtalker," "The Recruit," "Band of Brother," etc., for gunplay and movie clips from "Terminator 3," "Ballistic: Ecks vs. Sever," "The Rock," "2 Fast 2 Furious," etc., for car-chasing. The detection performance is somewhat sequence-dependent because different movies posses different acoustic conditions. However, both the HMM-based and SVM-based approaches averagely achieve over 90% recall and near 70% precision in detecting gunplay and car-chasing scenes. These results show the promising achievement of the proposed fusion schemes.

Table 3-3. Average performance of semantic concept detection by (a) HMM and (b) SVM.

Semantic Concept	Recall (a)	Precision (a)	Recall (b)	Precision (b)
Gunplay	0.612	0.727	0.531	0.741
Car-chasing	0.697	0.731	0.661	0.702

Table 3-4. Some detailed results in semantic concept detection by (a) the HMM-based approach and (b) the SVM-based approach.

Semantic Concept		Recall (a)	Precision (a)	Recall (b)	Precision (b)
Gunplay	'We Were Soldiers'	0.88	0.75	0.859	0.832
	'44 Minutes'	0.98	0.95	0.98	0.813
	'Imposter'	0.982	0.659	0.965	0.567
Car-chasing	'Ballistic: Ecks vs. Sever'	0.99	0.83	0.98	0.839
	'2 Fast 2 Furious'	0.985	0.917	0.977	0.914
	'The Rock'	0.99	0.629	0.99	0.619

Due to various acoustic conditions, the detection performances vary in different sequences. The accuracy of semantic concept detection would degrade when bad audio event detection is involved. For example, in Table 3-4, the detection performance from two fusion schemes remains similar in the first two gunplay test sequences. However, the precision of "Imposter" degrades significantly while the corresponding recall is similar to "44 Minutes." The reason is that many people yelling, strong alarm sounds, and violent background music occur in the test audio clip. These sound effects are often mis-detected as explosion sounds and degrade the detection performance. Similar situations occur in the case of "The Rock" in car-chasing detection.

We further investigate how system performance varies with respect to different lengths of texture windows. The F1-metric, which jointly considers precision and recall, is used to indicate the system performance:

$$\mathrm{F1}=\frac{2\times \mathrm{Precision}\times \mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}}. \tag{3-8}$$

Figure 3-7 shows the relationship between average performance of the HMM-based approach and lengths of texture windows. It's clear that the proposed system works particularly well when the length of texture window is set as five or six seconds. In this work, we simply take five-second segments as the basic units for semantic concept detection.

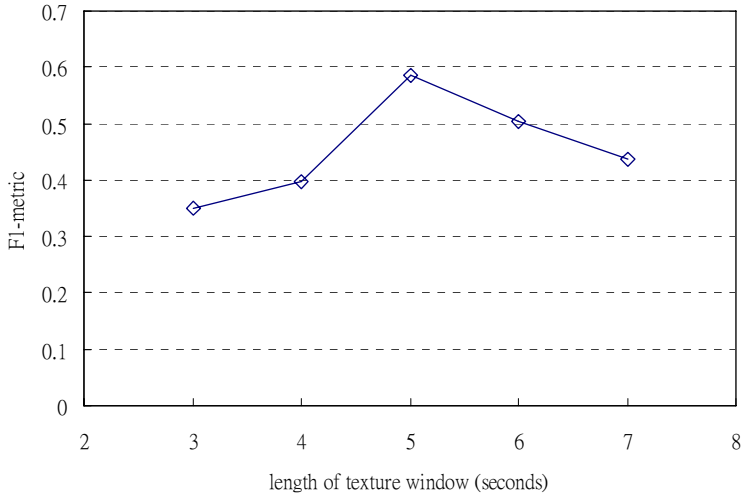


Figure 3-7. Relationship between lengths of texture windows and system performance.

3.7.3 Comparison with Baseline System

To show the superiority of the proposed framework, we compare the detection performance with that of the baseline case. The baseline system models semantic concepts directly by low-level features. For the semantic concept training data, the audio features described in Section 5-3 are first extracted. Then these features are modeled by HMMs rather than constructing pseudo-semantic features. The comparison demonstrates the difference between early fusion (baseline system) and late fusion (the proposed system) schemes [Snoe05]. In the experiment, the same training and testing data are used for the baseline system and the proposed framework.

Figure 3-8 illustrates the recall-precision curves of average detection performance. The proposed hierarchical framework shows its superiority over the baseline system. Because the baseline system doesn't take account of the information at event level, the precision rate degrades significantly as we increase recall. Linking the low-level features and high-level semantics by event fusion, i.e. the construction of pseudo-semantic features, provides a more robust performance in semantic concept detection.

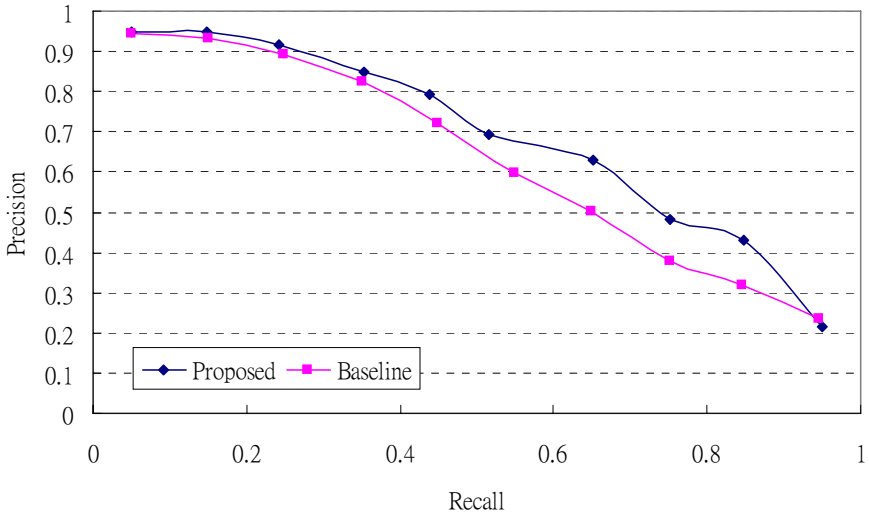


Figure 3-8. Comparison of the baseline and the proposed HMM-based approaches.

The results of semantic concept detection facilitate efficient semantic indexing for videos. With this help, we can develop a more efficient browsing interface for media accessing. Figure 3-9 shows a snapshot of a gunplay concept browsing interface, in which the curve displays the semantic confidence of each video segment. Although some false alarms exist, this visualized presentation helps users efficiently find what they want and facilitates browsing.

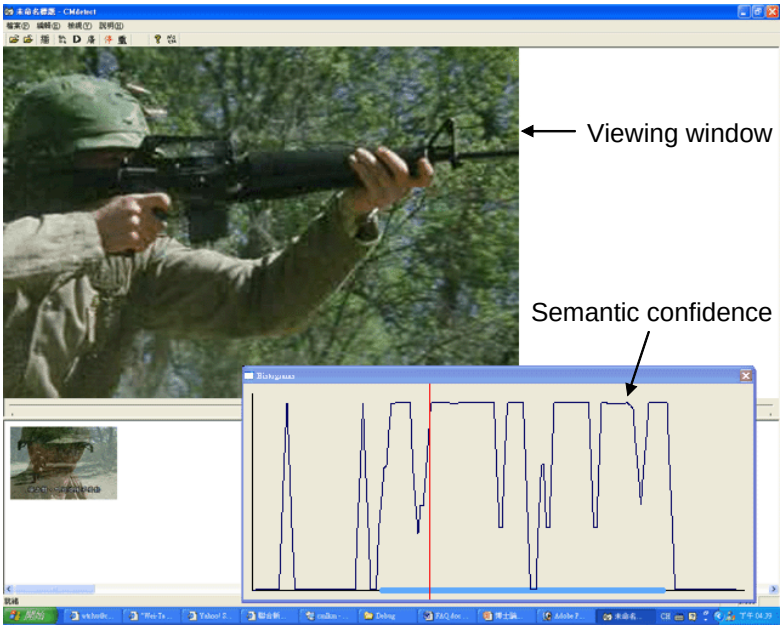


Figure 3-9. A snapshot of a semantic concept browsing system.

3.7.4 Discussion

Both the HMM-based fusion scheme and the SVM-based fusion scheme show their promising performance achievements. The most important advantage of event fusion approach is that event models can be trained separately, and new impacts from other events can be easily added to the framework. For example, more gunplay-related events such as helicopter-flying or people yelling can be modeled to augment the pseudo-semantic features.

Although the effectiveness of this work has been shown, some issues should be discussed more. The main reason of performance degradation is a) mixed audio signals and b) confused acoustic characteristics between different sounds. One example of the former case is the simultaneous occurrence of gunshot and explosion, while the bass environmental sound may be misclassified as an engine event because of their acoustic similarity. For the problem a), one of the solutions may be separating multi-source audio signals and analyzing them individually. The studies of independent component analysis [Hyva01] would provide new idea for this work. For the problem b), more acoustic features should be explored specifically for event modeling and discrimination.

3.7.5 Semantic Indexing Based on the Proposed Framework

This work presents a preliminary try to identify the concept of a semantic concept to facilitate multimedia retrieval. The results of semantic concept detection index videos with distributions of semantic concepts rather than occurrences of isolated events or objects. It provides the idea that concept-based indexing could be achieved by fusing the information of relevant events/objects. Although the proposed framework is only applied to action movies, it's believed to be generalized to other types of videos. Meanwhile, another encouraging idea of this work is the introduction of the late fusion of individual classifiers. Individual classifiers can be trained separately and added adaptively to the final meta-classifier. On the basis of this framework, different semantic concepts could be modeled and detected by taking account of various visual and aural events. For example, replacing audio event models by visual object models, visual semantic concept such as multi-speaker conversation could be modeled by the same framework. Results from different modalities can also be fused (by careful design of pseudo-semantic features) to construct a multi-modal meta-classifier. Hence the proposed framework can qualify general semantic indexing tasks.

This work realizes the idea of the semantic concept framework described in Chapter 2 and is a systematic approach to deal with concept detection in movies. Figure 3-10 shows the correspondence between the implementation and the general semantic concept detection framework. We select statistical pattern recognition

techniques to build detectors of audio events. After evaluating the confidences from different audio events, the mid-level representation which embeds the idea of combining classifiers is constructed. In other words, the detection results that are from simple and dedicated classifiers are combined to characterize complex and generic concepts. Two types of statistical techniques, i.e. discriminative and generative approaches, are applied to model this mid-level representation. Finally, after characterizing features, objects, and semantics, high-level concepts such as gunplay and car-chasing scenes can be detected automatically.

Although only aural information is used in this work, detectors based on visual information can also be built and their results can be combined to the semantic context models. Moreover, specific to a semantic concept, the production rules is implicitly embedded in the process, such as gunshot and explosion sounds are related to gunplay scenes. More domain knowledge may be applied to facilitate reasonable and accurate semantic concept detection in different types of videos. In this work, filmmaking rules and some ideas from media aesthetics are exploited to relate semantic concepts with audio events. The ideas of computational media aesthetics are survey in Appendix C.

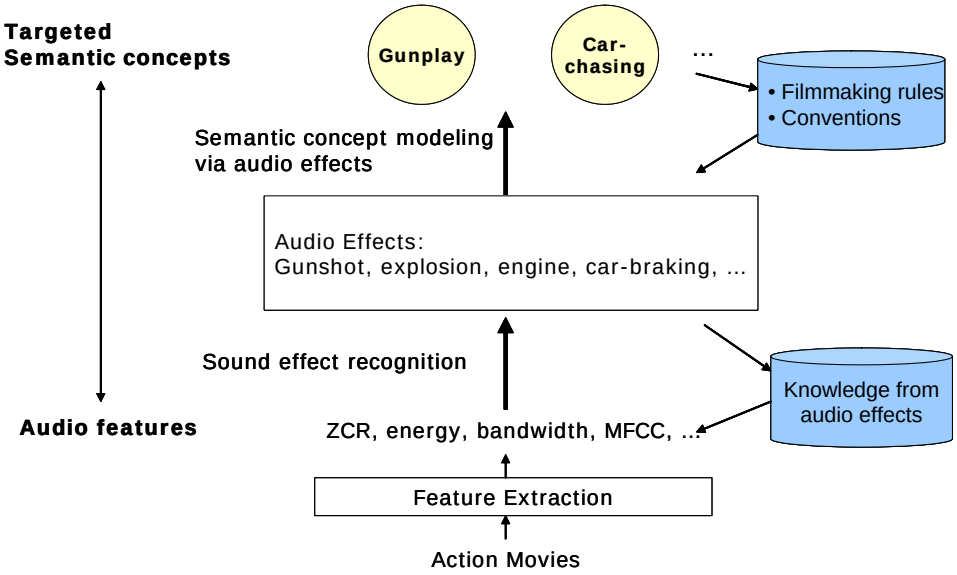


Figure 3-10. Audio semantic context detection in terms of the semantic concept detection framework described in Chapter 2.

3.8 Summary

We present a hierarchical approach that bridges the gaps between low-level features and high-level semantics to facilitate semantic indexing in action movies. The proposed framework hierarchically conducts modeling and detection at two levels:

audio event level and semantic concept level. After careful selection of audio features, HMMs are applied to model the characteristics of audio events. At the semantic concept level, generative (HMM) and discriminative (SVM) approaches are used to fuse pseudo-semantic features obtained from the results of event detection. Experimental results demonstrate a remarkable performance of the fusion schemes and signify that the proposed framework draws a sketch for constructing an efficient semantic indexing system.

The proposed framework can be extended to model different semantic concepts. It may be necessary to consider different combinations of events or include visual information according to the production rules of targeted films. Another possible improvement may include the elaborate feature selection by developing an automatic feature induction mechanism or applying the techniques of blind signal processing to deal with the problem of mixed audio effects.

Chapter 4
Semantic Analysis and Game Abstraction in Baseball Videos

4.1 Introduction

Sports video analysis has attracted much attention due to its potential commercial benefits. Various sports games that follow different rules and broadcasting characteristics draw different issues in video analysis. Recently, researchers have developed technologies and applications from different aspects [Yu05]. The most popular sports such as soccer [Ekin03, Leon04, Wang04-1, Wang04-2, Xie04, Xu04, Yu03], American football [Baba04], basketball [Nepa01], and baseball [Arik03, Han02, Hua02, Rui00, Shih03, Xion03, Zhan02, Zhon04], are widely studied. To the end of providing efficient media access and entertainment functionalities, scene classification [Hua02, Zhon04], concept detection [Han02, Nepa01, Shih03, Wang04-1, Xion03, Xu04, Xu03, Zhan02, Zhon04], highlight extraction [Arik03, Baba04, Bert05, Rui00], replay generation [Wang04-2], or game summarization [Ekin03, Li04, Tjon04] have been developed.

Although many studies were proposed to analyze sports video, most of previous works thoughtlessly ignore the real needs of sports audiences, who are the receivers and should be the judgers of analytical results. Generally, a sports fan wishes to know "what really happened in this game?" or "how about my favorite player's performance?" For those who don't have time to see the whole game, a game summary or highlight that consists of the most informative plays or exciting parts are attractive. As the famous remark "record is the life of a player" says, the requirement of practical sports video analysis is to accurately detect "what kind of concept occurs," "when and how a concept occurs," and "who did it." Explicitly knowing game details is the key factor to make summaries and highlights valuable and reasonable.

Starting from the demands of sports fans, we survey sports analysis techniques in terms of "explicitness" and "comprehensiveness." Explicitness means whether sports concepts can be exactly detected, such as a "double" in a baseball game or a "three-pointer shot" in a basketball game. Comprehensiveness means whether (almost)

all types of concepts can be detected. For example, thirteen baseball concepts are defined in Japanese and Taiwanese baseball leagues, and eight common concepts (goal, shot, foul, corner kick, offside, yellow card, red card, and save) are used in soccer games. To clarify the novelty and contribution of our work, we remark these issues as follows:

- Although baseball concept detection has been pursued for years, most of them are either not explicit or comprehensive enough. Zhang and Chang [Zhan02] proposed a concept detection method based on caption information, but they only focused on detection of the last pitch and scoring. Han et al. [Han02] developed a baseball digest system based on maximum entropy method and detected seven baseball concepts. Nonetheless, the detection performance is not very promising in the reported results. Other works such as [Shih03] also tackles baseball concept detection, but most of them didn't think over how to explicitly detect and recognize (almost) all concepts in baseball games. In our work, we deliberate upon detecting all concepts from the viewpoint of offensive side and therefore have full understanding of the game content.
- As compared to other popular sports, such as soccer, tennis, and basketball, more types of concepts take place in baseball games. That makes achieving comprehensiveness even harder. Moreover, currently explicitness is not easy to be accomplished in other sports video analysis. For example, whether a "fade-away shot" or "dunk" causes a score is hard to be discriminated, but a basketball fan often likes to see Michael Jordan's fade-away shot or Vince Carter's dunk. We concentrate our work on baseball games and elaborately exploit rules and visual information in concept detection.
- The ultimate goal of sports video analysis is to provide users practical applications or well-organized information. Therefore, we should turn academic works into realistic applications and evaluate performance by comparing with man-made results or conducting subjective tests.

In this chapter, we accentuate our works by carefully tackling with explicitness and comprehensiveness. A systematic framework that comprises reliable shot classification, explicit concept detection, and extended applications is proposed. We summarize these processes as follows:

- Reliable shot classification: Color and geometric information are exploited to classify shots into several canonical views. To reliably perform shot classification in different situations (different stadiums, time, or broadcasting channels), several methods to dynamically detect field color and pitcher

position are proposed.

- Rule-based concept detection and model-based concept detection: Official baseball rules are transformed into an efficient rule-based detection module. For the concepts that cannot be discriminated by simply using baseball rules, model-based detection module is further developed based on elaborately designed game-specific features.
- Extended applications: On the basis of explicit concept detection, attractive applications like the ones provides by MLB.com [MLB06] can be automatically developed. Accompanying with audio cues and inherent importance of concepts, more enjoyable game highlight or summarization could be made.

4.2 System Framework

4.2.1 Characteristics of Baseball Games

An important observation in baseball videos is that all concepts occur between two consecutive pitch shots. Thus the status changes within this duration give us important clues to indicate what happened in games. The progress of a typical concept is: 1) the pitcher releases the ball; 2) the batter hits out the ball; 3) the ball is caught by a fielder (field out) or falls on the ground (hit); 4) a fielder returns the ball to the infield; 5) the camera switches to the pitch view and the pitcher prepares the next pitch. Figure 4-1 illustrates two examples of the game progress. There may be no (duration (a)) or one (duration (b)) concept between two consecutive pitch shots. In broadcasting baseball videos, the status changes between two consecutive pitch shots can be detected by checking the caption information, including number of score, number of out, and base-occupation situation.

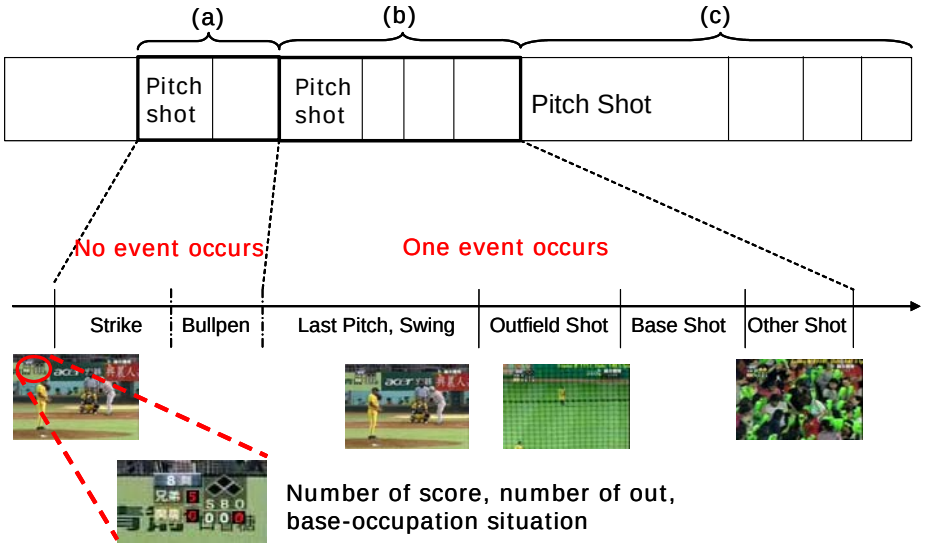


Figure 4-1. Examples of the game progress.

4.2.2 Overview of System Framework

Figure 4-2 shows the system framework, which includes three stages of processes: shot classification, concept detection, and extended applications. The target of shot classification stage is to classify video shots into pitch, infield, outfield, or other views. To accommodate to different broadcasting situations, an adaptive field color detection module and pitcher detection module are developed to dynamically extract color and geometric information and to facilitate shot classification.

In the concept detection stage, we specially extract caption information in pitch shots. According to baseball rules, the rule-based decision module infers what happened based on the information changes on the caption. However, some concept pairs such as “strikeout vs. field out” cannot be discriminated by simply using rules. For these “confused concepts,” we further develop classifiers based on visual and speech information. The visual classifiers characterize shot transition information and return an opinion with some confidence score by giving a testing video clip. On the basis of speech information, we exploit a key-phrase spotting module to evaluate the confidence of a specific concept. The opinions from visual and speech information are finally fused to facilitate confused concepts discrimination. After these processes, thirteen different concepts in baseball games are explicitly uncovered.

With the aid of explicit concept detection, practical and accurate applications can be automatically developed. To generate more elaborate game abstraction, we consider both the “informativeness” (content coverage) and “enjoyability” (perceptual quality) [Ngo05] in the summarization and highlight selection process. We design summarization and highlight selection algorithms to produce game abstraction that better matches fans’ need and expectation.

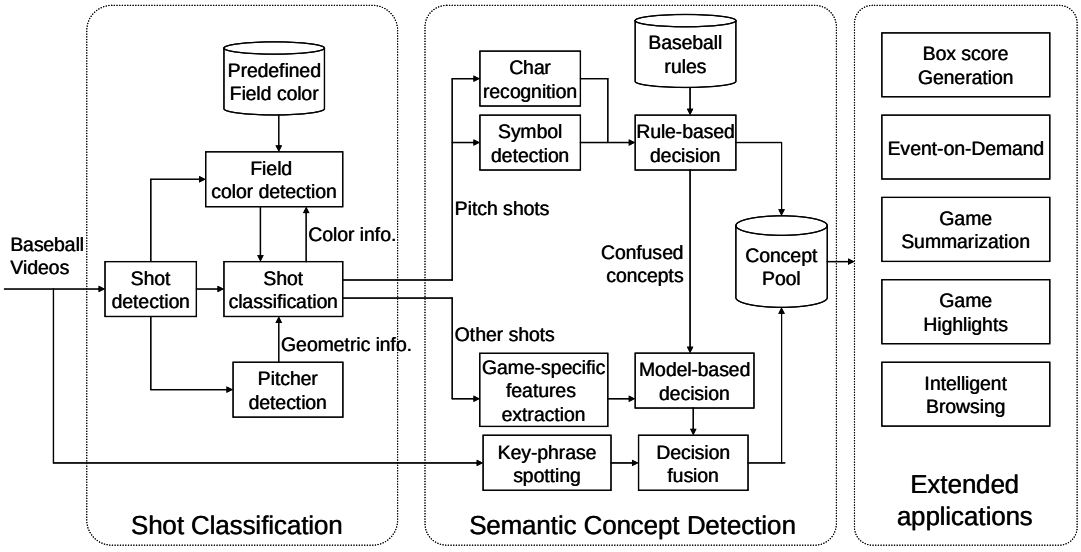


Figure 4-2. System framework of explicit concept detection and its applications.

4.3 Shot Classification

4.3.1 Procedure of Shot Classification

Figure 4-3 shows the process of a color-based shot classification module. Color ranges of the field, including grass and soil, are adaptively determined by a field color determination method. With the field color definition, we compute the ratio of field area to the keyframe of each video shot. To avoid some noises derived from gradual shot transition, the tenth frame from the starting of a shot is selected as its keyframe. Two thresholds, *t1* and *t2* (*t1* < *t2*), are defined for shot classification. The steps of classification are:

(a) If the field ratio (FR) is less than the threshold *t1*, the corresponding shot significantly differs from the field and is classified as "other" view. Typical examples include audience shots or commercial shots.

(b) If the field ratio is larger than the threshold *t2*, the corresponding shot is like the field. Based on edge information, an infield/outfield classification module is further developed to distinguish between infield and outfield views.

(c) If the field ratio is between *t1* and *t2*, the corresponding shot is first verified by a pitch shot detection module. If the frame still doesn't conform to the definition of a pitch view, it is further verified by the infield/outfield classification module. Finally, each shot is classified as a pitch, infield, outfield, or other view.

To derive the thresholds *t1* and *t2*, we gather the statistics of field color ratios from three games and construct their distributions for each canonical view. We model these distributions as Gaussians and find the classification boundaries according to the Bayesian theory [Duda01]. They are finally set as 0.1 and 0.48. Note that although different stadiums or different TV channels bring about significant changes in field color, the presentation of these canonical views is very similar. Therefore, we can feel free to set these thresholds after observing several games.

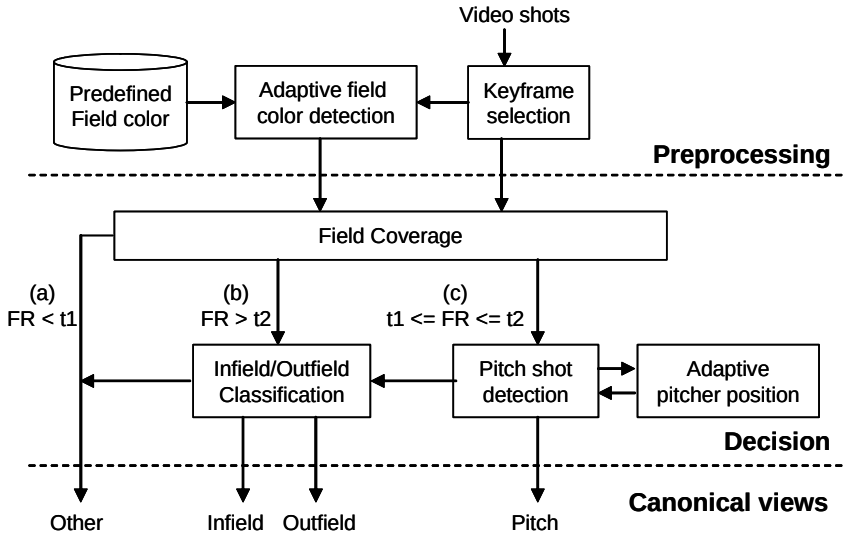


Figure 4-3. Diagram of shot classification.

4.3.2 Adaptive Field Color Determination

In baseball games, field region dominantly occupies the bottom part of video frames. To reliably classify shots, we develop a module to examine the bottom part of video frames and dynamically determine dominant colors. As the game proceeds, this module is triggered periodically to analyze a video clip and determines the latest color range of the field. In our implementation, it acts on every twenty minutes and analyzes three-minute (3-min) video clips.

All procedures of shot classification are performed in the HLS color space. Three color channels are respectively quantized into thirty equal-interval bins. For each color channel, an integrated histogram is constructed based on the color of pixels in a 3-min video clip (about 5400 frames). We check the integrated histogram and compute the percentage of each bin. If the histogram value is larger than ten percent of total value, the corresponding color range is viewed as the field color. Dominant colors often fall into two ranges, because the baseball field consists of grass and soil.

Note that the assumption of this process is to determine field color via dominant color detection. However, in real broadcasting videos, cameras often switch to the

audience or players, or commercials are inserted at inning changes. To remove the influence of these irrelevant shots, we define a default color range of field at the beginning of each game. For the bottom part of each frame, we check whether more than forty percent of pixels are “suspected” field pixels (by the default color definition). Only the frames with enough suspected field pixels are processed in the color determination module. The newly determined color range then updates the default field color definition. With the definition of field color, the processes in the following subsections can then proceed.

4.3.3 Infield/Outfield Classification

For the keyframe that is largely occupied by field (Figure 4-3(b)), it is further classified as infield or outfield. The outfield view often contains audience or stadium artifacts and displays high-texture content. Therefore, we use color adjacency histogram [Lee03] to represent edge information and distinguish between infield and outfield views. In our work, the difference between infield and outfield views doesn’t affect the performance of concept detection, but it may help in game highlight extraction.

4.3.4 Pitch Shot Detection

For the keyframe whose field ratio is between two thresholds (Figure 4-3(c)), the spatial layout of field pixels is checked through its horizontal and vertical profiles, as shown in Figure 4-4. If this keyframe is a pitch view, the field pixels should concentrate only on the bottom part of horizontal profile. On the other hand, because the pitcher is always in the left part of a pitch shot, we can find a valley in the vertical profile. To alleviate the slight differences between pitch shots in different TV channels, we define a sliding window of 50 pixels with 25 pixels overlapped to go through the left part of the vertical profile. Note that the video resolution in this work is 352×240. If there exists a range whose profile value is less than a threshold (a valley exists), the keyframe is declared to be a pitch view.

Although the field ratio of the case in Figure 4-3(c) is not as high as that in Figure 4-3(b), it is also possible to be a field view. The camera may track the ball on the air and doesn’t capture large part of the field region. Therefore, if no pitcher is detected in the keyframe, it is further confirmed by the infield/outfield classification module.

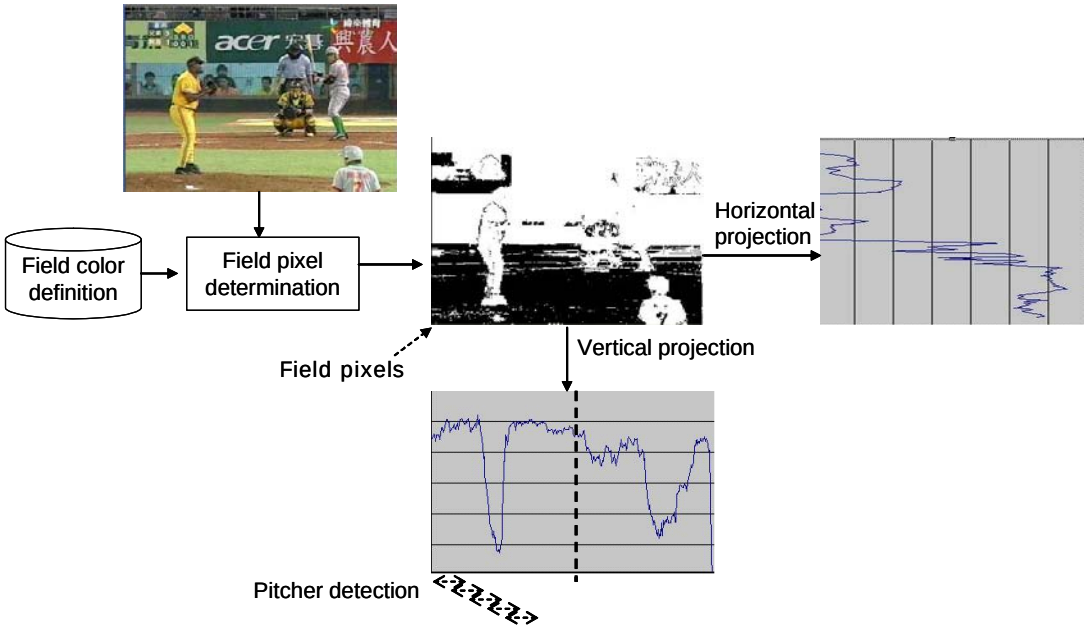


Figure 4-4. Pitch shot detection by field pixel profiles and pitcher detection

4.4 Concept Detection

Conventional baseball video analysis focuses on adopting visual or aural features to speculatively identify the highlighted parts. In this work, we emphasize that we can elaborately perform fine-granularity concept analysis. For example, if no base is occupied in the *i*-th shot, and the score increases by one but still no base is occupied in the (*i*+*1*)-th shot, we can infer that a home run (actually a "solo home run") occurs between these two shots. However, some concept pairs such as "single vs. walk" and "strikeout vs. field out" cannot be discriminated by simply using the rule-based decision method. We further take the contextual shot information into account and propose a model-based decision method to strive for explicit concept detection. The most important contribution of this work is that we propose a systematic method to seamlessly integrate baseball rules (domain knowledge) with audiovisual features (computational characteristics) and thoroughly explore semantic concepts in baseball games.

4.4.1 Rule-based Concept Detection

In broadcasting videos, informative caption data include "number of out," "number of score," and "base-occupation situation." Each effective baseball concept leads to changes of this information, such as "homerun" increases the score, "strikeout" increases out, and "hit" or "walk" change the base-occupation situation. Therefore, we can simply check the information changes on the caption and accomplish efficient implementation for concept detection.

4.4.1.1 Caption Feature Extraction

Caption information is often displayed as two types: text (such as number of score) or symbol (such as number of out and base occupation). For text information extraction, three steps are included:

- Character pixel determination: Characters often have higher intensity value as compare to the background. The pixel that has high intensity is viewed as a character pixel.
- Construct character template vectors: Given a region, the character pixels are first determined, and a 13-dimensional Zernike moment [Khot90] is extracted to represent their characteristics. For each number, e.g. two, we collect a 30-second video clip (about 900 frames) as training data. The Zernike moments extracted from a specific region in video frames are then averaged to construct the character template (a template vector).
- Character recognition: Given a test vector, it is compared with all trained character templates in terms of vector angle. The test vector is recognized as *i* if it has the smallest included angle to the *i*th template vector.

For symbol information, we just employ the intensity-based approach similar to character pixels segmentation. In the pre-indicated region, the base-occupation situation is displayed according to whether the corresponding base is highlighted or not.

In the duration between two consecutive pitch shots, the changes of number of out, number of score, and base-occupation situation are jointly considered in concept detection. They are:

- $o_{i,i+1}$, the difference of outs between the *i*th and the $(i+1)$th pitch shots, where $o_{i,i+1} \in \{0, 1, 2\}$. We don't deal with the situation of $o_{i,i+1} = 3$ because, in almost all TV channels, commercials are instantaneously inserted when three batters are out and the status resets to zero at the next inning.
- $s_{i,i+1}$, the difference of scores between the *i*th and the $(i+1)$th pitch shots, where $s_{i,i+1} \in \{0, 1, 2, 3, 4\}$. The case of $s_{i,i+1} = 4$ denotes the occurrence of a home run with four scores (the so-called "grand slam").
- b_i and b_{i+1}, the base-occupation situations in the *i*th and the $(i+1)$th pitch shots, where b_i and $b_{i+1} \in \{0, 1, ..., 7\}$. The number of occupied bases at these two shots (n_i and n_{i+1}) are calculated. To catch the difference of base-occupation

situation between two pitch shots, the value of $b_{i,i+1}$ (= b_{i+1} - b_i) is also considered. The meanings of feature values of b_i and n_i are listed in Table 4-1.

Table 4-1. Physical meanings of different base-occupation situations.

b_i	n_i	**Physical meaning**
0	0	No base is occupied.
1	1	Only the first base is occupied.
2	1	Only the second base is occupied.
3	2	Both the first and the second bases are occupied.
4	1	Only the third base is occupied.
5	2	Both the first and the third bases are occupied.
6	2	Both the second and the third bases are occupied.
7	3	All bases are occupied.

4.4.1.2 Feature Filtering

The features described above are concatenated as a vector $f_{i,i+1}$ to represent the game progress. However, many cases are illegal in baseball games. We should filter out the illegal features and identify the concepts implied by legal features.

When a concept occurs, there may be one or no batter reaching a base, and the runners (the players who occupy bases) would be still at bases or out or reach the home plate to get scores. Therefore, when a legal concept is invoked by a batter, one of the three situations might take place:

1) The batter is out, whether he suffers strikeout, field out, or touch out. This case contributes one to $o_{i,i+1}$.
2) The batter reaches a base, but no other runners are capable of reaching the home plate to get scores. This case contributes one to the number of occupied bases ($n_{i,i+1} = n_{i+1} - n_i = 1$).
3) The batter reaches a base, and some runners reach the home plate to get scores. No matter how many runners getting scores, $n_{i,i+1} + s_{i,i+1} = 1$. For example, assume that the second and the third bases are occupied in the ith pitch shot. The batter hits a double and reaches the second base, and both two runners reach the home plate to get two scores. The information change is $(n_{i+1} - n_i) + s_{i,i+1} = (1\text{-}2) + 2 = 1$.

According to these observations, a general decision rule for legal features can be mathematically expressed as:

$$f_{i,i+1} = \begin{cases} legal, & \text{if } (n_{i,i+1} + s_{i,i+1} + o_{i,i+1}) = 0 \text{ or } 1, \\ illegal, & \text{otherwise.} \end{cases} \quad (4\text{-}1)$$

The value of ($n_{i,i+1}$ + $s_{i,i+1}$ + $o_{i,i+1}$), denoted as $\alpha_{i,i+1}$, indicates whether the batter changes or not. If the value is 0 (no batter changes), nothing happened between the ith and the ($i+1$)th pitch shots. If the value is 1 (one batter changes), the batter is out or reaches some base, and a new batter comes at the ($i+1$)th pitch shot.

Furthermore, according to the baseball rules, no runner can go back to the previous base. We check the base-occupation situations in two consecutive pitch shots and filter out this kind of illegal features. For example, it would not happen if b_i = 2 and b_{i+1} = 1 in case of $s_{i,i+1}$ = 0 and $o_{i,i+1}$ = 0. (It's impossible that the occupied base is back in case of no score and no out.)

4.4.1.3 Concept Identification

Given a legal feature vector, we can view the process of concept identification as classifying it into a subset, which represents one baseball concept. The given feature vector is first classified as one of the four types of concepts by checking whether the batter changes ($\alpha_{i,i+1}$ = 0 or 1) and whether the number of out ($o_{i,i+1}$) increases. The baseball concept taxonomy is illustrated in Figure 4-5. Thirteen concepts are considered in this work: single (1B), double (2B), triple (3B), homerun (HR), stolen base (SB), caught stealing (CS), field out (AO), strikeout (SO), base on ball (Walk, BB), sacrifice (SAC), sacrifice fly (SF), double play (DP), and triple play (TP). Although they still don't cover all concepts in baseball games, they explicitly state what happens in a game and greatly expand the visibility of baseball videos.

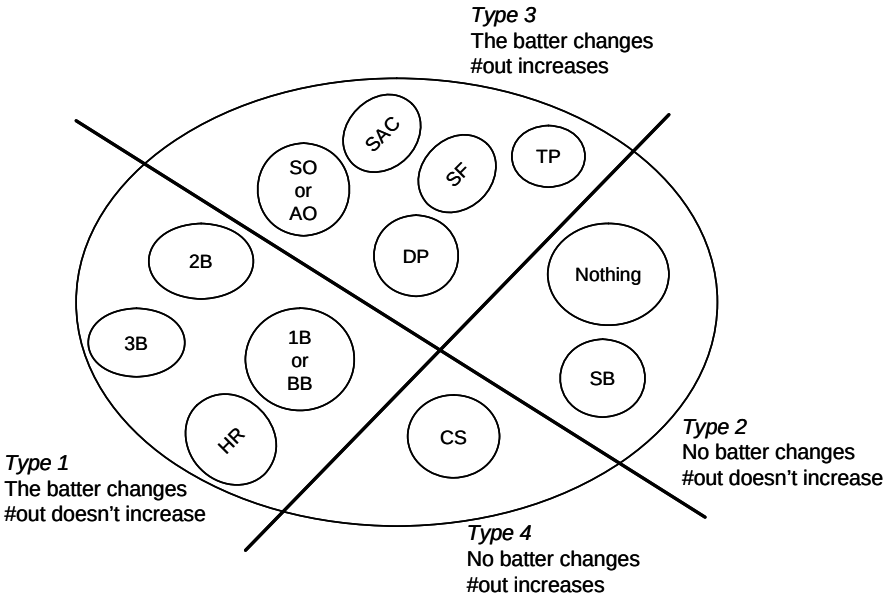


Figure 4-5. Taxonomy of baseball concepts.

The concept identification process can be conceptually modeled as a decision tree, as shown in Figure 4-6, and concepts are determined at leaves after tracing the tree. The general idea of tracing this tree can be described as follows:

(1) First, we check $o_{i,i+1}$ to detect whether the unknown concept causes an out or not.

(2) Second, according to $n_{i,i+1}$+$s_{i,i+1}$, we detect whether a new runner occupying a base or someone scores.

(3) Then we check base-occupation situation (b_{i+1} and $b_{i,i+1}$) to determine what really happened.

Note that the concept of triple play (TP) is a special case and is not included in Figure 4-5. It's a very unusual concept and is detected by a heuristic rule that is beyond the constraint of feature filtering in equation (4-1):

> If more than two bases are occupied and nobody out in the ith pitch shot, and nobody out, no score changes, and no base is occupied in the $(i+1)$th pitch shot ($o_{i,i+1}$=0, $n_{i,i+1}$=-2 or -3, $s_{i,i+1}$=0), a triple play would occur.

The rule-based process effectively detects most concepts by employing information changes on caption. However, some concept pairs such as "single vs. walk" and "strikeout vs. field out" lead to the same information changes on caption and cannot be explicitly discriminated by simply using rules. In baseball games, these kinds of "confused" concepts can be categorized as four types, as shown in Table 4-2. To make the concept detection process more explicitly, we develop a model-based

approach that adopts contextual shot information and elaborate the detection results. We primarily deal with the cases of "single vs. walk" and "strikeout vs. field out" because other confused situations rarely happen.

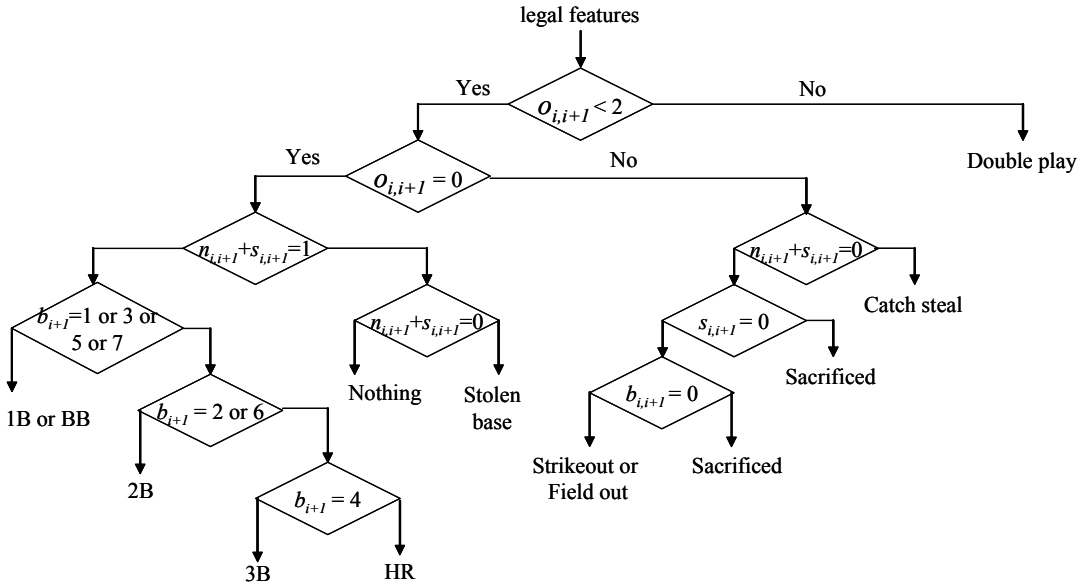


Figure 4-6. Concept detection process on decision tree.

Table 4-2. Confused concept in baseball games

Confused concepts	**Information changes between two consecutive pitch shots**
Single and walk (base on ball, hit by pitch, or intentional walk)	The first base is occupied and no out increases.
Strikeout and field out	The number of out increases one and the number of occupied bases and score don't change.
Stealing, wild pitch, passed ball, and balk	No out increases and the runner advances to the next base.
Caught stealing and pickoff	The number of out increases one and the number of occupied bases decreases.

4.4.2 Model-based Concept Detection

The contextual information of shot transition and its temporal duration often provide clues for concept identification. For example, as a single occurs, the camera switches to the field to show the action of fielder. On the other hand, as a base on ball occurs, close-up on the pitcher or the batter is often displayed to show their facial expression. In the model-based concept detection, we jointly consider information of shot transition, temporal duration, and motion magnitude for discriminating concepts that

are implicitly hidden after rule-based concept detection.

4.4.2.1 Shot Context Features

According to the observation of broadcasting style and baseball rules, we propose the following features to describe concept characteristics. Note that these features are extracted within the duration from the end of previous effective concept to current pitch shot, as shown in Figure 4-7.

- *ConsecutivePF*: indicating whether a field view displayed immediately after the last pitch view. If the batter hits out the ball, this kind of shot pair occurs and indicates higher probability of the occurrence of "single" or "field out." In Figure 4-7, the last pitch view is at the third shot, and the shot pair occurs at the third-fourth shots to indicate *ConsecutivePF* = 1.

  The first field shot right after the last pitch shot plays an important role in extracting shot context features and is particularly defined as the *pivot shot*. If there is no field shot within this duration, the last shot of this duration is defined as the pivot shot.
- *PitchBeforeFieldView*: indicating how many pitch views before the pivot shot. In general, more pitch shots occur before the pivot shot in the concepts of "walk" and "strikeout," because the pitcher has to pitch at least four or three balls before they take place. In this example, the batter hits the ball at the second pitch (Figure 4-7(3)), and therefore, *PitchBeforeFieldView* = 2.
- *DiffPitchField*: indicating the time difference between the last pitch shot and the pivot shot. If the batter doesn't hit out the ball, i.e. *ConsecutivePF* = 0, *DiffPitchField* is often larger in "walk" and "strikeout" cases than that in "single" and "field out" ones.
- *FieldDuration*: indicating the time duration of the pivot shot. When the ball is hit out, the duration of field shot is often short because the fielder should deal with the ball as soon as possible to prevent extra base hit. In Figure 4-7, *FieldDuration* = 1237-1151 = 86 frames.
- *Motion*: indicating the motion magnitude of the pivot shot. When the ball is hit out, the camera tracks the ball or the fielder and demonstrates higher motion. Therefore, higher motion is often derived from "single" or "field out" concepts, and lower motion is derived from "walk" or "strikeout" cases.

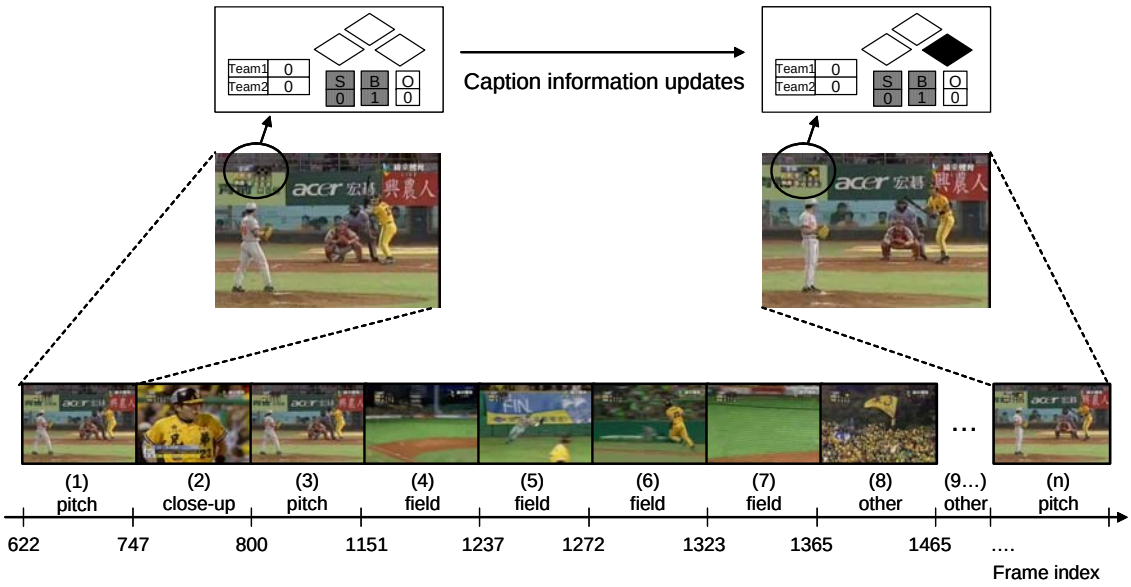


Figure 4-7. An example of shot context feature extraction.

4.4.2.2 Modeling

All the shot context features are normalized to the range [0,1] before training or testing. We manually selected twenty training sequences, ten of them are "single" and another ten sequences are "walk," from the same TV channel to construct a "single-walk" classifier. K-nearest neighbor modeling is implemented for each classifier due to its simplicity. Through the rule-based decision described in previous sub-section, the sequences decided as "single or walk" are further discriminated by the classifier. The shot context features obtained from the suspected sequence are then classified as a "single" or "walk" concept by the *k*-nearest neighbor algorithm. The same process is applied to detect "field out" or "strikeout." In this work, *k* is set as 8 for classification accuracy and efficiency.

4.4.3 Combine Visual Cues with Speech Information

4.4.3.1 Overview

In the aforementioned work, we employ caption data and shot transition information to infer what happened in baseball games. Although this system works well in most situations, its performance in discriminating confused concepts is still not good enough.

Commentator's speech, which completely states the game progress, plays an important role for audiences to realize the game status. Therefore, it's attractive to exploit a speech recognition module and facilitate concept detection through speech information. We apply a key-phrase spotting module that maps speech signal to limited number of key-phrases, which provide some clues to the occurrence of some

effective concepts or actions, such as hit, out, and catch.

Figure 4-8 shows the fusion scenario that combines visual and speech information for concept detection. It consists of concept detection and confidence calculation from visual and speech perspectives and the integrated decision module. From visual data, rule-based methods and model-based methods [Chu05-4] identify what concepts occurring and where their boundaries are. Based on these concept boundaries, a key-phrase spotting module is applied to spot what key-phrases the commentator has spoken, which may provide clues for identifying what really happened in specific intervals. The concepts detected from visual and speech data are described as visual concepts and speech concepts for convenience. The confidences of visual and speech concepts are estimated respectively to be the bases of integrated decision. Based on the strategy of combining classifier decisions [Kitt98], we find the consensus from two modalities and make an integrated decision.

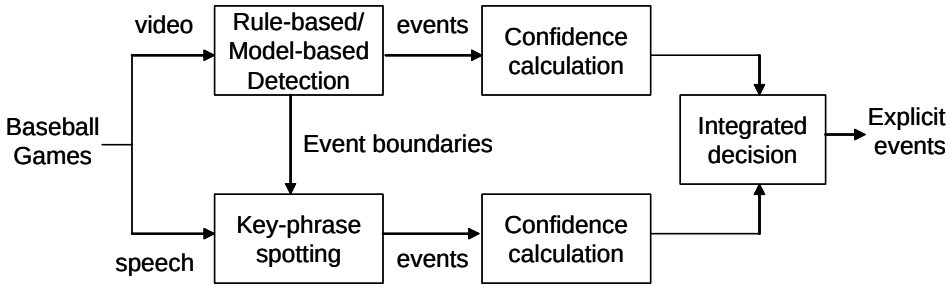


Figure 4-8. The scenario that fuses visual and speech information.

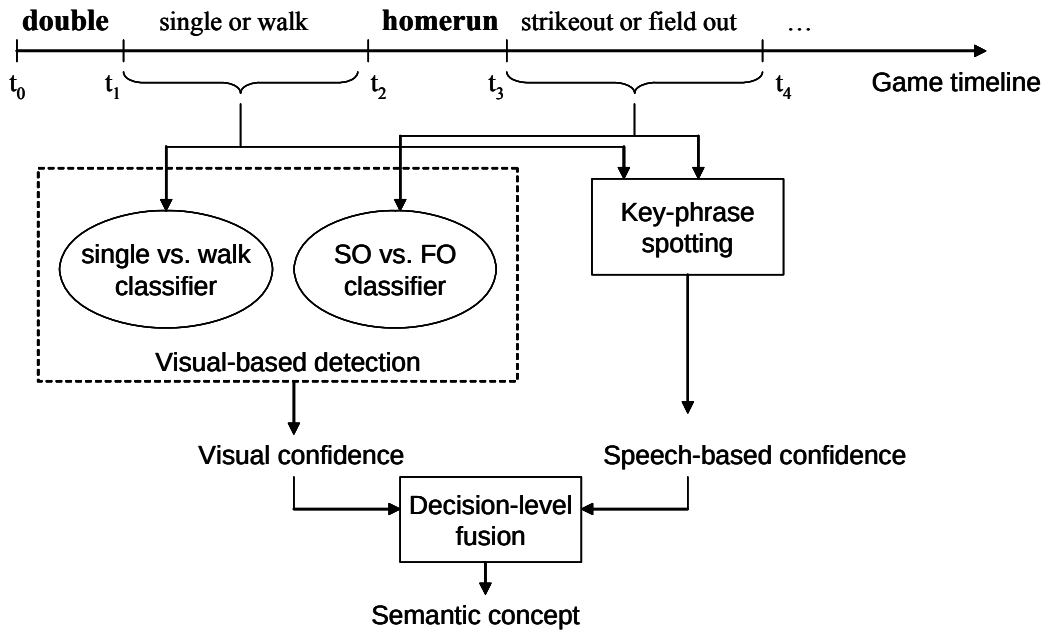


Figure 4-9. Examples of visual and speech concept detection.

Because the commentator not only speaks the concepts just occurred but also review the player's past performance or chat with others, recognizing all his speech derives too many noises in concept detection. In this work, we mainly appeal to key-phrase spotting module to facilitate confused concept discrimination. Only the durations of the occurrence of confused concepts, such as ranges t_1 to t_2 and t_3 to t_4 in Figure 4-9, are applied with key-phrase spotting.

We exploit the key-phrase spotting system developed in [Chen98], which is capable of extracting salient key-phrase fragments from an input utterance in real-time. High degree vocabulary flexibility and recognition accuracy can be achieved for any designate task. In this work, we define the mapping between baseball concepts and commonly used key-phrase in advance, as shown in Table 4-3. A specific speech concept is identified if one or more of its corresponding key-phrases are recognized in the designated duration. For example, if the phrase "touch out" is recognized in the case of "strikeout vs. field out" confusion, the occurred concept claimed by speech information is "field out" rather than "strikeout."

Table 4-3. Mapping between concepts and conventional key-phrases (in Mandarin Chinese).

Concepts	**Corresponding Key-phrases**
Single	R_1={安打(hit), 一壘安打(single)}
Walk	R_2={觸身球(hit by pitch), 保送(walk), 四壞球(four balls)}
Strikeout	R_3={三振(strikeout), 三振出局(strikeout)}
Field out	R_4={刺殺('touch out' or 'out before reaching bases'), 接殺(catch out)}

Although the key-phrase spotting module is now only applied to recognize Chinese, it is capable to be extended to other languages. The same framework, including visual and speech concepts detection, is general for any baseball game.

4.4.3.2 Information Fusion

After detecting concepts from visual and speech data, the problem narrows to making the final decision according to the detected results. It's a trivial task if both the opinions from video and speech are identical. For example, "strikeout" is surely the final answer if both visual and speech concepts are claimed as "strikeout." However, because both visual and speech concept detection modules are not perfect, it's often that the opinions from different modalities conflict. Therefore, we define and evaluate the confidence of two opinions and make the final decision.

(1) Confidence of visual-based detection

In constructing two classifiers that discriminate single from walk and strikeout from field out, visual information including pitch-field pattern, field shot duration, motion, and etc., are used as the feature vectors [Chu05-4]. K-nearest neighbor modeling is used to construct these classifiers. We derive the posterior probabilities to be the confidence of visual concepts.

Let the feature vector from visual data be $\mathbf{x}_1$, and K_1 (K_2) be the number of patterns among $\mathbf{x}_1$'s K nearest neighbors that belong to class C_1 (C_2). The estimated posterior probabilities [Cove67] are given by

$$P(C_1|\mathbf{x}_1)=\frac{K_1}{K} \quad \text{and} \quad P(C_2|\mathbf{x}_1)=\frac{K_2}{K}, \tag{4-2}$$

where $K_1+K_2=K$, and thus $P(C_1|\mathbf{x}_1)=1-P(C_2|\mathbf{x}_1)$.

With the K-nearest neighbor classifier that classifies classes C_1 and C_2, a test vector $\mathbf{x}_1$ is assigned to class C_1 if $K_1>K_2$, with the confidence value $P(C_1|\mathbf{x}_1)$.

(2) Confidence of speech-based detection

The confidence of speech concept is represented by "the posterior probability of the concept C_i occurs given the recognized key-phrases." Similar to visual-based detection, the recognized key-phrases are viewed as feature vectors. In the case of "single vs. walk" confusion, the feature vector from speech data $\mathbf{x}_2$ may be constructed only by the key-phrases relevant to single ($\mathbf{x}_2=R_1$), only by the key-phrases relevant to walk ($\mathbf{x}_2=R_2$), or both ($\mathbf{x}_2=R_1,R_2$). Definitions of the key-phrases R_1~R_4 are in Table 4-3. Considering these three cases, the posterior probabilities are estimated as:

Case 1:

$$P(C_1|\mathbf{x}_2=R_1)=\frac{\#(C_1)}{\#(\text{only the key-phrases in } R_1 \text{ are recognized})},$$

$$P(C_2|\mathbf{x}_2=R_1)=\frac{\#(C_2)}{\#(\text{only the key-phrases in } R_1 \text{ are recognized})}.$$

Case 2:

$$P(C_1|\mathbf{x}_2=R_2)=\frac{\#(C_1)}{\#(\text{only the key-phrases in } R_2 \text{ are recognized})},$$

$$P(C_2|\mathbf{x}_2=R_2)=\frac{\#(C_2)}{\#(\text{only the key-phrases in } R_2 \text{ are recognized})}.$$

Case 3:

$$P(C_1 \mid \mathbf{x}_2 = R_1, R_2) = \frac{\#(C_1)}{\#(\text{key-phrases in } R_1 \text{ and } R_2 \text{ are recognized})},$$

$$P(C_2 \mid \mathbf{x}_2 = R_1, R_2) = \frac{\#(C_2)}{\#(\text{key-phrases in } R_1 \text{ and } R_2 \text{ are recognized})}.$$

The notation #(.) denotes the number of a specific situation. Based on this estimation method, we evaluate the posterior probability of a speech concept given the recognized key-phrases. Note that if no key-phrase in R_1 or R_2 is recognized, it means that no contribution can be derived from speech concept detection, and the discrimination work is done by visual-based detection only.

The case of discriminating strikeout and field out is done by considering key-phrases in R_3 and R_4. In the experiments, these probabilities were estimated based on the results of speech concept detection from five games.

(3) Combining Visual and Speech Opinions

In the duration where concepts C_1 and C_2 (single and walk, for example) cannot be explicitly discriminated, assume that the concept C_1 is detected from visual information, with confidence $P(C_1|\mathbf{x}_1)$. However, the concept detected from speech information is C_2, with confidence $P(C_2|\mathbf{x}_2)$. These two opinions compete and we have to make the final decision by checking their confidence values. To combine the opinions from different classifiers, Kittler et al. [Kitt98] describe the theoretical framework of different combining strategies. On the basis of the features from visual and speech data $Z=(\mathbf{x}_1,\mathbf{x}_2)$, we apply the sum rule to combine visual and speech opinions as follows:

$$\text{assign } Z \to C_j \text{ if } \sum_{i=1}^{2} P(C_j \mid \mathbf{x}_i) = \max_{k=1}^{2} \sum_{i=1}^{2} P(C_k \mid \mathbf{x}_i). \tag{4-3}$$

Although Kittler et al. proposed five rules (sum, product, max, min, and majority vote) for combining classifiers, we have similar performances by applying different rules. The experimental results shown in the next section are all based on the sum rule.

4.4.4 Results of Concept Detection

Most of the evaluation data in this paper are taken from the games of Chinese Professional Baseball Leagues (CPBL) [CPBL06], which are broadcasted by Videoland Sport Channel [Vide06]. To evaluate concept detection in various broadcasting situations, four broadcasting games with total length about ten hours are used. Three of them are CPBL games in year 2004 (CPBL1 and CPBL2 in Table 4-4)

and year 2005 (CPBL3), and one of them are from MLB (Tigers vs. Yankees at 2005/5/26). They are recorded directly from TV, and the commercials are not intentionally filtered out. Because the proposed framework only considers the caption information in pitch shots and shot transitions (pitch-field pairs), the commercials that are often classified as "other" shots would not degrade the detection performance. This flexibility makes the proposed approach more practical in developing a system that real-time analyzes broadcasting videos and immediately provides analytical results right after the game.

For shot classification, we briefly evaluate its performance by checking three half innings in three different games (210 shots), where averagely 92% accuracy can be achieved. Table 4-4 and Table 4-5 show the detection results of six frequently occurred concepts in terms of precision and recall rates. The term Hit/BB denotes the concepts of "single" or "walk," and "Out" denotes "strikeout" or "field out." The numbers in parentheses (n1/n2) in each row denote the count of concepts to calculate precision and recall. Overall, we obtain very promising results in detecting most concepts. At least 0.85 of precision rate and 0.9 of recall rate can be achieved. The detection performance in MLB is slightly worse because of worse shot classification and character recognition accuracy deriving from poorer video quality. Note that although only common concepts are shown in Tables 3-4 and 3-5, other rare concepts could also be correctly detected by the proposed method. For example, the only "triple" concept in CPBL2 and the only "catch steal" concept in CPBL3 are both correctly detected.

Table 4-4. Detection results of hit/bb, double, and home run.

Game		**Hit/BB**	**Double**	**Home Run**
CPBL1	Prc.	1 (15/15)	1 (6/6)	1 (2/2)
	Rec.	1 (15/15)	1 (6/6)	1 (2/2)
CPBL2	Prc.	1 (15/15)	1 (3/3)	1 (2/2)
	Rec.	0.83 (15/18)	1 (3/3)	1 (2/2)
CPBL3	Prc.	1 (17/17)	1 (3/3)	
	Rec.	0.89 (17/19)	1 (3/3)	
MLB	Prc.	1 (18/18)	1 (3/3)	
	Rec.	0.95 (18/19)	1 (3/3)	
Total	Prc.	1 (65/65)	1(15/15)	1 (4/4)
	Rec.	0.92(65/71)	1(15/15)	1 (4/4)

Table 4-5. Detection results of out, sacrifice, and double play.

Game		**Out**	**Sacrifice**	**Double Play**
CPBL1	Prc.	1 (35/35)	1 (5/5)	1 (3/3)
	Rec.	0.95 (35/37)	1 (5/5)	1 (3/3)
CPBL2	Prc.	1 (34/34)	1 (4/4)	0.75 (3/4)
	Rec.	0.89 (34/38)	1 (4/4)	1 (3/3)
CPBL3	Prc.	0.98 (43/44)	1 (2/2)	1 (2/2)
	Rec.	0.91 (43/47)	1 (2/2)	1 (2/2)
MLB	Prc.	1 (25/25)	0.67 (4/6)	0.75 (3/4)
	Rec.	0.81 (25/31)	1 (4/4)	0.75 (3/4)
Total	Prc.	0.99(137/138)	0.88(15/17)	0.85(11/13)
	Rec.	0.90(137/153)	1(15/15)	0.92(11/12)

Table 4-6 shows the results of discriminating confused concepts only based on visual information, i.e. 'Hit/BB' and 'Out' concepts in Tables 3-4 and 3-5. The discrimination performances of single, walk, and filed out are satisfactory, while that in strikeout is still needed to be improved. Considering the statistical characteristics of the pitcher and the batter would be our future direction to develop more reliable classifiers. Overall, the proposed framework achieves satisfactory performance without being drastically affected by game variations. An on-line system demo is in the "explicit concept detection" part at http://www.cmlab.csie.ntu.edu.tw/~wtchu/baseball/index.html. We present sample results of concept detection and provide concept-on-demand services on the web.

Table 4-6. Classification results of confused concepts.

Game		**Single**	**Walk**	**Strikeout**	**Field out**
CPBL1	Prc.	0.83 (10/12)	0.67 (2/3)	0.55 (6/11)	0.96 (23/24)
	Rec.	0.91 (10/11)	0.5 (2/4)	1 (6/6)	0.74 (23/31)
CPBL2	Prc.	1 (12/12)	1 (3/3)	0.8 (4/5)	0.93 (27/29)
	Rec.	0.8 (12/15)	1 (3/3)	0.44 (4/9)	0.93 (27/29)
CPBL3	Prc.	0.8 (8/10)	0.57 (4/7)	0.55 (11/20)	0.92 (22/24)
	Rec.	0.57 (8/14)	0.8 (4/5)	0.73 (11/15)	0.69 (22/32)
MLB	Prc.	0.86 (6/7)	0.73 (8/11)	0.3 (3/10)	1 (15/15)
	Rec.	0.55 (6/11)	1 (8/8)	0.33 (3/9)	0.68 (15/22)
Total	Prc.	0.88(36/41)	0.71(17/24)	0.52(24/46)	0.95(87/92)
	Rec.	0.71(36/51)	0.85(17/20)	0.62(24/39)	0.76(87/114)

To show the effectiveness of fusing multiple modalities, we evaluate discrimination results from three different games, which are totally nine hours in length and consist of 228 plays. The performance of three phases including visual concept only, speech concept only, and integrated decisions are demonstrated in Figure 4-10 (single vs. walk) and Figure 4-11 (strikeout vs. field out). F1 metrics, which jointly consider precision and recall, are illustrated:

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{4-4}$$

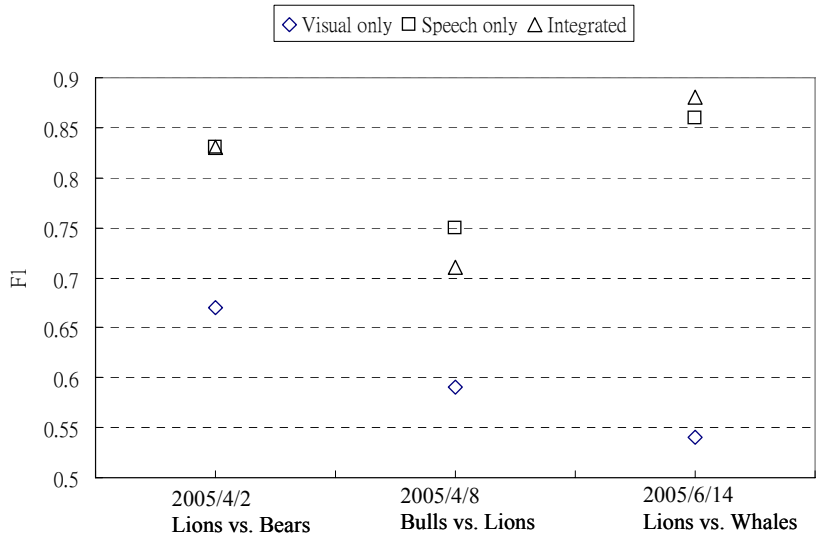


Figure 4-10. Discrimination performance of single vs. walk.

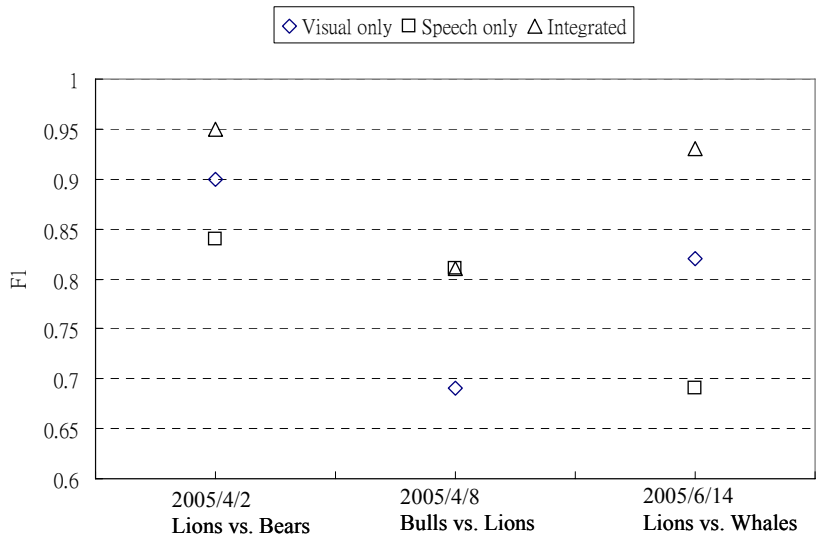


Figure 4-11. Discrimination performance of strikeout vs. field out.

In Figure 4-10, we see that combining two classifiers using the proposed fusion scheme outperforms single classifiers, except for the case of "2005/4/8 Bulls vs. Lions." The cause of this exception lies on some extremely abnormal broadcasting situations or shot classification errors, which make the evaluation of visual concept detection unreliable. Figure 4-11 shows the performance of discriminating strikeout from field out, where the enhancement of modality fusion is significant in two of the three games. The game of "2005/4/8 Bulls vs. Lions" relatively has worse performance because of lower character recognition accuracy caused by poorer video quality.

The overall performance of concept detection (totally thirteen different types of concepts) is listed in Table 4-7. With the help of speech information, the detection performance increases (in terms of F1) by 8% ~ 20% relative to the visual only method [Chu05-4].

Table 4-7. Overall performance of concept discrimination.

Games	Decision	Precision / Recall	F1
Lions vs. Bears	Visual	0.88 / 0.82	0.85
	Visual + speech	0.96 / 0.89	0.92
Bulls vs. Lions	Visual	0.76 / 0.68	0.70
	Visual + speech	0.85 / 0.74	0.79
Lions vs. Whales	Visual	0.77 / 0.73	0.75
	Visual + speech	0.93 / 0.88	0.90

Note that some byproducts can be obtained after concept detection. "Runs battered in (RBI)," which denotes number of scores as a direct result of a concept, can be calculated from the changes of scores. "Left on base (LOB)," which denotes the total number of runners who did not score when the batter made an out, can be calculated by checking how many bases were occupied before the batter was out. This information represents the effectiveness of concepts and can be good indicators for game abstraction.

4.5 Extended Applications

4.5.1 Automatic Game Summarization

A reasonable game summary should include the clips with scoring and the progress of effective offenses, like the sequence of "a single, a sacrifice bunt, and a double" that causes a score. We argue that approaches based on low-level features and probabilistic methods cannot accurately achieve the requirements without "explicit concept

detection." With the aid of explicitness, we can develop superior summarization and highlight extract modules to appropriately represent the content of games.

4.5.1.1 Significance Degree of Concepts

An attractive summary is a "condensed game [MLB06]", which consists of effective concepts that direct game progress and affect the final result of the game. To maintain informativeness of a condensed game, we give different significance degrees to different types of concepts according to their contributions. Babaguchi et al. [Baba04] propose an idea to define the significance degrees of concepts for American football, while they perform game summarization from existing text-based game logs rather than the results of automatic concept detection. We follow similar ideas and modify the definition of significance especially for baseball videos. Five levels of significance degrees are defined in the following:

- Rank 1: state change concepts. Only three states exist in team sports: "the two teams tie", "team A leads", and "team B leads". The concepts that cause one team to score and change the current state into a different state are called as state change concepts. They directly change the states of a game and are the indicators of winning pitcher, losing pitcher, and winning RBI. It is evident that these kinds of concepts pose the greatest significance.
- Rank 2: hits with RBIs. Hits with RBIs, not matter they are single, double, or home run, change score of a team and indirectly affect the result of the game. They also indicate the effectiveness of hits.
- Rank 3: hits without RBI and walk. Although no score is obtained, the number of hits is concerned with a player's batting average.
- Rank 4: outs with LOBs. These kinds of outs show that the batter fails to help teammates score. Larger LOB indicates more negative influence when a play makes an out.
- Rank 5: outs without LOB. Normal outs generally cover more than half of cases and give the least significance.

4.5.1.2 Selection of Summarization

According to different requirements of users, we provide various summaries that have different lengths and information coverage. We generate the most compact condensed game by concatenating rank-1 concepts, while a richer condensed game can be formed by collecting rank-1, rank-2, and rank-3 concepts. In addition to concept rank that is defined for each isolated concept, context of concepts in a half inning should also be considered in concept selection. In baseball games, there may be a chain of concepts to result in scoring. Although some of these concepts may be in lower rank,

the chain of concepts should be collected together to maintain the completeness of summary. For example, in Figure 4-12, a chain of double, strikeout, and single concepts occur and finally cause scoring. The single leads to a score because the second base is occupied. Hence it's no doubt that both the double and single concepts should be collected in the summary. Moreover, the audience usually expects the player to have a good play when some bases are occupied. The result of his play impresses the audience, no matter it's a good play causing RBI or a bad play causing LOB. Therefore, we also take account of the context of concepts and collect the strikeout concept in summary. On the other hand, if only one rank-3 concept occurs alone (no other concepts with ranks $\leqq$ 3), it should be ignored because fragmentary hits don't cause effective results.

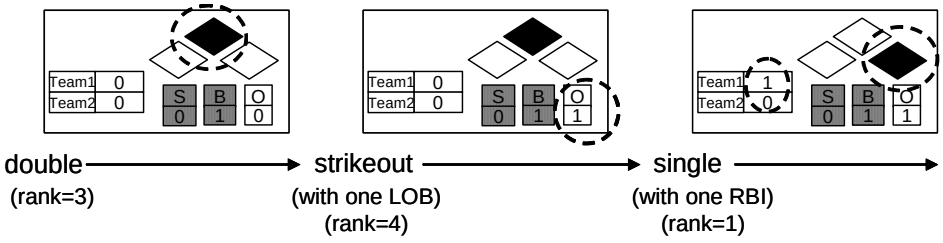


Figure 4-12. A chain of concepts that result in scoring.

On the basis of the ideas described above, three levels of game summaries are automatically generated by the following methods.

- Level 1: Only the concepts with rank 1 and rank 2 are collected. This level of summary contains the most compact results.
- Level 2: Basically, only concepts with ranks 1~3 are collected. Rank-3 concepts and rank-4 concepts are considered to be discarded or added through checking concept context:
  - Rank-1 and rank-2 concepts are definitely picked as the summary.
  - Check each rank-3 concept i.

    If both the ranks of the (i+1)-th and (i+2)-th (r_{i+1} and r_{i+2}) concepts are less than 4, pick them all as the summary.

    If r_{i+1} < 3 and r_{i+2} = 5, just pick the i-th and (i+1)-th concepts as the summary.

    If r_{i+1} = 4 and r_{i+2} = 5, ignore all the i-th, (i+1)-th, and (i+2)-th concepts.

Note that rank-3 and rank-5 concepts would not occur successively because of the inherent baseball rules.

This level of summary meets audience's expectation the most and is close to the condensed game made by professional sports reporters.

- Level 3: All concepts with ranks 1~5 are collected. This level of summary contains the most complete content of a game, while eliminating commercials or other irrelevant clips.

Note that the temporal relationships between concepts should be maintained, because a condensed game formed by disordered concepts is meaningless. Therefore, different ranks of concepts may be interlaced in the final summary. Moreover, the final length of summary depends on the content of a game. If two teams have a keen competition, the length of summary will be larger due to more rank-1 and rank-2 concepts. Table 4-8 shows the lengths of summaries from two different competitions.

Table 4-8. Lengths of summaries at different levels.

Games	**Length**
Original game of 'Lions vs. Whale' (2005/6/14)	3 hours 33 minutes
Level-1 summary	4 minutes 10 seconds
Level-2 summary	23 minutes 15 seconds
Level-3 summary	51 minutes 32 seconds
Original game of 'Bulls vs. Lions' (2005/4/8)	3 hours 14 minutes
Level-1 summary	2 minutes
Level-2 summary	20 minutes 12 seconds
Level-3 summary	49 minutes 43 seconds

4.5.1.3 Evaluation of Summarization

To evaluate the effectiveness of the proposed summarization method, we compare automatic game summarization with man-made condensed games, which are taken from a sport TV station [Vide06]. Concepts in the man-made condensed games are selected by professional sports reporters. Although the selected concepts may not be exactly the same from different reporters or different TV channels, they can be viewed as good references for evaluation. After checking the concepts in automatic summarization and man-made condensed game, two indicators are calculated:

$$\text{Precision}=\frac{N_c}{N_s}, \qquad \text{Recall}=\frac{N_c}{N_m},$$

where N_S is the number of concepts in automatic summary, N_m is the number of concepts in man-made summary, and N_c is the number of concepts in both summaries.

Table 4-9 shows the summarization performance of two games. The values n_2/n_1 in each inning denote that n_1 concepts are collected by the proposed process, and

among them, n_2 concepts are in man-made summary. From Table 4-9, the precisions of two level-2 summaries are 31/35=0.886 and 25/31=0.806, respectively. The corresponding recalls are 31/33= 0.939 and 25/30=0.833. Details of summary comparison can be seen in the "game abstraction" part at http://www.cmlab.csie.ntu.edu.tw/~wtchu/baseball/index.html.

Table 4-9. Performances of different levels of summaries.

Lions vs. Whales (2005/6/14)					
Inning	**1**	**2**	**3**	**4**	**5**
Man-made summary	5	12	0	3	1
Automatic summary	4/4	12/12	0	3/3	1/1
Inning	**6**	**7**	**8**	**9**	**Total**
Man-made summary	0	4	3	5	33
Automatic summary	0	4/4	3/6	4/5	31/35
Bulls vs. Lions (2005/4/8)					
Inning	**1**	**2**	**3**	**4**	**5**
Man-made summary	0	4	4	6	0
Automatic summary	0	4/4	3/3	6/8	0/4
Inning	**6**	**7**	**8**	**9**	**Total**
Man-made summary	7	0	5	4	30
Automatic summary	4/4	0	5/5	3/3	25/31

4.5.2 Automatic Highlight Generation

Another attractive application is game highlight extraction. To maintain entertaining functionalities within short time duration, highlight extraction poses different concerns from summarization. It is evident that effective concepts such as state change concepts or hits with RBIs should be highlighted. In addition, beautiful defense play such as diving catch or catch steal should also be highlighted, although they just cause a normal field out. In highlight extraction, we integrate the impacts of concept ranks, audio energy dynamics, and occurrence time to generate game highlight that well retains 'enjoyability' of a game.

4.5.2.1 Significance Degree of Concepts

- Rank-based Significance

For the requirement of highlight, we slightly modify the definition of concept rank. Double play, triple play, and catch steal concepts are categorized as rank-2 concepts to cover important defense. The rank-based significance degree S_r ($0 \le S_r \le 1$) of the ith concept E_i is quantitatively defined as

$$S_r(E_i) = 1 - \frac{r_i - 1}{5} \cdot \alpha, \tag{4-5}$$

where r_i ($1 \le r_i \le 5$) denotes the rank of the ith concept, and α ($0 \le \alpha \le 1$) is the parameter controlling the weight of concept rank.

- Time-based Significance

The concepts occurring at the latter stage of games are usually more attractive to users, especially when the two teams tie or have slight score difference. The time-based significance S_t ($0 \le S_t \le 1$) is defined as

$$S_t(E_i) = 1 - \frac{N - I(E_i)}{N} \cdot \beta, \tag{4-6}$$

where $I(E_i)$ denotes the inning in which the concept E_i occurs, N is the number of total innings in a game (usually nine innings in a game), and β ($0 \le \beta \le 1$) is the parameter controlling the weight of occurrence time.

- Audio-based Significance

The anchorperson often comments excitedly and the audience cheers loudly when a beautiful play or an important hit occurs. We extract audio energy and analyze its dynamics over time to show how the anchorperson or the audience reacts to each concept. Audio energy dynamics can be broadly classified into regions of attack, sustain, decay, and silence [Dora02]. We particularly focus on detecting attack because it indicates the occurrence of an exciting concept. A beautiful defense play, which may be viewed as a normal field out at the concept detection stage, can be figured out by employing audio cues.

We evaluate the envelope of power spectrum and only concentrate on how audio energy increases:

$$\begin{aligned} &\textit{if } e_k - \text{mean}(e_{k-w}, ..., e_{k-1}) > 0 \\ &\quad d_k = e_k - \text{mean}(e_{k-w}, ..., e_{k-1}), \quad k = 1, 2, ..., M, \\ &\textit{else } \quad d_k = 0, \end{aligned}$$

where e_k denotes the average energy of the k-th audio segments, and M denotes the number of audio segments within the duration of an concept. Energy difference d_k is calculated by subtracting average energy of previous w segments from e_k. Each audio segment is of length one second, and w is set as four in this work. The maximum energy difference within the concept duration is chosen and is quantized into one to five to be the clues of audio-based significance:

$$D_i = \text{Quantize}(\max(d_k)). \tag{4-7}$$

Accordingly, the audio-based significance S_a ($0 \le S_a \le 1$) is defined as

$$S_a(E_i) = 1 - \frac{5 - D_i}{5} \cdot \gamma, \tag{4-8}$$

where γ ($0 \le \gamma \le 1$) is the parameter controlling the weight of audio cues.
By combining the impacts of concept rank, occurrence time, and audio energy dynamics, the integrated significance degree $S(E_i)$ ($0 \le S \le 1$) is given by

$$S(E_i) = S_r(E_i) \cdot S_t(E_i) \cdot S_a(E_i). \tag{4-9}$$

Different highlights could be obtained by changing the weighting parameters of α, β, and γ. In our experiments, we set α, β, and γ as 0.5, 0.2, and 0.3, respectively.

4.5.2.2 Highlight Selection Algorithm

In general, highlight selection can be formulated as a knapsack problem. That is, given segments of different significance and lengths, find the most significant set of segments that fit in a knapsack of fixed length. The significance of each concept is estimated by the process described above. As regards the concept length, we just extract a concept from the last pitch to the first pitch of the next concept. This method reserves the most significant parts and provides efficient concept presentation. However, the last concept in each half inning would be very long because commercials would be inserted. Therefore, we limit the length no more than forty seconds to prevent a very long concept.

In our work, we implement a greedy approach to select highlighted concepts. By considering the time limitation given by the user and concept context, the highlight selection algorithm is as follows:

Input: the user-defined highlight length T and the set of concepts E in the game.
Output: the set of highlighted concepts A.

```
HIGHLIGHT_SELECTION(T, E)
1  A ← ∅
2  sort E into nonincreasing order by significance degrees
3  for each e_i ∈ E
4     do if length of (A ∪ {e_i}) < T
5           then A ← A ∪ {e_i}
6                SMOOTH(A)
7  return A
```

Similar to the context idea in automatic summarization, adjacent relationships between highlighted concepts are considered in the SMOOTH process. For three adjacent concepts A-B-C, if both A and C concepts are selected as highlight, B is also selected to maintain the complete progress of game highlight. Finally, the selected concepts are sorted by the occurrence time to maintain temporal coherence.

4.5.2.3 Evaluation of Highlight

Due to lack of ground truth for evaluating game highlight, we invited 24 persons, including 21 males and 3 females, to perform subjective experiments based on highlights extracted from two games. We impose two assumptions on the subjects: 1) none of the subjects saw these games before. This assumption is reasonable and is for simplification purpose, because we cannot expect every subject affords to spend more than six hours to see two baseball games. 2) The subjects judge the selected highlighted concepts based on the concepts themselves rather than their preference on specific teams or specific players.

The experimented scenario is set to be concept-based. Because of the assumption 1, we didn't ask subjects "Does the game highlight contain the most highlighted parts of this game?" Instead, we request subjects to evaluate each selected concept. This evaluation somehow represents the "accuracy" of the proposed highlight selection method. Because a concept's significance sometimes depends on the effectiveness of the succeeding concepts, we present multiple concepts together if they are in the same half inning. After the presentation of one half inning, the subjects give one opinion score (from one to five, indicating from bad to excellent) to each selected concept to judge whether it's a highlight part.

The selected highlight concepts and their corresponding meanings are listed in Tables 3-10 and 3-11. Table 4-12 shows the subjective results of highlights with different lengths. Eleven concepts and eight concepts are selected to construct 7-min and 5-min highlights, respectively. From Table 4-12, highlights from both games satisfy users and get average score larger than 3.3. The shorter highlight getting higher score indicates that the proposed significance degree modeling positively captures the characteristics of highlights. Moreover, human's subjective satisfaction is slightly affected by the competitive content of games. In "Bulls vs. Lions", three home runs occurred and two teams have a keen competition. On the other hand, the team "Lions" dominates in "Lions vs. Bears", and the game presents flat content. Therefore, the concepts selected in "Bulls vs. Lions" often excite the subjects more and get higher scores.

Recently, game highlights are popular materials for representing game content in sports news or on-line entertainment services [MLB06]. A man-made highlight, e.g.

the highlight reel of an MLB game, consists of video shots elaborately edited and remarkable comments. This kind of game highlight impresses the audience while it requires lots of professional equipments and working time. In this work, we present an automatic highlight selection method that provides satisfactory highlights and is free from user intervention. For personalization purpose in digital home environment, users can adjust the weights with respect to concept rank, occurrence time, and audio energy dynamics to generate different flavors of highlights.

Table 4-10. The selected concepts in "Lions vs. Bears."

Lions vs. Bears (2005/4/2)	
Inning	**Selected concepts**
Top 2nd	sacrifice fly (RBI=1)
Bottom 5th	hit by pitch
Top 6th	walk double sacrifice fly (RBI=1)
Top 6th	sacrifice bunt single (RBI=1) double (RBI=1)
Bottom 8th	single steal walk

Table 4-11. The selected concepts in "Bulls vs. Lions."

Bulls vs. Lions (2005/4/8)	
Inning	**Selected concepts**
Top 4th	home run (RBI=1) single home run (RBI=2)
Bottom 4th	sacrifice fly (RBI=1)
Bottom 6th	home run (RBI=2) field out (good defense play)
Bottom 8th	walk field out (good defense play)
Bottom 9th	single sacrifice bunt sacrifice bunt

Table 4-12. The evaluation results of highlights from two games.

Game highlights	**Average mean opinion score**
Lions vs. Bears 7-min highlight (11 concepts)	3.35
Lions vs. Bears 5-min highlight (8 concepts)	3.43
Bulls vs. Lions 7-min highlight (11 concepts)	3.67
Bulls vs. Lions 5-min highlight (8 concepts)	3.87

4.5.3 An Integrated Baseball System

The results of explicit concept detection imply an intuitive application, i.e. concept-on-demand service via a scoreboard-like interface, as shown in Figure 4-13. Users can select the games we have processed and see what happened in the game at a glance. The concept list (left-up side of Figure 4-13) shows what kind of concept occurred, who did it, and the corresponding timestamps.

The detected metadata of baseball videos can be stored in a database for more flexible access. We cooperate our results with a question analysis system [Day05][ASQA06] and build a baseball question answering system, as shown in Figure 4-14. Users can input a natural query string like “I want to see the homeruns shot by Player A.” Through the query analysis, the system knows the concept of interest is homerun, which was hit by Player A. By checking the detected metadata stored in database, this system automatically retrieves the corresponding video clips and answer user’s question by video.

The context of baseball concepts often represents the conventions or tactics of a team or the performance of a player. Therefore, a sequential mining technique is integrated to find the subtle characteristics. The mining results are shown as probabilistic presentations. The occurrence probability of a specific concept, which follows a series of concepts, is obtained by giving enough training data. Figure 4-15 shows a flash-based user interface, in which the mining results corresponding to a specific play are displayed. For example, in the case that no out and the first and the second bases are occupied, the occurrence probability of a sacrifice is 71.43%, and the occurrence probability of a double or a strikeout is 14.29%. These mining results not only enhance the entertainment functionality of watching baseball games, but also provide some foundations for further knowledge-level analysis.

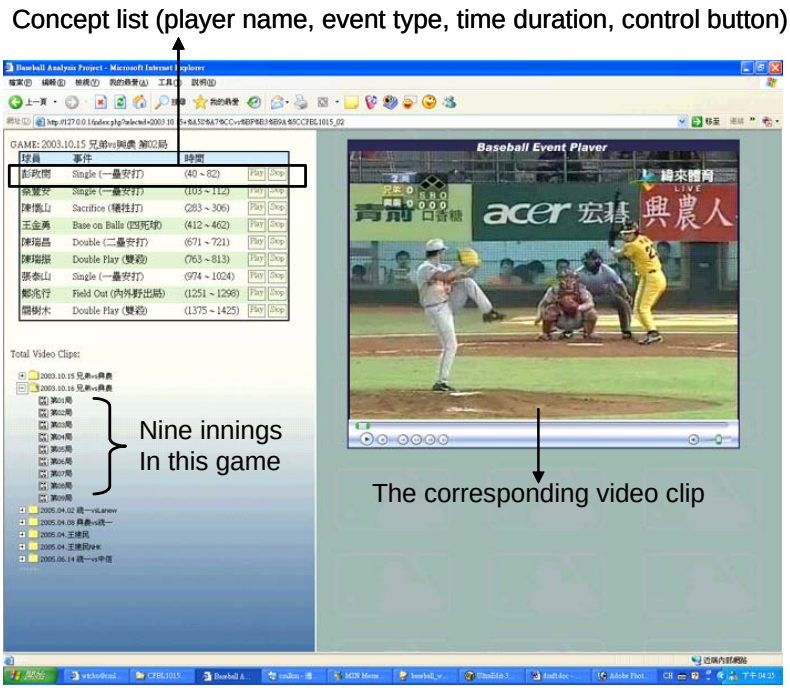


Figure 4-13. Snapshot of the baseball concept-on-demand system.

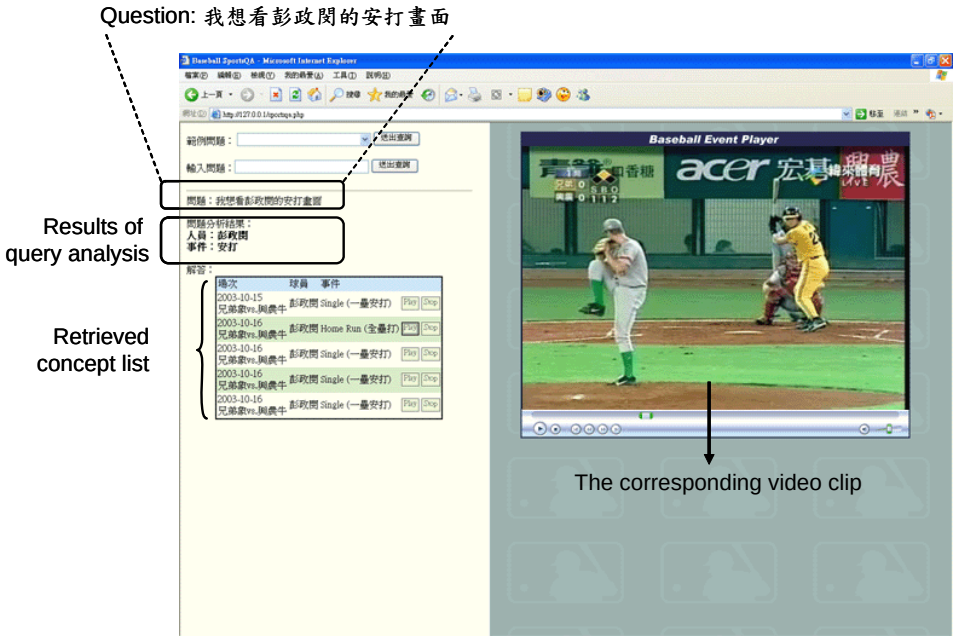


Figure 4-14. Snapshot of the baseball question answering system.

Figure 4-15. Integrated user interface and the presentation of mining results.

4.6 Discussion and Summary

We have presented a complete strategy to perform explicit concept detection and develop practical applications for broadcasting baseball videos. Color and geometric information is utilized to achieve shot classification. Adaptive field color determination and dynamic pitcher position detection are devised to make shot classification more reliable and general. Then, the rule-based and model-based decision methods are integrated to explicitly detect thirteen baseball concepts. Official baseball rules are transformed into a decision tree in the rule-based decision module, while the context of shots is considered in the model-based decision module. Speech information is also taken into account to improve the performance of semantic concept detection. A fusion scheme based on combining probabilistic classifiers is proposed. Finally, on the basis of explicit concept detection, automatic game summarization and highlight selection are implemented to preserve "informativeness" and "enjoyability" within short duration. Elaborate design of the significance degree of concepts and various evaluations are presented. The proposed approaches automate broadcasting baseball video analysis and facilitate various applications.

The primary idea corresponds to the framework described in Chapter 2 and is illustrated in Figure 4-16. Based on visual features, statistical pattern recognition techniques are used to recognize number of score and out, and rule-based methods are used to perform shot classification. The mid-level representation is constructed by information changes on caption and shot transition information. With the help of baseball rules and broadcasting conventions, we achieve explicit semantic concept

detection via the methods of decision tree and k-nearest neighbor. In this framework, the results of different types of classifiers are combined to infer what happened in baseball games. From feature to semantics, the hybrid approach that integrates rule-based and statistical techniques effectively bridges the semantic gap.

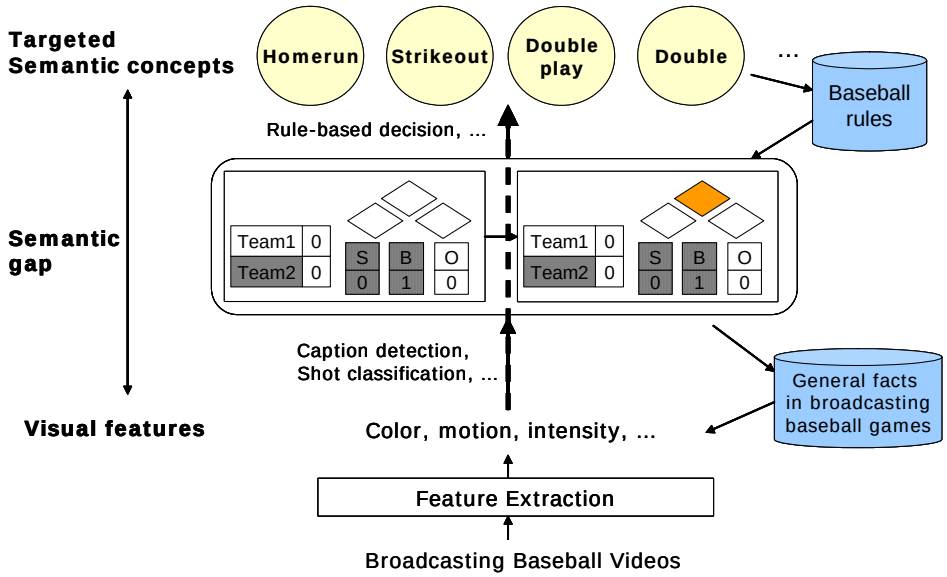


Figure 4-16. Explicit baseball concept detection in terms of the framework described in Chapter 2.

In summary, the major contribution of this work is that we propose a systematic method to explicitly and comprehensively analyze baseball videos. We believe that the analytical results and developed applications are more realistic to users. It's arguable that this work specifically concentrates on the domain of baseball games and is not intuitive to be extended. However, utilizing domain knowledge more exhaustively leads to more practical functionalities. We engage in thorough studies of baseball video analysis and report results that appropriately match the demands of most users. From all of our survey, the proposed method best exploit official rules in concept detection and game abstraction, and provides the most comprehensive and practical results in baseball video analysis.

Although we explicitly detect various concepts in baseball games, some special concepts are still not included. For example, player substitution and defense error are subtle or are determined subjectively by the umpire. Furthermore, there is still space to improve the highlight selection method. More sophisticated concept modeling can be designed. For example, increase the significance degree of the concept that a slugger gets a strikeout, or suppress a non-scoring concept's significance if only one concept is selected in the half inning. Therefore, various applications with different purposes and for different services can be developed.

Chapter 5
Semantic Analysis in Sports Videos through Ball Trajectory

5.1 Introduction

Many techniques based on color, motion, caption [Zhan02], or external game logs [Baba04] have been proposed for sports video analysis. Most approaches focus on detecting predefined event sets in games and performing game abstraction. On the other hand, some implicit game statistics, which may be helpful in tactic analysis or improving athlete's performance, have drawn little attention. For example, possession time in soccer games indicates each team's performance, and pitch type usage in baseball games indicates the pitching tactics. This subtle but useful information cannot be modeled or detected through conventional content-based approaches. In this chapter, we propose a new medium, i.e. ball trajectory, in sports video analysis. It is believed that ball trajectories can provide new sights in more advanced and useful sports video analysis.

Recently, approaches based on ball trajectory have been proposed to facilitate implicit game status extraction. Yu et al. [Yu03] detect and track ball trajectory in soccer games, and perform possession time analysis and play-break structure discovery. For baseball games, the well-known K Zone system [Guez02] is reported to track pitching baseball trajectory. Two cameras (locating high above the home plate and the fist base) and three subsystems are equipped to real-time tracking ball in broadcasting baseball games. More specifically, Theobalt et al. [Theo04] track the position, velocity, rotation axis, and spin of the pitching ball with low-cost commodity.

The approaches described above are only applied to some specific games or should be equipped with high-cost tracking instruments. Nowadays, tremendous video sequences can be accessed on the internet [MLB06], but such entertaining functionality is not provided. Techniques that automatically extract ball trajectory without specific equipment settings are worth developing to enrich the experience of watching ball games. In this chapter, we focus on extracting ball trajectory from single-view video sequences. We apply a Kalman filter-based approach [Welc04] to

perform ball tracking and generate trajectory candidates. This approach robustly provides good tracking performance even some real ball candidates are missing in some frames. From the detected trajectory candidates, the optimal ball trajectory is determined by a physical model [Adai02].

Extracting the ball trajectory significantly aids subtle information analysis in ball games. For example, in baseball games, the pitching ball trajectory indicates the skill of a pitcher, and the sequential pattern of several consecutive pitches unveils the pitching tactics. In soccer games, how the ball moves aids in detecting possession time and play-break structure. With the help of goal-mouth detection, concepts like goal and shot can be automatically detected [Yu03]. Moreover, in tennis games, we can analyze game tactics, such as drop-shot, volley, or passing shot, through checking the ball trajectory. With the helps of such informative information, we are able to advance sports video analysis towards more useful applications.

In this chapter, we take baseball trajectory extraction as the main example, while the same process can be applied to other sports videos. Some sample results and possible applications will be introduced for baseball, soccer, and tennis videos.

5.2 System Overview

Figure 5-1 illustrates the framework of ball trajectory extraction. Given a video sequence, ball candidates in each frame are first detected by checking color, position, size, and shape information. Several ball candidates may be extracted for one frame, and the real ball object may be misdetected because the ball is occluded by players or is merged into with white regions, such as player's white uniform or advertisement.

On the basis of ball candidates, a trajectory forming process is elaborately designed to concatenate isolated ball-like objects and generate a reasonable ball trajectory. This process consists of three stages: trajectory segment generation, trajectory candidate generation, and physical model-based validation.

(1) A Kalman filter-based approach is used to track the positions of ball-like objects and generate trajectory segments.

(2) Because the real ball object is often occluded by other objects, it's often the case that none of the trajectory segments present the real ball trajectory. We develop a trajectory candidate generation module to interpolate the ball position between two adjacent trajectory segments. Trajectory candidates that last the whole video sequences are consequently generated.

(3) Many trajectory candidates may be detected for a video sequence. However, only one trajectory is the valid ball trajectory. We devise a physical model-based validation module to validate the detected candidates and

determine which of them is the most reasonable ball trajectory. The planar ball position (in terms of (x,y)) of each frame is finally determined.

Figure 5-2 shows two sample results of trajectory detection in baseball games. The pitch in Figure 5-2(a) is captured from TV broadcasting [Vide06] and the pitch in Figure 5-2(b) is downloaded from internet [MLB06]. We can see that the proposed approach works well in different types of pitching conditions (Chinese Professional Baseball League vs. Major League Baseball and right-hander vs. left-hander).

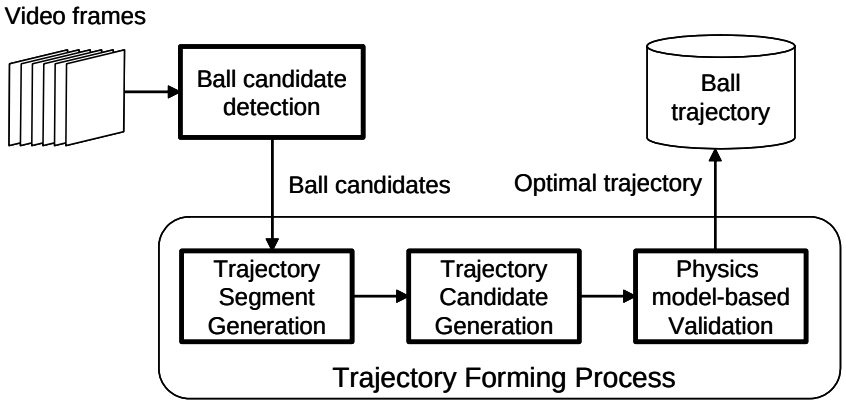


Figure 5-1. The framework of ball trajectory extraction.

(a) (b)

Figure 5-2. Two sample results of ball trajectory detection. (a) Chinese Professional Baseball League, right-hander; (b) Major League Baseball, left-hander.

5.3 Ball Candidate Detection

Given a video sequence, we first detect white background regions that always remain white in the whole sequence (usually 10~18 frames in 30-fps pitching baseball sequences). For each video frame, the pixels in white background regions are omitted first. White objects that are out of background regions are viewed as ball candidates if they meet certain constraints, including color, position, size, and shape. Figure 5-3

illustrates the process of ball candidate detection.

1) Color filter: Color of the ball is similar to white even in different broadcasting situations. Therefore, the objects whose color is not close to white are filtered out. The definition of ball color may be different in different sports. In our implementation for baseball trajectory extraction, objects are claimed to be white if all their RGB values are larger than 150.
2) Position filter: In baseball games, the ball always flies in specific region, either in different broadcasting styles, right-hander or left-hander. We can feel free to filter out the suspected white objects in very high or very low regions. In our implementation, we discard all objects higher than (1/5)×(frame height) and lower than (4/5)×(frame height).
3) Size filter: Although the size of ball may be different in various game broadcasts, it falls within a specific range. This filter sieves out the white objects with reasonable size and ignores pixel-size white noises or massive objects caused by the player's white uniform or advertisement boards. In baseball experiments, we use 352×240 images, and the range of reasonable ball size is from 2 pixels to 10 pixels.
4) Shape filter: Finally, the ball should be similar to a circle on screen. The objects that are far from circle are filtered out. An object's radius r is defined as the maximum value of width or height. An object is viewed as a circle if the ratio of object area to πr^2 is larger than 0.3.

Note that the parameters described above are just for pitching baseball sequences. Different settings may be employed when we apply the process to find ball trajectories in other ball games.

After these filtering processes, reasonable ball-like objects are detected. Figure 5-4 shows the detected ball candidates in different video frames, in which the x axis denotes the frame index, and the y axis denotes the diagonal distance between the ball object and the left-top corner of the frame. However, many of them are noises or none of them is the real ball. We have to devise a method to concatenate some ball candidates and generate reasonable trajectories.

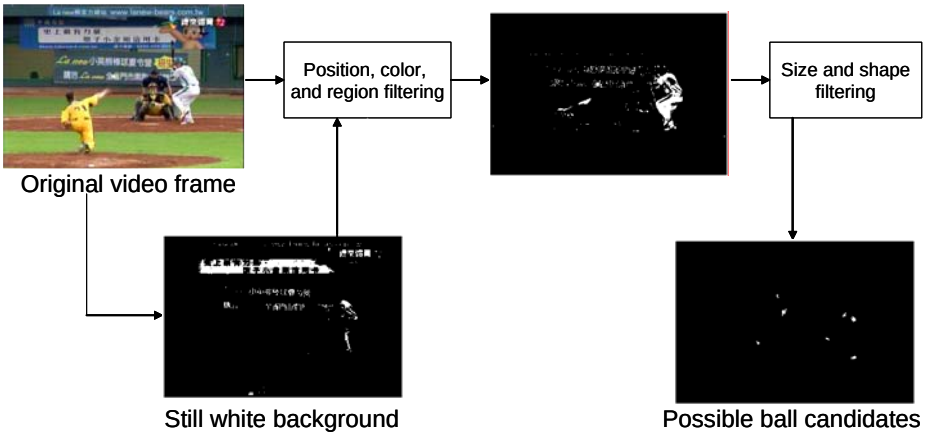


Figure 5-3. The Flowchart for ball candidate detection

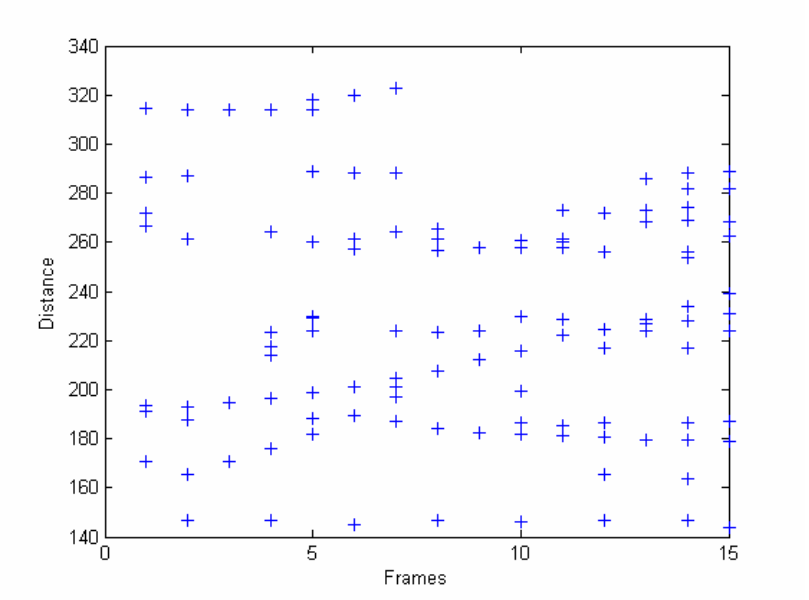


Figure 5-4. Ball candidates in different video frames.

5.4 Trajectory Forming Process

The trajectory forming process consists of three steps. We first connect neighboring ball candidates in adjacent frames to form trajectory segments. If the real ball is completely detected in the whole pitching sequences, we can feel free to say that one of these trajectory segments is the real ball trajectory. However, ball is often misdetected because of occlusion, merging, or deformation, and the real trajectory is cut into several disjoint segments. Therefore, the process of trajectory candidate generation interpolates the missing part between two segments and tries to generate trajectory candidates that last for the whole video sequence. Because only one of the trajectory candidates is the real ball trajectory, we have to find the one that best

matches the movement of a ball. In the final stage, an elaborate trajectory validation process is developed, on the basis of physical principles, to filter out illegal trajectories.

5.4.1 Trajectory Segments Generation

The main challenge of trajectory generation is that the ball often overlaps with white objects and is not detected (filtered out) in the process of ball candidate generation. Therefore, the developed trajectory forming process should estimate the missing ball positions and generates reasonable trajectory candidates. In this work, we apply a Kalman filter-based approach to track the ball positions.

In general, the Kalman filter describes a system as:

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_k, \tag{5-1}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \tag{5-2}$$

$\mathbf{x}_k$ is the state vector (representing the vector of estimated ball position at the *k*th frame), $\mathbf{A}_k$ is the system evolution matrix, and $\mathbf{w}_k$ is the system noise. $\mathbf{z}_k$ is the measurement (positions of the detected ball candidates), $\mathbf{H}_k$ is the unit array, and $\mathbf{v}_k$ is the measure noise.

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of measurements [Welc04]. Equation (5-1) denotes a *time update* process and equation (5-2) denotes a *measurement update* process. The time update equation forward projects the current status to *a priori* estimate for the next time step. The measurement update equation incorporates a new measurement into *a priori* estimate to obtain an improved *a priori* estimate.

In general, the time update process can be seen as a predictor, and the measurement update process can be seen as a corrector. The whole process proceeds "estimation" and "update" alternately. It adaptively adjusts the evolution matrix for estimation, according to the real measurements. Figure 5-5 illustrates the iterative process of a Kalman filter.

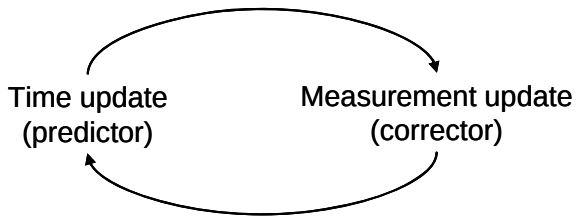


Figure 5-5. The iterative process of a Kalman filter [Welc04].

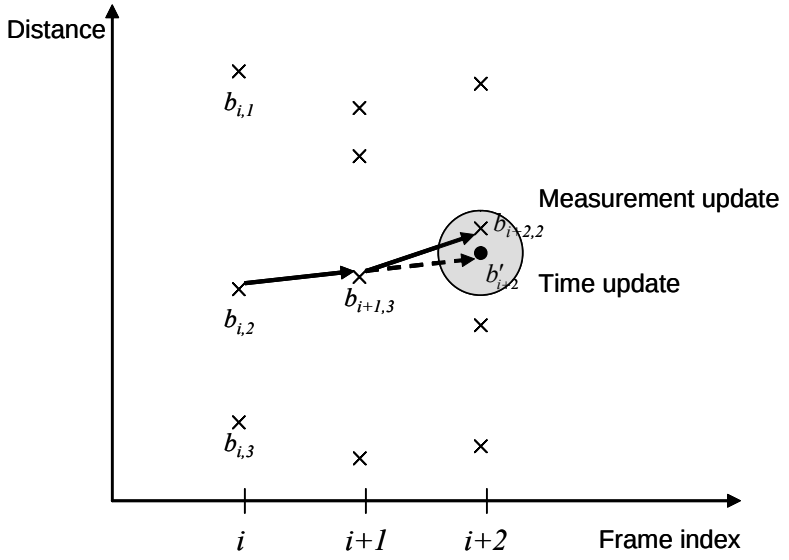


Figure 5-6. Kalman filter-based tracking in ball trajectory extraction.

Figure 5-6 illustrates the tracking process, in which each cross denotes a detected ball candidate. We first find trajectory seeds to start the Kalman filtering process. A trajectory seed is a pair of ball candidates that are spatially close to each other in two adjacent frames. In Figure 5-6, the ball candidates $b_{i,2}$ and $b_{i+1,3}$ in the ith and $(i+1)$-th frames are concatenated as an initial trajectory seed. In our implementation, two ball candidates are viewed as a trajectory seed if both their vertical and horizontal distances are less than 15 pixels.

After using the found seed to estimate the system evolution matrix $\mathbf{A}_k$, we grow the trajectory forward along the time dimension. The suspected ball position b'_{i+2} in the $(i+2)$-th frame is estimated by the Kalman filter, as illustrated by the dash arrow in Figure 5-6. In the measurement update stage, if there is any ball candidate close to the estimated position, the trajectory extends along this measurement, and we update the system evolution matrix. In this example, the ball candidate $b_{i+2,2}$ is close to the estimated position and is the measured ground for extending the trajectory and updating tracking parameters. In our implementation, b'_{i+2} and $b_{i+2,2}$ are claimed to be close because their Euclidean distance is less than 15 pixels. The classical predictor-corrector process [Welc04] repeats until all video frames are analyzed or no close candidates can be the basis for trajectory growing.

Figure 5-7 shows all the trajectory segments found by the Kalman filter-based process. Note that the real ball trajectory may not exist at all frames. Therefore, we have to concatenate these trajectory segments in a reasonable way and construct trajectory candidates which remain in the whole duration.

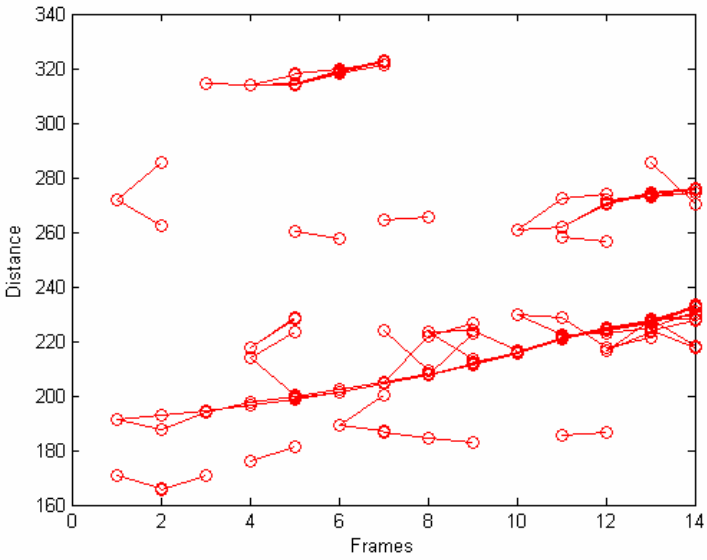


Figure 5-7. Examples of the detected trajectory segments.

5.4.2 Trajectory Candidates Generation

To generate trajectory candidates from the found trajectory segments, a process that consists of three stages are developed. From the trajectory segments pool, we search for segments that are probably able to be concatenated as a reasonable trajectory. By evaluating how close between the endpoints of two segments, this process decides whether to concatenate them or not. The connection process repeats until all trajectory segments are examined. The ball is often misdetected in the final few frames of pitching because of many noises caused by the batter's uniform or catcher's chest protector. If the detected trajectory candidate only lacks a few frames (less than 4 frames), we fit the trajectory with a polynomial and extend it to be a complete trajectory candidate. This process is illustrated in Figure 5-8, and the details of the three stages are described as follows.

1) Find stage: For each trajectory segment T_i, which ends at the e_i-th frame, find the trajectory segments that start before the (e_i+5)-th frame, and the distance between their end points is less than a threshold. In the example of Figure 5-16, the trajectory T_j is close to T_i and is selected to be the candidate for connection.
2) Connect stage: If T_i is longer than 3 frames, use a polynomial to fit T_i, and estimate the ball position at the (e_i+3)-th frame. If the Euclidean distance between the estimated ball position and the endpoint of T_j is less than a threshold, connect T_i and T_j. The missing ball positions at the (e_i+1)-th and the (e_i+2)-th frames are determined by the estimated values based on T_i.

The process returns to the first stage until the connected trajectory reaches the final video frame or no other segments are valid for connecting.

3) Extend stage: If the connected trajectory segments are longer than (L-4) frames and shorter than L frames (L is number of total frames of a pitching), use a polynomial to fit this trajectory and estimate the rest of this trajectory.

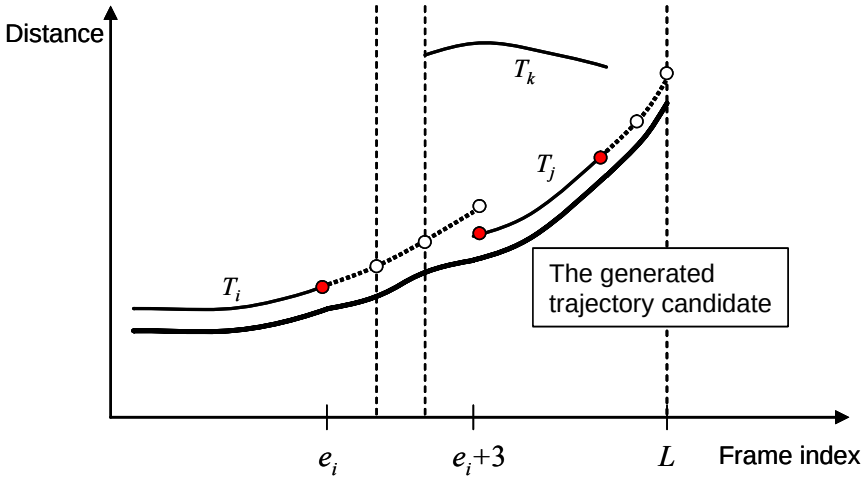


Figure 5-8. An example of trajectory candidate generation.

After these processes, we generate trajectory candidates that last for the whole video sequence, as illustrated in the bold solid line in Figure 5-8.

The trajectory generation process may generate many trajectory candidates that are drastically different from the real ball trajectory, though all of them conform to the constraints described above. For example, in pitching baseball videos, the catcher's white kneecap often moves smoothly and forms a plausible trajectory. Therefore, we devise a validation method to examine the detected trajectory candidates on the basis of physical principles. A real ball trajectory should conform to aerodynamics, in which the gravity, air friction, velocity, spin rate, and other factors affect the movement of the flying ball.

5.4.3 Physical Model-Based Trajectory Validation

5.4.3.1 Physical Model of Ball Trajectory

There have been many literatures on aerodynamics of baseball, by which we can confirm the reasonability of the generated trajectory candidates. According to the physics of baseball [Adai02], the trajectory of a ball can be roughly determined by its velocity, rotation axis, and spin rate. To induce the physical characteristics of a flying baseball, we simulate trajectories of fastball, curveball, and slider and gather the statistics of corresponding trajectory vectors. The position of a ball in x (right of batter), y (up from batter), and z (towards batter) directions can be formulated as

follows:

$$x_{t+1} = x_t + v_x(t) \times t + \frac{1}{2} a_x(t) \times t^2,$$
$$y_{t+1} = y_t + v_y(t) \times t + \frac{1}{2} a_y(t) \times t^2, \quad (5\text{-}3)$$
$$z_{t+1} = z_t + v_z(t) \times t + \frac{1}{2} a_z(t) \times t^2,$$

where x_t is the horizontal position at time t, $v_x(t)$ is the velocity in horizontal direction, and $a_x(t)$ is the corresponding acceleration. Related to the releasing angle and beginning velocity v_0, the x, y, z components of velocity are:

$$v_x(0) = v_0 \times \sin(aim_x \times \frac{\pi}{180}),$$
$$v_y(0) = v_0 \times \sin(aim_y \times \frac{\pi}{180}), \quad (5\text{-}4)$$
$$v_z(0) = v_0 \times \cos(aim_x \times \frac{\pi}{180}) \times \cos(aim_y \times \frac{\pi}{180}),$$

where aim_x (aim_y) is the included angle between the releasing vector (the direction of v_0) and the yz (xz) plane, as shown in Figure 5-9.

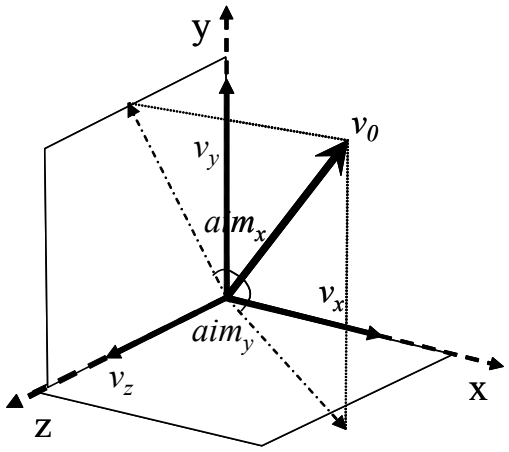


Figure 5-9. Velocity components of the releasing ball.

The evolution of velocity can be formulized as:

$$v_x(t+1) = v_x(t) + a_x(t) \times t,$$
$$v_y(t+1) = v_y(t) + a_y(t) \times t, \quad (5\text{-}5)$$
$$v_z(t+1) = v_z(t) + a_z(t) \times t,$$

and the evolution of acceleration becomes:

$$a_x(t) = B \times s_y \times v_z(t) - fv \times v_0 \times v_x(t), \quad (5\text{-}6)$$

$$a_y(t) = B \times s_x \times v_z(t) - G - fv \times v_0 \times v_y(t), \tag{5-7}$$

$$a_z(t) = B \times \left(s_y \times v_x(t) - s_x \times v_y(t) \right) - fv \times v_0 \times v_z(t), \tag{5-8}$$

where B and fv are coefficients for spin and air friction, G is the acceleration of gravity, s_x is the spin rate with rotation axis x. In each direction, the evolution of acceleration is affected by the force evoked by spin and the drag force. For example, in equation (5-6), the acceleration in x direction is strengthened by the drag force caused by y-direction spin (the first item), and is reduced by the air friction that is proportional to the x-direction speed. On the other hand, the gravity should also be considered in formulizing the y-direction acceleration. The imbalance of pressure caused by spin is known as "Magnus effect" [Adai02].

With these formulas, given the beginning velocity v_0, spin rate in x and y directions (s_x and s_y), we can simulate ball trajectories in different conditions. To find the criterion for valid ball trajectories, we simulate trajectories with different parameter sets that are possible conditions a pitcher can evoke. Table 5-1 shows the ranges of simulation parameters. Totally 715 (5×13×11) different trajectories were simulated. Note that in physics books the orientation of rotation axis also should be considered. However, in our experiments, the difference of rotation orientation affects slightly and can be neglected.

Table 5-1. Ranges of simulation parameters.

Parameters	Range
v_0	60, 70, 80, 90, 100 (mph)
s_x and s_y	-600, -500, …, 400, 500, 600 (rad/s)
aim_x and aim_y	-5°, -4°, …, 3°, 4°, 5°

For each simulated trajectory, we compute the included angle of two adjacent flying vectors:

$$\theta_i = \cos^{-1}\left(\frac{\mathbf{v}_i \bullet \mathbf{v}_{i-1}}{\|\mathbf{v}_i\| \times \|\mathbf{v}_{i-1}\|} \right), \tag{5-9}$$

where $\mathbf{v}_i = \left(x_i - x_{i-1}, y_i - y_{i-1}, z_i - z_{i-1} \right)$.

After simulating all possible trajectories, the included angles of adjacent vectors are gathered to be the reference for trajectory validation. Figure 5-10 shows the angle histogram. In this histogram, we can see that all legal angles fall into the range less than 1.2°. This characteristic imposes a constraint on legal trajectory.

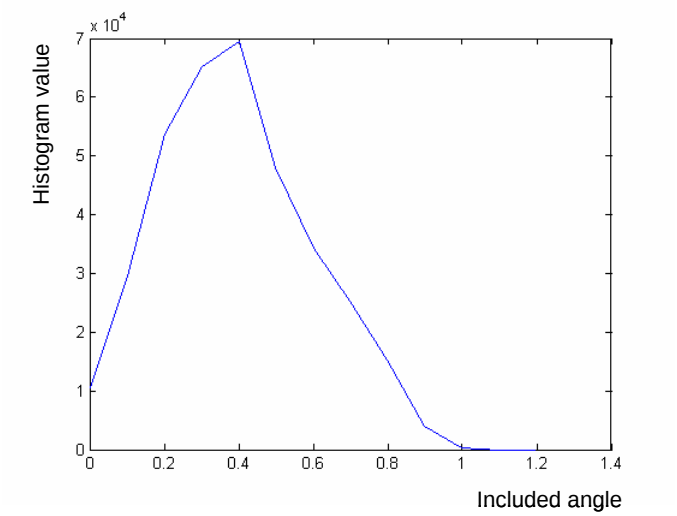


Figure 5-10. The angle histogram of trajectory vectors.

5.4.3.2 Trajectory Validation via Physical Limitation

With the physical limitation derived from trajectory simulation, we can filter out abnormal trajectory candidates. However, what we extract from single-view video sequences are 2-dimensional (2-dim) ball trajectories, in terms of pixels. We should estimate the depths (the ball positions at the *z*-axis) so that the constraints described in the previous subsection can be applied.

In this work, we estimate the proportion of the vertical movement (movement at the *y*-axis) to the depth in our simulation processes. The ratio of the vertical movement to the depth is estimated as 0.0558. Actually, this ratio matches the naïve estimation, which can be calculated through dividing average vertical movement (a reasonable assumed value is 1 meter) by the distance from the mound to the home plate (18.44 meters). On the other hand, the average vertical movement in our dataset is estimated as 38.1736 pixels. Proportionally, the depth of the detected 2-dim trajectories is estimated as 38.1736/0.0558 ≈ 684 pixels, as illustrated in Figure 5-11. The depth of each ball candidate is then obtained through the estimated depth divided by the frame number of the sequence.

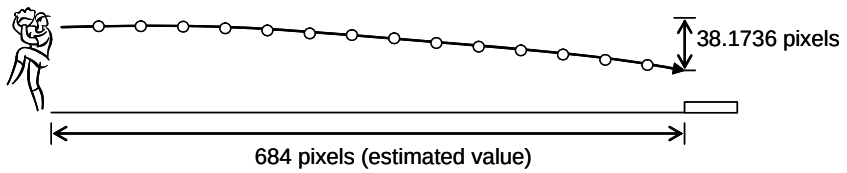


Figure 5-11. An illustration of the relation between the vertical movement and the depth.

For each detected 3-dim trajectory candidate, the included angle between two adjacent vectors is computed. A trajectory candidate is viewed as abnormal if one of the included angles of its flying vectors is larger than 3°. This threshold is set according to the constraint derived from Figure 5-10, and is loosed to cover slight noises caused by detection or tacking errors.

After trajectory validation, the trajectory that conforms to physical principles is retained, and the ball positions in each frame are determined. Figure 5-12 shows an example of the detected trajectory. The proposed trajectory process can be applied to any types of pitching and broadcasting styles, as shown in Figure 5-2.

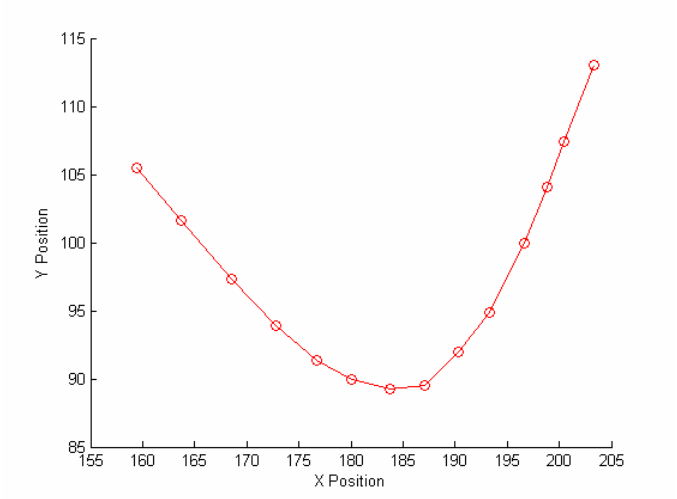


Figure 5-12. The detected ball trajectory

The physical model describes the movement of flying object, which is not affected by any external force except for gravity and air friction. The object that hits another object, e.g. the ball hits the ground or the catcher's glove, would not follow the characteristics described above. Therefore, we focus on extracting the trajectory of the "unaffected" flying object in sports videos.

5.5 Trajectory-based Analysis in Different Sports

The proposed trajectory extraction process can be applied to different ball games, by adjusting some parameters in ball detection and trajectory processing. In this section, we respectively describe the assistance of ball trajectory in analyzing baseball, soccer, and tennis videos. These works suggest new viewpoints for analyzing sports videos.

5.5.1 Pitch Type Recognition in Baseball Videos

In baseball games, the trajectory of a pitching ball indicates the skill of the pitcher. In

addition, the sequential pattern of consecutive pitches implies the pitching tactics. For example, fastball and breaking ball are often dispatched alternately to confuse the batter. On the basis of the extracted ball trajectories, we perform pitch type recognition for consecutive pitches and facilitate game tactics analysis in baseball videos.

5.5.1.1 Pitch Type Recognition

In this work, we mainly focus on recognizing three typical pitching trajectories: fastball, curveball, and breaking ball. Fastball is a straight and very fast pitch, and its trajectory is relatively straight. A fastball is thrown with backspin, so that the Magnus effect produces an upward force on the ball. This counteracts the force of gravity, and causes the ball to follow a flatter trajectory, as shown in Figure 5-13(a).

A curveball is thrown with rotating counterclockwise – as seen from above – by a right-handed pitcher. The Magnus effect produces a downward force on the ball. This force combines with the gravity force to make the ball curve down, as shown in Figure 5-13(b).

There are many other pitch type variations by changing the speed and spin axis. For example, a slider is a kind of fast curveball. It is thrown at a higher velocity than the standard curveball, and will break less than the curveball. Screwball is a kind of reverse curveball, which breaks away from a right-handed batter. These variations and other unmentioned ones, such as change, palmball, knuckleball, splitter, and cut fastball, are roughly categorized as "other breaking balls" in this work. Figure 14 shows the typical trajectories of different pitch types.

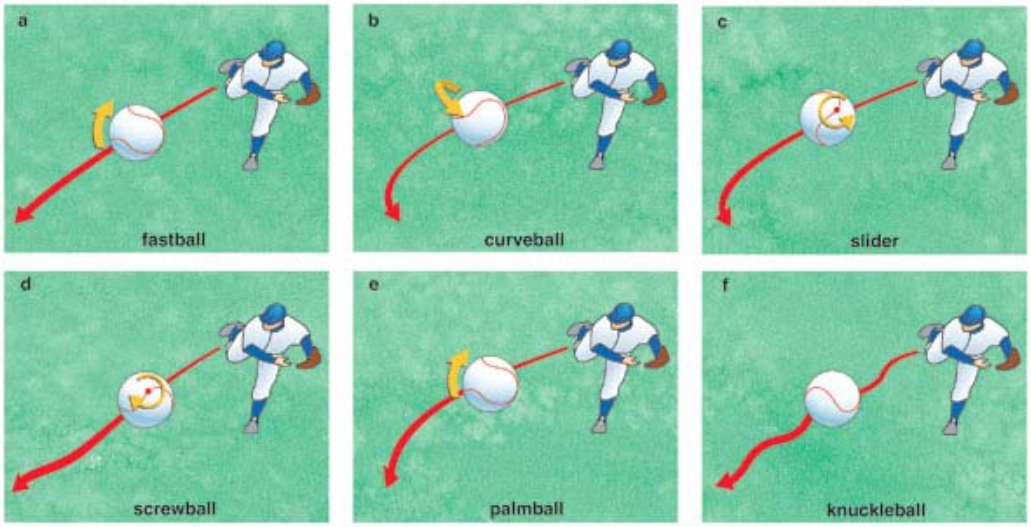
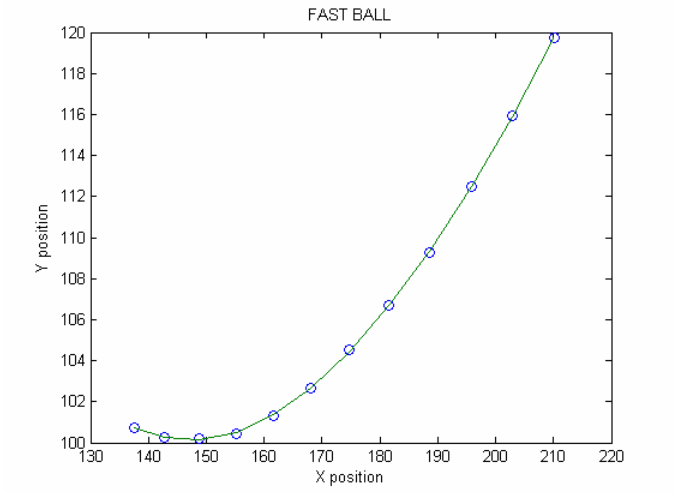
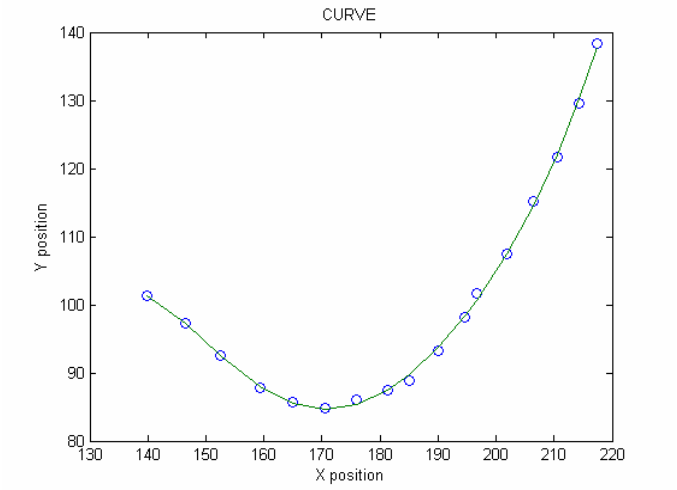


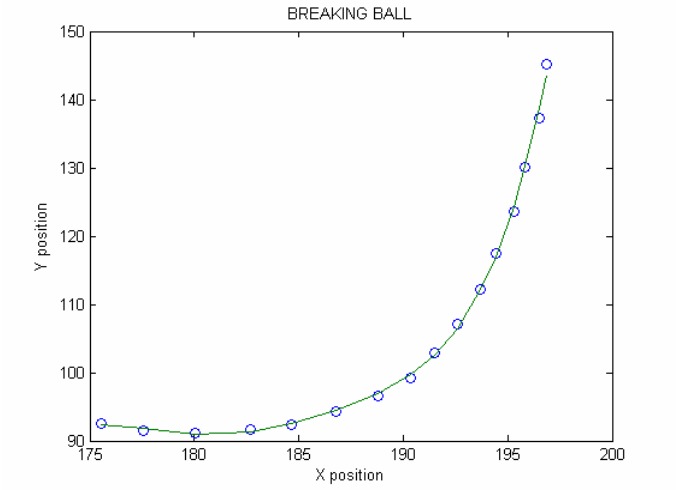
Figure 5-13. Some illustrated examples of different pitch types. (This figure is quoted from [Bahi05])



(a) Fastball



(b) Curveball



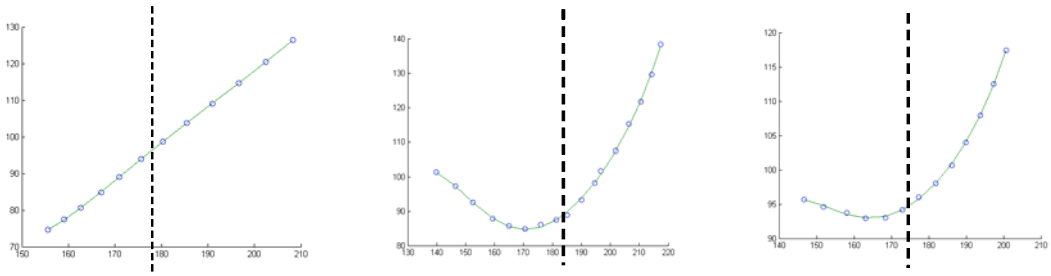
(c) Breaking ball

Figure 5-14. Ball trajectories of different pitch types.

Based on the observations described above, we elaborately design two trajectory-based features to classify a ball trajectory into one of the three types of pitches.

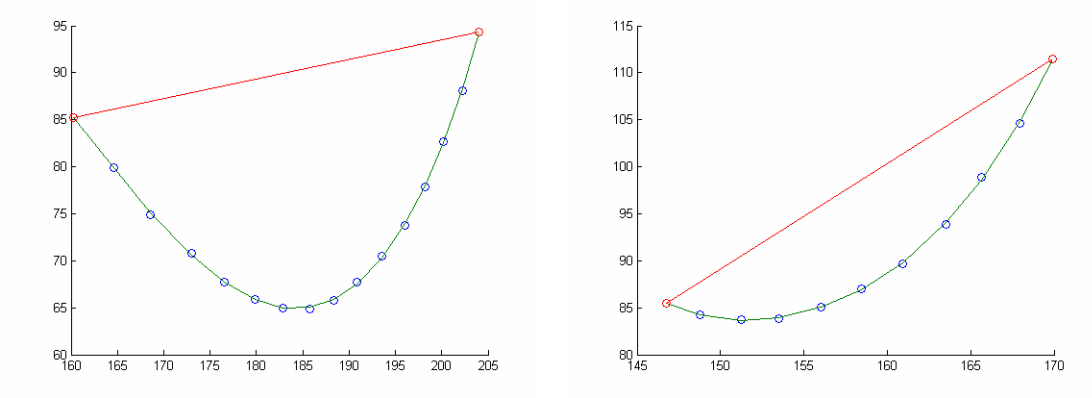
(A) Difference of mean vertical vectors in the fore and the later part of a trajectory (DVV): As compare to curveball and other breaking balls, the variation between the fore part and the later part of a fastball's trajectory is small. We particularly focus on the vertical variations of the ball trajectory. As shown in Figure 5-15, the vertical variation of fastball is significantly different from that of curveball and breaking ball. A fastball has steady vertical movement, but curveball often moves upward in the fore part and drops drastically in the later part. The trajectories of other breaking balls move like the combination of these two pitches, but most of them also drop rapidly in the later part. Therefore, this feature would be a good clue for discriminating fastball trajectory from others.

(B) Area of the arciform region (AAR): The curvatures in curveball and other breaking balls are evidently different. We extract the area of the arciform region, as shown in Figure 5-16, to be the feature for discriminating curveball and other breaking balls. Generally, curveballs have the largest vertical variation among all pitch types.



(a) Fastball (b) Curveball (c) Slider

Figure 5-15. Vertical variations in fastball, curveball, and slider.



(a) Curveball (b) Slider

Figure 5-16. Examples of AAR for curveball and slider.

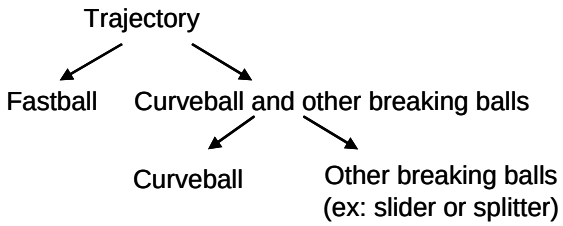


Figure 5-17. The progressive process for pitch type recognition

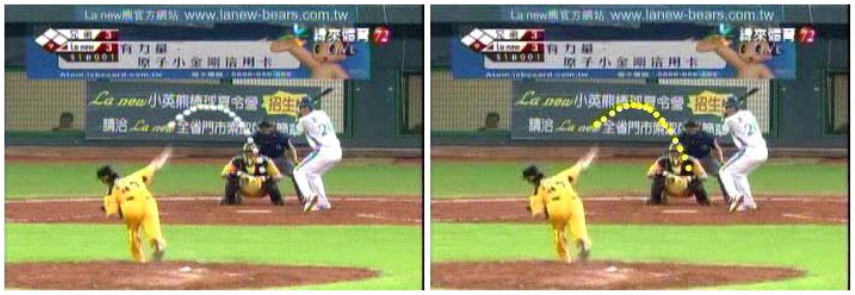
The process of pitch type recognition is illustrated in Figure 5-17. We first distinguish fastball from other types of pitches, and then curveball and other breaking ball are categorized.

5.5.1.2 Evaluation of Trajectory Extraction

We use 38 pitching sequences, including fastball, splitter, slider, and curveball, for evaluation. The ball position of each video frame is manually identified as ground truth. The estimation error, in terms of pixels, between ground truth and the estimated ball positions is calculated. Table 5-2 shows the average and maximal estimation error in four types of pitches. Note that we use 352×240 MPEG-1 bitstreams, and the size of ball usually ranges from 4×4 to 6×6 pixels. Great ball detection accuracy is reported in Table 5-2. Actually, the reported errors are relatively small and can be viewed as the noises derived from the manually defined ground truth. Assume that the error from human remains consistent in different types of pitches, the values in Table 5-2 also show reasonable results. Fastball goes straight and is relatively easy to be estimated, while curveball turns drastically and raises the difficulty of accurate tracking and detection. Splitter and slider, which act between fastball and curveball, have medium performance.

Table 5-2. Extraction performance in terms of estimation error.

Pitch type	Avg. estimation error (pixel)	Max. estimation error (pixel)
Fastball (18 sequences)	0.791	1.73
Splitter (5 sequences)	1.02	1.91
Slider (9 sequences)	1.28	2.48
Curveball (6 sequences)	1.28	2.71



(a) (b)

Figure 5-18. Comparison of (a) the truth ball trajectory and (b) the extracted trajectory

Figure 5-18 shows a trajectory extraction result that juxtaposes the real ball trajectory and the extracted one. More results can be seen at our website (http://www.cmlab.csie.ntu.edu.tw/~wtchu/baseball/index.html).

5.5.1.3 Evaluation of Pitch Type Recognition

About the performance of pitch type recognition, we collect 85 fastball, 11 curveball, 33 slider, and 6 splitter sequences as the evaluation dataset. After extracting the corresponding trajectories, we calculate the corresponding prescribed DVV and AAR, and use Gaussian distributions to describe each pitch type's features characteristics. Table 5-3 shows the statistics of each pitch type. In DVV statistics, DVV of fastballs are significantly different from other types of pitches. Moreover, we can easily differentiate curveballs from other breaking balls through checking AAR. According to DVV statistics in Table 5-3, the threshold for discriminating fastball and others is manually set as 3. On the basis of AAR, the threshold for discriminating curveball and other breaking balls is set as 1000. The probability distributions of DVV and AAR are illustrated in Figure 5-19.

Table 5-4 shows the performance of pitch type recognition. In general, satisfactory performance could be achieved for different pitch types. Only five fastballs are misclassified as breaking balls, three breaking balls are misclassified as fastballs, and two breaking balls are misclassified as curveballs.

Table 5-3. Statistics of DVV and AAR.

Pitch type	DVV (mean)	DVV (std)	AAR (mean)	AAR (std)
Fastball	2.12	0.59	—	—
Curveball	6.25	1.0	1669	418
Slider	3.94	0.58	667	264
Splitter	4.22	0.84	829	230

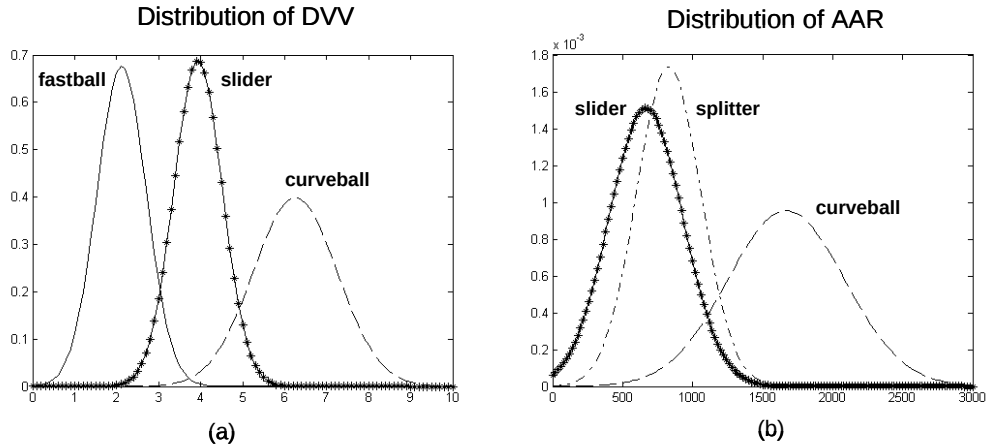


Figure 5-19. Probability distributions of DVV and AAR.

Table 5-4. Performance of pitch type recognition.

Pitch type	Precision	Recall
Fastball	96.38	94.12
Curveball	84.62	100
Breaking balls	87.18	87.18

Collecting a series of pitches from a specific match, we may employ a sequential mining method to discover pitching tactics of a team's pitchers. Moreover, pitches thrown by different pitchers' or by the same pitcher at different time may act a little different. Carefully examine how the ball moves would help a pitcher improve his skill. Overall, trajectory extraction in baseball videos provides a new medium for automatic analyzing baseball games.

5.5.2 Penalty Kick Analysis in Soccer Videos

5.5.2.1 Soccer Trajectory Extraction

Yu et al. [Yu03] proposed a system for soccer video analysis through ball trajectory. By checking ball's moving speed and goalmouth detection, "shot" events can be detected. Moreover, by checking the direction of the moving ball, ball possession time

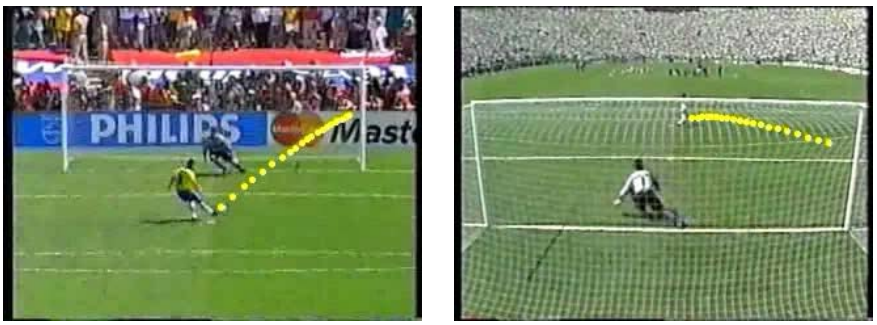
of each team can be estimated.

With the help of trajectory extraction, we analyze soccer videos from different viewpoints. We focus on the sequences that contain penalty kicks. In soccer games, the goalkeeper should setup his strategy (either jumps right or left) to hold up the ball. The moving relationship between the goalkeeper and the ball often grasps the eyesight of the audience, and is extremely important to the game results. Through gathering the trajectory results of a series of penalty kick videos, we can mine the keeping strategy of the goalkeeper, such as how he reacts to a ball kicked by a right-footer.

The proposed Kalman filter-based approach can be employed in extracting soccer trajectory as well. While the procedure of trajectory extraction is the same as that for baseball, some parameters should be modified, such as ball size and ball color. In general, because the angle of flying soccer varies significantly in penalty kicks, we have to loose the constraints of reasonable position described in Section 5.3. Table 5-5 shows the parameters we used in soccer trajectory extraction, in which no position constraint is set. The major challenge to extract a soccer trajectory is that the included angle between the ball trajectory and the middle line of the screen can vary significantly. It's not the case in an ordinary baseball trajectory.

Table 5-5. Parameters in soccer trajectory extraction.

Ball candidate detection	
Color filter	RGB values > 100
Position filter	None
Size filter	10 pixels ≤ object size ≤ 100 pixels
Shape filter	Ratio > 0.2



(a) (b)

Figure 5-20. Examples of trajectory extraction for penalty kick in soccer videos.

Figure 5-20 shows sample results of trajectory extraction in two different penalty kick sequences, which are captured from two different cameras. These results demonstrate the feasibility of the proposed approach.

The extracted trajectory information greatly helps in analyzing the tactics of the goalkeeper. We can characterize the habitual behavior of a specific goalkeeper by checking the moving relationship between him and the flying ball. More practical applications based on soccer ball trajectory can be developed in the future.

5.5.2.2 Evaluation of Soccer Trajectory Extraction

We take the penalty kick series in "World Cup 1994: Brazil vs. Italy" as the evaluation data. The resolution of video frame is 320×240. After evaluating the four different kicks in the series (two are captured from the front side of the goalmouth, and two are captured from the back side), the average estimation error is 1.09 pixels, and the maximal estimation error is 2.74 pixels. They are slighter larger than that in baseball trajectory extraction. We address that the following challenges are specific for soccer videos and probably the causes of larger estimation errors.

(1) Estimation of soccer trajectory is harder than baseball because different trajectories may have significantly different flying directions. Generally, a kick causing a goal means the ball flies into the region of goalmouth (7.32 meters × 2.44 meters), which is extremely larger than the strike zone in baseball.

(2) Penalty kick videos are often suffered from noises caused by the audience, as shown in the upper regions of Figure 5-19(a) and Figure 5-19(b). White moving background objects, which are derived from the movement of the audience, often cause a plausible trajectory and make confusion in the trajectory forming process. On the other hand, the white background objects, which are derived from advertisement boards, are often still in a pitching baseball sequence.

(3) For penalty kick videos, the ball size changes drastically when the ball flies from the penalty mark to the goalmouth. Seeing from the back side of the goalmouth, the ball size may vary from 4×4 pixels to 10×10 pixels within a period less than one second.

5.5.3 Tactics Analysis in Tennis Videos

5.5.3.1 Tennis Trajectory Extraction

Likewise, the ball trajectory also plays an important role in tennis games. Through checking the pattern of trajectory, many hidden information can be found, such as:

(1) Ground strokes: two players move along the base line and keep stroking the ball until the opponent is out of position.

(2) Approach and volley (smash): a player returns the ball, and then approaches to the net immediately to prepare for a volley (smash).

(3) Passing shot: when the opponent approaches to the net, the player returns the ball at the opposite direction so that the opponent cannot successfully return the ball.

These events are often the most interesting parts of a tennis game. They can be detected by checking the position relationship between the ball trajectory and the players.

On the basis of the proposed trajectory extraction approach, we are able to automatically extract tennis trajectory and facilitate more advanced tennis video analysis. Figure 5-21 shows two sample results of trajectory extraction in a tennis game.

(a) (b)

Figure 5-21. Examples of trajectory extraction for tennis videos.

The major challenge to extract a tennis trajectory is that the size of tennis ball is smaller than baseball; moreover, it flies faster than baseball (for man players, the moving speed of tennis ball is 180 km/hr in average). Therefore, not only the ball color and size are different, we also have to adjust the parameters of the tracking module such that reasonable trajectory segments can be concatenated. In our implementation, we extract tennis trajectories in MPEG-1 video sequences, with 560×416 resolution. Table 5-6 shows the parameters for tennis trajectory extraction.

Table 5-6. Parameters in tennis trajectory extraction.

Ball candidate detection	
Color filter	RGB values > 100
Position filter	None
Size filter	2 pixels ≤ object size ≤ 10 pixels
Shape filter	Ratio > 0.3

5.5.3.2 Evaluation of Tennis Trajectory Extraction

We compare trajectory extraction results of six different sequences with manually defined ground truths. The evaluation data is from "2005 US Open Quarterfinal: Andre Agassi vs. James Blake". The average estimation error is 1.84 pixels, and the maximal estimated error is 11.7 pixels. The estimation errors of tennis trajectories are generally larger than baseball trajectories. We address that the following challenges are specific for tennis videos and probably the causes of larger estimation errors.

(1) Tennis is smaller but flies faster than baseball, and sometimes even human cannot recognize. Therefore, we often miss the real ball object in the ball candidate detection stage. It's often the case that we have to interpolate the missing positions between two trajectory segments. However, because of the shooting angle, the ball's moving speed is different at the upper part and the bottom part of screen. The changeable speed makes accurate interpolation nontrivial and may cause larger estimation errors.

(2) The resolution of evaluated video sequences is larger, and sometimes the position ground truth of deformed balls is hardly defined.

Overall, the low estimate errors again demonstrate the accuracy of the proposed trajectory extraction process.

5.6 Discussion and Summary

We developed a system to automatically extract ball trajectory from sports videos. By checking color, position, size, and shape information, ball candidates in each video frame are detected. The Kalman filter-based approach is applied to track the ball position and generate trajectory segments. On the basis of trajectory segments, a process is designed to generate trajectory candidates, which last in the whole video sequence. We evaluate the reasonability of each trajectory and obtain the final trajectory result based on a physical model-based validation method. The proposed approach can be applied to different sports, by adjusting the constraints of ball color, ball size, and reasonable region.

The trajectory information can significantly aid in sports video analysis. It provides a new type of metadata or be the clues for detecting subtle concepts, such as pitch type of a pitching baseball or offense tactics in tennis videos. In baseball videos, we elaborately design two features based on the ball trajectory, and perform pitch type recognition to distinguish between fastball, curveball, and other breaking balls. The experimental results show that we have only few errors in trajectory extraction and achieve high detection accuracy in pitch type recognition. In soccer and tennis videos,

the same approach can be employed to extract ball trajectories so that more advanced video analysis can be favored.

The primary idea corresponds to the framework described in Chapter 2 and is illustrated in Figure 5-22. Based on color, size, shape, and region features, a heuristics-based approach is applied to detect ball candidates in each frame. From ball candidates to trajectory, we concatenate ball candidates to generate reasonable trajectories by applying a tracking method and physical model-based validation. In this work, ball trajectories play the role of mid-level representation, and are the foundations for more advanced concept detection. From trajectories to concepts, such as fastball and curveball, we develop a heuristics-based approach to recognize pitch types.

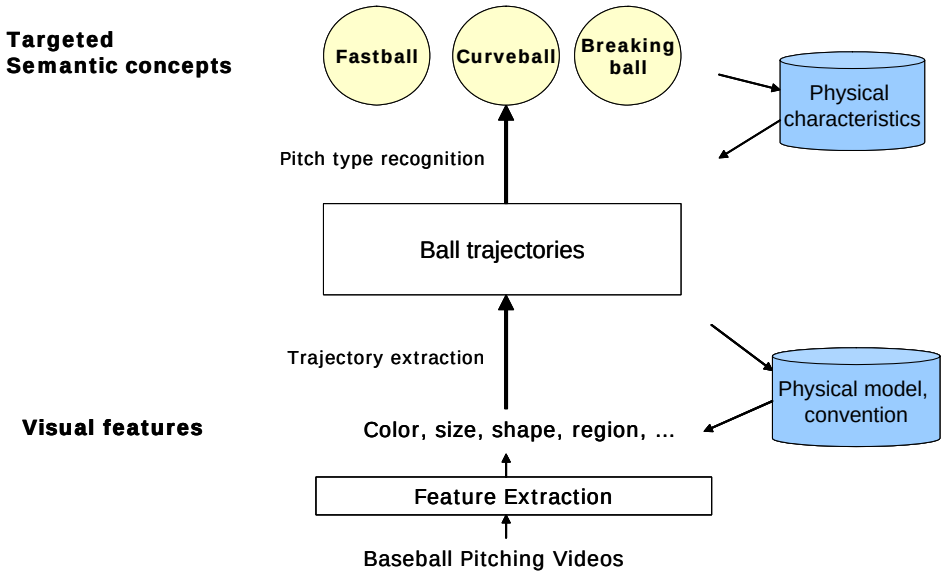


Figure 5-22. Trajectory extraction in terms of the framework described in Chapter 2.

In the future, we plan to recognize more types of breaking balls, such as slider and splitter. A sequential mining module can be applied to find subtle pitching patterns for a specific pitcher. Likewise, the same idea can be applied in deriving goalkeeper's defensive strategies. By checking the relative position between players and tennis ball trajectory, many interesting events can be automatically detected. In other words, the results of ball trajectory extraction can be applied to either entertainment or game analysis.

Chapter 6
Future Research and Conclusions

6.1 Discussions

We would like to have some discussions about cooperating semantic content analysis with other disciplines before concluding this dissertation. Content adaptation and multimedia communication applications are the main subjects of this discussion.

6.1.1 Content Adaptation Architecture

With the drastic advances of video coding and transmission technologies, various multimedia communication applications such as mobile TV and video-on-demand services mushroomed in recent years. Accompanying with the popularity of digital TV and digital video broadcasting [Reim06], more and more high-definition visual content are produced and streamed over heterogeneous networks. To alleviate the inefficiency of content dissemination or usage, we have the urgent needs of elegant content analysis and adaptation techniques to facilitate intelligent manipulation of multimedia communications.

The goal of content analysis is to scrutinize or classify digital content such that browsing, managing, presenting, and disseminating content could be efficient and/or effective. For example, with structural analysis, video adaptation [Chan05] applications such as video summarization and transcoding are developed to manipulate information more flexibly.

Although content analysis techniques are widely studied in recent years, relatively little attention has been paid to jointly consider the issues of content analysis and multimedia communication. In this section we focus on this interdisciplinary subject and propose some integrated applications where multimedia communication and content analysis are collaborated with each other, seamlessly. A content repurposing framework that facilitates scalable delivery and differential services is proposed to demonstrate the mutual impacts of content analysis and media communications. Then we discuss the cooperative tasks in terms of two phases: 1) content-aware multimedia communication applications and 2) content-aware multimedia transmission architecture. Several convincible integrated approaches are proposed and some possible further developments are also discussed.

Figure 6-1 shows the overall content repurposing architecture. The incoming digital content is first dissected by various content analysis techniques. What kind of analysis should be applied is decided by the capability of clients and the target of applications. We call the content after analysis "*structured content*" since it has been well classified, indexed, or organized. Users can browse the structured content and query specific portions of it, directly. With the aid of organization functionality, users can easily obtain a content summary that reserves important information within a shorter duration. This content-aware application scenario eases the burden and reduces the cost of content transmission. For example, it's expensive and time-consuming to transmit a complete sports video to users, while a content summarization module is able to shrink a lengthy game video into a short summary or highlight video clips such that less transmission time and bandwidth are required.

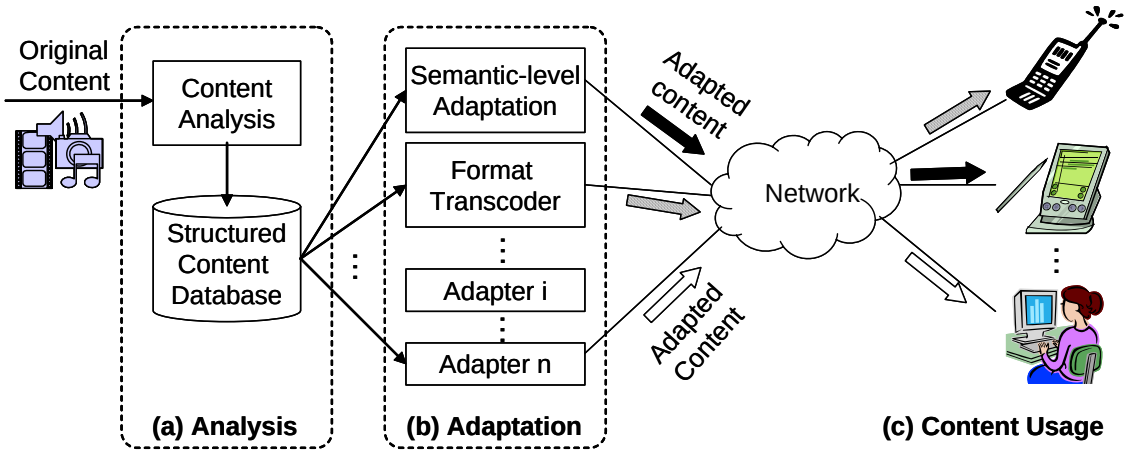


Figure 6-1. Overall architecture of the content adaptation process.

The proposed architecture provides a realistic instance of universal multimedia access pursued by MPEG-21 [Burn03], which draws a picture of transparent digital data usage through taking resource and description adaptation to match characteristics of users, terminals, and networks. With content analysis techniques, a variety of repurposing processes can be performed to build fruitful multimedia communication applications.

6.1.2 Content Adaptation Modeling

Content adaptation can be modeled as a resource allocation problem [Moha99] as follows:

$$\max\left\{\sum_i V_i\right\} \text{ such that } \sum_i R_i \le R_{client}\,, \tag{6-1}$$

where V_i and R_i are the values and resources used by the ith adapted content item

M_i, and R_{client} is the maximum resource available at the client. Mohan et al. [Moha99] proposed this general model and adapted internet content (web pages) for universal access. However, how to appropriately evaluate the value of content, which affects the adapted results most, is not well defined. While previous works assumed that the content value V_i is some function of the resource R_i, we argue that it's more reasonable to take content semantics into account.

Accompanying with the advance of content analysis, we are able to explicitly uncover embedded content semantics and clearly develop content value estimation functions. The content value V_i of the adapted content item M_i can be defined as:

$$V_i = \sum_{j=1}^{J} w_j \times f_j\left(M_i\right), \tag{6-2}$$

where f_j is a specific mapping function that evaluates the content value of M_i. Different mapping functions can be defined to represent values of various types of content. For example, if M_i is a video clip representing a baseball event, a function f_j can be defined according to the contribution (or importance) of its appearance to the game, which implies f_j("a hit causing score") > f_j("an out"). Other mapping functions can be defined as well, say according to user preference and consuming bit-rates. The integrated content value V_i of the content item M_i is then calculated by linearly combining content values derived from different perspectives. The weighting parameters w_j can be determined heuristically or learnt from subjective evaluation results.

The original content (e.g. a baseball video) can be indexed by a content analysis module (e.g. a concept detection module) so that the content is segmented into short-term content items (e.g. baseball concepts), as shown in Figure 6-2. After content values evaluation, we sort them in descending order and estimate the corresponding consumed resources by calculating each item's data size. To solve the constrainted resource allocation problem, we can just apply a greedy selection method such that the total content values are maximal, subject to the client's resource constraint. After we determine how many and which events should be selected, they are concatenated according to the temporal order of the original content.

The combination of the prescribed adaptation processes enables the flexibility of content presentation. We call the data after repurposing *adapted content* in this work. As shown in Figure 6-1, the adapted content is generated by the efforts of analysis and repurposing techniques. The whole processes in Figure 6-1(a) and Figure 6-1(b) can be viewed as presenting the same content with different appearances (e.g. video resolutions in 1024×768 vs. 352×240) or information coverage (e.g. audio only vs. video).

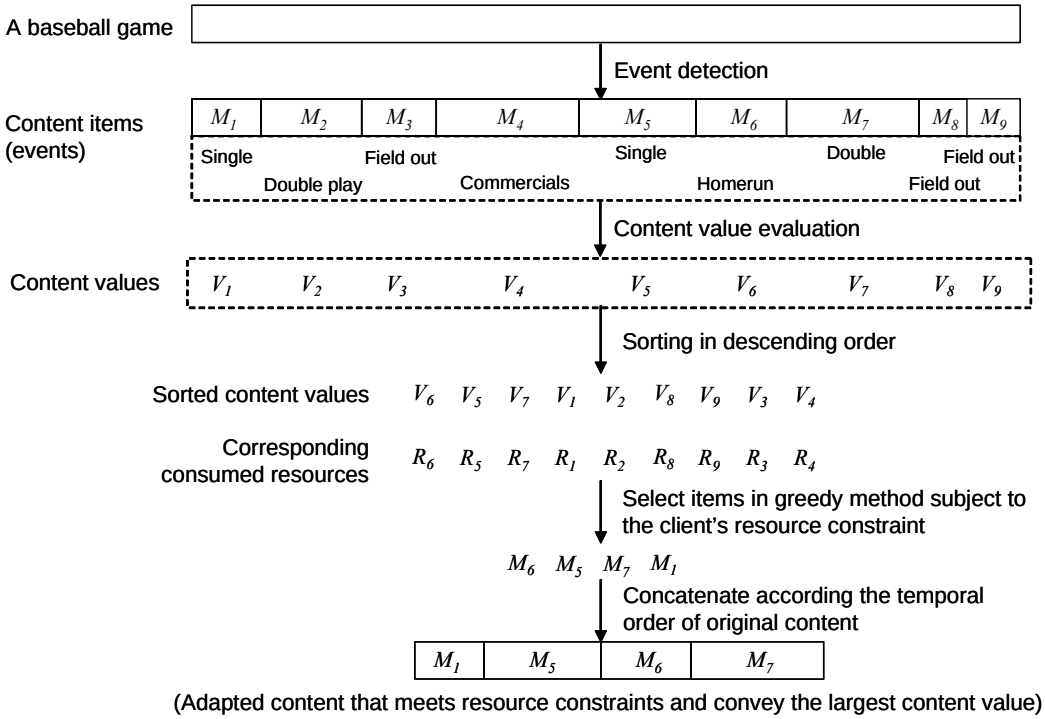


Figure 6-2. An example process of content adaptation.

We present the idea of collaborating content analysis technologies with multimedia communication issues. We suggest that semantics analysis facilitate the evaluation of content value, which was estimated by an over-simplified assumption in the conventional content adaptation model. Based on the content adaptation techniques, several examples are given to show the tight collaboration between content analysis and media communications.

6.2 Future Research

In addition to the works described in this dissertation, multimedia semantic analysis remains a very challenging problem in general domain media. Many related issues are still left for long-term research.

How to define the analysis levels for different cases is still an open issue. In our works, we empirically define analysis levels and develop techniques to bridge them, according to the domain knowledge or observation. There may be other ways, such as machine learning or pattern analysis, to determine the semantic granularities.

General semantic concept detection is a seriously challenging problem. Many methodologies have been taken to model the relationships between semantic concepts, but none of them is always superior to others. Advances in feature extraction, feature design, pattern classification, data mining, ontology, or knowledge management would provide more supports on this topic. Multimedia semantic analysis researches

certainly remain ongoing.

Collaborating multimedia content analysis with other research fields would arouse many interesting studies, such as the multimedia communication applications described in Section 6.1. Actually, studies on content analysis often rely on the supports from computer vision, machine learning, and pattern recognition researches. The interdisciplinary nature of content analysis makes it even appropriate to cooperate with other fields.

6.3 Conclusions

In this dissertation, we have presented a framework that serves as a guideline for multimedia semantic analysis. We respectively study visual and aural characteristics in different media, and develop corresponding techniques to bridge low-level features and high-level semantics. The contributions of this dissertation are summarized as follows.

- Multilevel framework: The proposed multilevel framework conceptually analogizes multimedia semantic analysis to the process of language learning. Mapping functions that bridge different levels can be flexibly implemented according to the content characteristics.
- Semantic concept detection in baseball games: With the help of clear domain knowledge and elaborate visual analysis, we develop a system to explicitly detect concepts in broadcasting baseball videos and build realistic applications. Moreover, a novel attempt on visual content is to analyze by extracting the ball trajectory.
- Semantic concept detection in movies: In terms of nondeterministic relationship and aural perspective, we develop a system that statistically models semantic concept such as gunplay and car chasing in movies. Two types of models, i.e. generative and discriminative models, are implemented to conduct this study.

These studies would be essential parts of future multimedia information retrieval, digital asset management, and digital home applications. The attempts we made could be the foundations of content-aware systems or human-centric functionalities.

This dissertation describes systematic studies on multimedia semantic analysis. Nonetheless, the semantic gap problem that is involved with many non-mechanical factors is far from solved. Human perception and cognition, personal style of the video producer, and the insufficiency of recent computational methodologies affect the effectiveness of semantic analysis. We believe that the related researches will

attract much more attention in the near future, and bridging the semantic gap would significantly change our experience in the digital world.

Appendix A
Hidden Markov Model

Hidden Markov model has been successfully applied in speech recognition for a few decades. Many variations have also been developed for other research fields, such as computer vision and image/video analysis. An HMM is a statistical model where the system being modeled is assumed to be a Markov process with unknown parameters. The major challenge of using an HMM is to estimate the hidden parameters from observable data. In this section, we briefly introduce the formulation of HMM and describe the training and testing issues.

A.1 Specification

A hidden Markov model λ consists of the following parameters [Rabi89].

1. N, the number of states in the model. The individual states are labeled as {1, 2, …, N}, and the state at time t is denoted as q_t.
2. M, the number of distinct observation symbols in all states. The individual symbols are denoted as $V=\{\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_M\}$.
3. The state transition probability distribution $A=\{a_{ij}\}$, where a_{ij} is the probability of taking a transition from state i to state j, i.e.
   $a_{ij} = P[q_{t+1}=j \mid q_t=i],\ 1 \leqq i, j \leqq N.$ (A-1)
4. The observation probability distribution $B=\{b_j(k)\}$, where $b_j(k)$ is the probability of emitting symbol $\boldsymbol{v}_k$ when state j is entered. Let $\mathbf{O}=(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t)$ be the observed output of the HMM. The state sequence $\mathbf{Q}=q_1, q_2, \ldots, q_t$ is not observed (hidden), and $b_j(k)$ can be written as:
   $b_j(k) = P[\mathbf{o}_t=\boldsymbol{v}_k \mid q_t=j],\ 1 \leqq k \leqq M,\ 1 \leqq j \leqq N.$ (A-2)
5. The initial state distribution $\pi=\{\pi_i\}$ in which
   $\pi_i = P[q_0=i],\ 1 \leqq i \leqq N.$ (A-3)

Since a_{ij}, $b_j(k)$, and π_i are all probabilities, they must satisfy the following properties:

$$a_{ij} \geq 0,\ \ b_j(k) \geq 0,\ \ \pi_i \geq 0 \quad \forall \text{ all } i, j, k,$$

$$\sum_{j=1}^{N} a_{ij} = 1,\ \ \sum_{k=1}^{M} b_j(k) = 1, \text{ and } \sum_{i=1}^{N} \pi_i = 1.$$

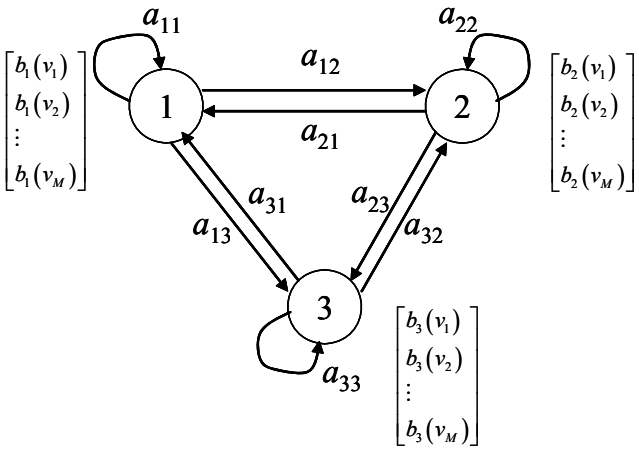


Figure A-1. An example of a 3-state ergodic HMM.

On the basis of this specification, Figure A-1 shows a 3-state ergodic HMM, which means any state can transit to any other state. In speech recognition, we usually employ left-right HMMs, in which no transitions are allowed to states whose indices are lower than that of the current state.

A.2 Inside HMM

Given the form of HMM, three basic problems have to be discussed before they can be applied to real-world applications.

- **The Evaluation Problem**: Given a model λ and a sequence of observations $\mathbf{O}=(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$, how do we efficiently compute $P(\mathbf{O}|\lambda)$, i.e. the probability that the model generates the observations?
- **The Decoding Problem**: Given a model λ and a sequence of observations, what is the most likely state sequence $\mathbf{Q}=(q_1, q_2, \ldots, q_T)$ in the model that produces the observations?
- **The Learning Problem**: Given a model λ and a set of observations, how can we adjust the model parameter to maximize $P(\mathbf{O}|\lambda)$?

The following subsections briefly describe the widely accepted methods for solving these problems.

A.2.1 Solution to the Evaluate Problem — The Forward Algorithm

To compute $P(\mathbf{O}|\lambda)$, we first enumerate all possible state sequences $\mathbf{Q}$ of length T, which generate the observation sequence $\mathbf{O}$, and then sum all the probabilities. The probability of each state sequence is the product of the state sequence probability and the joint output probability. Given an HMM, the state sequence probability denotes the occurrence probability of a specific state sequence. Given an HMM and a specific state sequence, the joint output probability denotes the occurrence probability of a specific observation sequence. That is,

$$\begin{aligned} P(\mathbf{O}|\lambda) &= \sum_{\text{all } \mathbf{Q}} P(\mathbf{Q}|\lambda) P(\mathbf{O}|\mathbf{Q},\lambda) \\ &= \sum_{\text{all } \mathbf{Q}} \left(\pi_{q1} a_{q1q2} a_{q2q3} \cdots a_{qT-1qT}\right) \times \left(b_{q1}(\mathbf{o}_1) \cdot b_{q2}(\mathbf{o}_2) \cdots b_{qT}(\mathbf{o}_T)\right). \end{aligned} \tag{A-4}$$

However, direction evaluation of the equation (A-4) requires enumeration of $O(N^T)$ possible state sequences, which results in exponential computational complexity. Therefore a more efficient algorithm, i.e. *forward algorithm*, is designed for the evaluation problem. The trick is to store intermediate results and use them for subsequent state-sequence calculations to save computation.

The *forward probability* $\alpha_t(i)$ is defined as

$$\alpha_t(i) = P(\mathbf{o}_1 \mathbf{o}_2 ... \mathbf{o}_t, q_t = i|\lambda). \tag{A-5}$$

This denotes the probability that the HMM is in state i at time t having generated partial observation sequence, $\mathbf{o}_1$, $\mathbf{o}_2$, …, $\mathbf{o}_t$. We can solve for $\alpha_t(i)$ inductively by the forward algorithm:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), \quad 1 \le i \le N. \tag{A-6}$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}), \quad 1 \le j \le N, \ 1 \le t \le T-1. \tag{A-7}$$

3. Termination

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{A-8}$$

The core of this algorithm is the induction step. The forward probability, $\alpha_{t+1}(j)$, is the product of observation probability, $b_j(\mathbf{o}_{t+1})$, and the summation of the forward probabilities over all the N possible states before time t. This algorithm requires $O(N^2T)$ calculations and efficiently evaluate an HMM.

A.2.2 Solution to the Decoding Problem — The Vertibi Algorithm

To find the best state sequence, $\mathbf{Q}=(q_1, q_2, \ldots, q_T)$, for the given observation sequence, $\mathbf{O}=(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$, we define the quantity

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P\left[q_1 q_2 \ldots q_{t-1}, q_t = i, \mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_t \,|\, \lambda\right], \tag{A-9}$$

where $\delta_t(i)$ denotes the highest probability along a single path, which accounts for the first t observations and ends in state i. By induction we have

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] \cdot b_j(\mathbf{o}_{t+1}). \tag{A-10}$$

Based on these definitions, the Viterbi algorithm that solves the decoding problem proceeds as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \quad 1 \le i \le N, \tag{A-11a}$$

$$\varphi_1(i) = 0. \tag{A-11b}$$

2. Recursion

$$\delta_t(j) = \max_{1 \le i \le N}\left[\delta_{t-1}(i) a_{ij}\right] b_j(\mathbf{o}_t), \quad 1 \le j \le N, \quad 2 \le t \le T, \tag{A-12a}$$

$$\varphi_t(j) = \arg\max_{1 \le i \le N}\left[\delta_{t-1}(i) a_{ij}\right], \quad 1 \le j \le N, \quad 2 \le t \le T. \tag{A-12b}$$

3. Termination

$$P^* = \max_{1 \le i \le N}\left[\delta_T(i)\right], \tag{A-13a}$$

$$q_T^* = \arg\max_{1 \le i \le N}\left[\delta_T(i)\right]. \tag{A-13b}$$

4. Path backtracking

$$q_t^* = \varphi_{t+1}\left(q_{t+1}^*\right), \quad t = T-1, T-2, \ldots, 1, \tag{A-14}$$

$\mathbf{Q}^* = \left(q_1^*, q_2^*, \ldots, q_T^*\right)$ is the best sequence.

The major difference between the Vertibi algorithm and the forward algorithm is the maximization in Eq. (A-12a) over previous states, which is used in place of the summing procedure in Eq. (A-7). The kernel idea of this algorithm is like dynamic programming, which stores the optimal sub-solutions as it sees partial observations, and backtracks to find the optimal path.

A.2.3 Solution to the Learning Problem — Baum-Welch Algorithm

The most difficult problem of HMMs is to adjust the model parameters (A,B,π) for maximizing the likelihood $P(\mathbf{O}|\lambda)$, by giving a model and a set of observations. The Baum-Welch algorithm (also known as the forward-backward algorithm) based on expectation-maximization (EM) strategy is developed to solve this problem. The main idea is to iteratively refine the HMM parameters by maximizing the likelihood $P(\mathbf{O}|\lambda)$ at each iteration. Because the procedure is complicated and is not appropriate to be addressed in the appendix, we leave the details in the excellent survey [Rabi89]. More extensive HMM-based studies on speech recognition can be seen in the books of [Rabi93] and [Huan01].

In the works of semantic concept detection in movies, we employ HMMs to model audio events and semantic concept. In audio event model/detection, audio features, such as zero-crossing rate and sub-band energy ratio, are the observations $\mathbf{O}$. By the forward algorithm, we detect an audio event by evaluating the likelihood $P(\mathbf{O}|\lambda_i)$, where i is the index of audio models. Likewise, in the semantic concept model/detection, the pseudo-semantic features are the observations. The same evaluation method is applied to audio event detection and semantic concept detection, while the basis observations are different.

Many packages based on C [HTK][GHMM] and Matlab [Murp05][Ghah06] can be downloaded for rapidly constructing your own applications. In our works, we employ the easy-to-use Matlab package developed by Kevin Murphy [Murp05].

Appendix B
Support Vector Machine

This section briefly describes the idea of support vector machine (SVM) in terms of training and testing. The introductory content is mainly quoted from [Hsu06] and [Well05]. Extensive studies can be seen in [Vapn98] and [Plat00].

B.1 Introduction

In recent years, support vector machine has been proved to be a powerful tool for data classification. A classification task is usually involved with training and testing data that consist of some data instances. Each data instance in the training set contains one class label and several features. The goal of SVM is to produce a model which predicts the labels of data instances by giving features.

Given a training set that consists of instance-label pairs (x_i,y_i), i=1,2,…,N, where $x_i \in \mathbb{R}^d$ and $y_i \in \{+1,-1\}$, the task is to predict whether a test sample belongs to one of two classes (1 or -1). Let's consider a very simple example that can be linearly separated: We can draw a straight line $f(\boldsymbol{x})=w^{\mathrm{T}}\boldsymbol{x}$-b such that all cases with y_i=-1 fall on one side and have $f(x_i)<0$ and cases with y_i=+1 fall on the other and have $f(x_i)>0$. If we can find such straight line that discriminates two classes of data, we can classify new test cases according to the rule $y_{\text{test}}=\text{sign}(x_{\text{test}})$.

With the above definition, we can write down the following constraint that any solution must satisfy

$$w^T x_i - b \le -1 \quad \forall y_i = -1, \tag{B-1}$$

and $$w^T x_i - b \ge +1 \quad \forall y_i = +1, \tag{B-2}$$

or in one equation

$$y_i \left(w^T x_i - b \right) - 1 \ge 0. \tag{B-3}$$

We now formulate the primal problem of the SVM:

$$\min \frac{1}{2}\|w\|^2$$
$$\text{subject to } y_i\left(w^T x_i - b\right) - 1 \geq 0 \quad \forall i. \tag{B-4}$$

By this formulation, we maximize the margin, subject to the constraints that all training cases fall on either side of the boundary. The vectors that lie on the boundary are called support vectors, since they support the decision boundary and determine the solution to the problem.

The formulation of SVM can be further generalized as follows:

$$\min\left(\frac{1}{2}\omega^T\omega + C\sum_{i=1}^{N}\zeta_i\right)$$
$$\text{subject to } y_i\left(\omega^T\phi\left(x_i\right) + b\right) \geq 1 - \zeta_i, \quad \zeta_i \geq 0. \tag{B-5}$$

The training vectors x_i are mapped to a higher dimensional space by the function ϕ. We want to find a linear hyperplane with the maximal margin to separate two classes of data, in the higher dimensional space. In order to prevent overfitting, we loose the optimization constraints and allow some misclassification in training, with some penalty C (C>0) on the error terms ζ_i.

Figure B-1 shows a 2-dimensional illustration of the SVM classifier. There would be infinite decision boundaries to separate two classed of data. The optimal one we want is the boundary with maximal margin, allowing some misclassified cases to prevent overfitting. The determined decision boundary that has the maximal margin to two data instances is expected to have better discriminative power in testing.

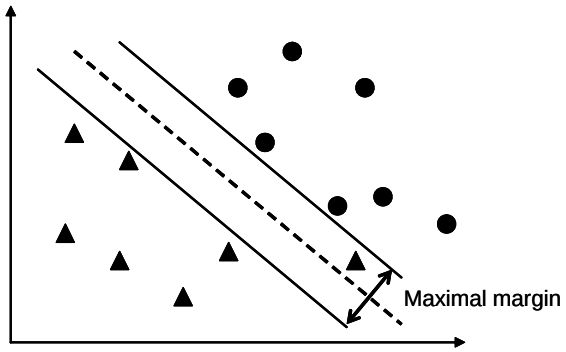


Figure B-1. A 2-dimensional illustration of the SVM classifier.

B.2 Training and Testing

In the training stage, two factors have to be determined: (1) the function that maps data instances to higher dimensional space; and (2) the parameters of the hyperplane that facilitates the minimal training error.

For the first problem, we define the kernel function as $K\left(x_i, x_j\right) \equiv \phi\left(x_i\right)^T \phi\left(x_j\right)$ to present different types of mapping functions. Some classical kernels are:

- Liner: $K\left(x_i, x_j\right) = x_i^T x_j$.
- Polynomial: $K\left(x_i, x_j\right) = \left(\gamma x_i^T x_j + r\right)^d, \quad \gamma > 0$.
- Radial basis function (RBF): $K\left(x_i, x_j\right) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \quad \gamma > 0$.
- Sigmoid: $K\left(x_i, x_j\right) = \tanh\left(\gamma x_i^T x_j + r\right)$.

Here, γ, r, and d are kernel parameters. According to the study in [Hsu02], radial basis function is preferred to being the kernel function.

Once we determine the kernel function, we have to find appropriate parameters to find the hyperplance that has maximal margin to two classes of data. There are two parameters in using RBF kernels: C and γ. The goal of training is to identify good (C, γ) pair (based on training data) so that the classifier can accurately predict unknown data, i.e. testing data. To reliably adjust the parameters, we usually separate training data to v subsets. One of the subset is considered unknown and is tested using the classifier trained on the remaining v-1 subsets. Then the prediction accuracy on this set can more precisely reflect the performance on classifying unknown data. This training method is called v-fold cross-validation. The cross-validation procedure prevents the overfitting problem and is widely adopted in training a reliable classifier.

In the testing stage, the SVM classifier just takes the input vector and evaluates which side the test vector is located. It returns the label by checking the sign after evaluation.

B.3 Multiclass SVM

As mentioned above, SVM classifiers generally perform binary classification. However, multiclass classification should be achieved in many real-world applications, including the semantic concept detection work described in Chapter 3. To match this demand, many variations that combine several binary classifiers into a multiclass classifier have been developed.

One of the variations for multiclass classification is one-against-all method. It constructs k SVM classifiers, where k is the number of classes. The ith SVM is trained on the basis of the data instances in the ith class with positive labels, and all instances in other classes with negative labels.

The second variation is one-against-one method. This method constructs $k(k\text{-}1)/2$ SVM classifiers, where each one is trained based on the data from two classes. In testing, voting strategy is often used to decide the label of the test vector. Assume that we test on a SVM classifier that is trained for distinguishing class *A* and class *B*. If the classifier says the test vector is in *A*, then vote for class *A* is added by one. Otherwise, vote for class *B* is added by one. After checking all the $k(k\text{-}1)/2$ classifiers, the class that has the largest votes is the answer.

The third method for multiclass classification, which is used in our work, is the directed acyclic graph SVM (DAGSVM). It has the same training process as the one-against-one method, thus $k(k\text{-}1)/2$ classifiers are constructed. In the testing phase, it uses a rooted binary directed acyclic graph which has $k(k\text{-}1)/2$ internal nodes and k leaves. Each node is a binary one-against-one SVM classifier. Given a test vector, we start from the root node and evaluate whether the flow should move left or right according to the output value of root node. The process keeps going until it reaches a leaf node, which indicates the predicted class.

According to the studies in [Hsu02], we apply the DAGSVM [Plat00] to classify gunplay, car-chasing, and other concepts. On the basis of the pseudo-semantic features, three one-against-one SVM classifiers are combined.

Some SVM packages, such as LIBSVM [LIBS] and SVM toolbox [Cawl00], can be downloaded for rapidly constructing your own applications. In our work, we employ the LIBSVM package developed by Chang and Lin [LIBS] for SVM training and testing. This tool adopts “grid-search” on C and γ using cross-validation. Pairs of (C, γ) are tried and the one with best cross-validation accuracy is picked.

Appendix C
Computational Media Aesthetics

C.1 Film Grammar

Since the movie or film was invented in the late 19th century, many techniques have been applied to filmmaking, and the audiovisual medium is capable to convey thoughts or to express information. Although various kinds of films have been produced in the last century, some filmmaking guidelines are widely adopted. Film grammar [Mona00], therefore, gradually takes shape after a long-time grope. It elucidates the relationship among audiovisual elements that are employed by filmmakers.

One of the most famous filmmaking techniques is Montage [Brau98]. It is an idea of film editing, deriving from the concept that there should be contrast between two different independent shots. The Russian director Sergei M. Eisenstein firstly applied this concept in the movie "The Battleship Potemkin (1925)". Since that, both the directors and the audience realize the influence of elaborate arrangement of video shots.

In recent years, the emergence of multimedia further triggers a new revolution in filmmaking. Herbert Zettl addresses the relationships between media elements and filmmaking [Zett99]. He defines "media aesthetics" as a study and analysis of media elements such as lighting, motion, color and sound by themselves and their roles in creating effective productions. This subject describes filmmaking from the viewpoints of aestheticians and directors. Dorai and Venkatesh [Dora02], on the other hand, propose a study named "computational media aesthetics" to algorithmically investigate filmmaking from audio/video elements in a computational manner. The following subsections will describe the idea of "computational media aesthetics" and some recent works.

C.2 Computational Media Aesthetics (CMA)

The definition of computational media aesthetics is [Dora02]:

> *The algorithmic study of a number of image and aural elements in media and the computational analysis of the principles that have emerged underlying their use*

and manipulation, individually or jointly, in the creative art of clarifying, intensifying, and interpreting some event for the audience.

On the basis of the idea of media aesthetics, Dorai and Venkatesh propose a computational framework to analyze videos and to facilitate high-level video/film abstraction. This framework exploits media production knowledge to elucidate the relationships between basic visual and aural elements, their intended meaning, and perceived impacts on content users.

The two-tiered framework is shown in Figure C-1. It consists of "primitive feature extraction" and "semantic construct extraction". In the stage of primitive feature extraction, audio and/or video features are directly extracted from signals. Attributes of audio/video elements that are described in Zettl's work are estimated in a computational manner.

The semantic construct extraction stage sets this framework apart from other schemes. The key difference is that this framework analyzes content based on production knowledge or the so-called film grammar in filmmaking. The production knowledge both defines what and how to extract the aesthetic elements that constructs semantics. The examples of semantic construct shown in Figure C-1 are tone, tempo, and rhythm. Taking tempo as the example, it is determined by the combination of average shot length, motion, and sound energy. The interrelationship between tempo and the related primitive features are inspired from the film grammar.

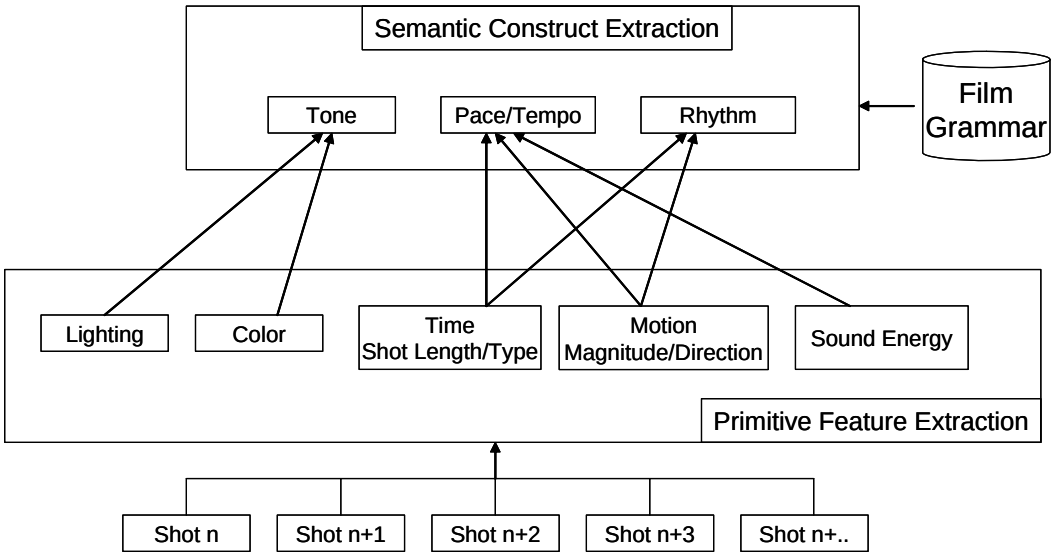


Figure C-1. Computational media aesthetics framework [Dora02].

C.3 Examples of CMA Applications

In this subsection, we describe some recent works based on computation media aesthetics.

C.3.1 Formulating Film Tempo [Dora02]

Adams et al. formalize the concept of tempo as being a function of the "information delivery rate" thrust at the viewer. Two dominant factors about tempo are motion (both object motion and camera motion) and shot rate (shot length). Therefore, the authors extract these two components for constructing tempo. This step corresponds to the stage of "primitive feature extraction" in Figure C-1.

The tempo function is defines as a linear combination of shot length and motion magnitude factors.

$$T(n) = \alpha\left(W\left(s(n)\right)\right) + \frac{\beta\left(m(n) - \mu_m\right)}{\sigma_m}, \tag{C-1}$$

where $s(n)$ denotes the frame numbers of the nth shot, $m(n)$ denotes the motion magnitude of the nth shot. The mean and standard deviation of motion magnitude are denotes as μ_m and σ_m , respectively. $W(.)$ is the weighting function used in shot length normalization. The weights α and β can be adjusted for different applications or different videos.

Finally, they generate a tempo curve for a movie. The region with sharp slope means that some important events occur or story changes.

C.3.2 Horror Film Genre Typing and Scene Labeling via Audio Analysis [Monc03]

Moncrieff et al. analyze audio tracks of films and find specific sound patterns that result from the changes in sound energy intensity over time. They focus on the detection of scenes with a high degree of horror thematic content, rather than scenes that contain a single brief scary shot. Through sound energy analysis, four types of sound energy events are studied: (1) surprise or alarm; (2) apprehension, or the emphasis of an event; (3) surprise followed by sustained alarm; and (4) apprehension building up to a climax.

C.3.3 Pivot Vector Space Approach for Audio-Video Mixing [Mulh03]

The goal of this work is to find appropriate music segments (from a music database) to help mixer dub in background music, by giving a video clip without sounds.

Several features are extracted from video and music segments. These features are categorized as dynamics (light, color energy, color brightness for video and dynamics for music), motion (motion vectors of video and tempo of music), and pitch (color hue of video and pitch of music). According to the values of features, they are classified as low, medium, or high level. For example, they can say that a clip is low-dynamics, medium motion, and high pitch. In a word, the system finds the music segments that have similar dynamics, motion, and pitch levels to the given video clip.

C.4 Semantic Indexing vs. CMA

For multimedia content analysis, we believe that many studies have the same ultimate goal, while the ideas or methodologies are different. The content described in this dissertation can be roughly categorized as semantic indexing. In this section, we briefly compare the works on semantic indexing and CMA.

Both semantic indexing and CMA have the same purpose for systematic studies on multimedia content analysis. No matter how they process or model this task, they have to start from extracting features from audio/video signals. That's the indispensable process for computational framework. In addition to this common process, the works of semantic indexing and CMA have many differences. Table C-1 lists the characteristics of them from different perspectives.

Table C-1. Comparison between semantic indexing and CMA.

	Semantic indexing	CMA
Purpose	Identify "what it is" or "what things /objects exist in a video segment".	Identify "how this video segment affects our emotion" or "what circumstance the director wants to present to viewers".
Approach	Complete and theoretically attractive probabilistic frameworks, such as HMM and SVM, afford modeling and training.	Up to now, only simple approaches, such as linear combination of feature vectors, have been proposed.
Prior knowledge	The phenomena of co-occurrence of relevant objects/events. The spatial /temporal relationships between relevant objects.	Film grammar and production rules in different video applications.
Flexibility	The same framework could be applied to various video applications by just extracting different features and detecting different events.	According to different production rules, the kernel function such as linear combination should be changed.

References

[Adai02] Adair, R.K., "The physics of baseball," Harper Collins, New York, 2002.

[ASQA06] ASQA, Academia Sinica Question Answering System, http://asqa.iis.sinica.edu.tw/clqa/

[Arij91] Arijon, D., "Grammar of the film language," Sliman-James Press, 1991.

[Arik03] Ariki, Y., Kumano, M., and Tsukada, K., "Highlight scene extraction in real time from baseball live video," Proceedings of ACM International Workshop on Multimedia Information Retrieval, pp. 209-214, 2003.

[Bach96] Bach, J., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R., and Shu, C., "The virage image search engine: an open framework for image management," Proceedings of SPIE Storage and Retrieval for Image and Video Databases, pp. 76-87, 1996.

[Bach05] Bach, N.H., Shinoda, K., and Furui, S., "Robust highlight extraction using multi-stream hidden Markov models for baseball video," Proceedings of IEEE International Conference on Image Processing, vol. 3, pp. 173-176, 2005.

[Baba04] Babaguchi, N., Kawai, Y., Ogura, T., and Kitahashi, T., "Personalized abstraction of broadcasted American football video by highlight selection," IEEE Transactions on Multimedia, vol. 6, no. 4, 2004, pp. 575-586.

[Bahi05] Bahill, A.T., Baldwin, D.G., and Venkateswaren, J., "Predicting a baseball's path," American Scientist, vol. 93, no. 3, pp. 218-225, 2005.

[Bart05] Bartsch, M.A., and Wakefield, G.H., "Audio thumbnailing of popular music using chroma-based representations," IEEE Transactions on Multimedia, vol. 7, no. 1, pp. 96-104, 2005.

[Beni05] Benitez, A.B., "Multimedia knowledge: discovery, classification, browsing, and retrieval," PhD Thesis Graduate School of Arts and Sciences, Columbia University, 2005.

[Bert05] Bertini, M., Del Bimbo, A., and Nunziati, W., "Highlights modeling and detection in sports videos," Pattern Analysis and Applications, 2005.

[Bow02] Bow, S.T., "Pattern Recognition and Image Preprocessing," Marcel Dekker, 2002.

[Brau98] Braudy, L., and Cohen, M., “Film theory and criticism: introductory readings,” Oxford University Press, 1998.

[Burn03] Burnett, I, Van de Walle, R., Hill, K., Bormans, J., and Pereira, F., “MPEG-21: goals and achievements,” IEEE Multimedia, Oct.-Dec., pp. 60-70, 2003.

[Cai03] Cai, R., Lu, L., Zhang, H.-J., Cai, L.H., “Highlight sound effects detection in audio stream,” Proceedings of the IEEE International Conference on Multimedia and Expo, vol. 3, pp. 37-40, 2003.

[Cawl00] Cawley, G.C., Support vector machine toolbox, http://theoval.sys.uea.ac.uk/svm/toolbox/, 2000.

[Chan98] Chang, S.-F., Chen, W., and Sundaram, H., “Semantic visual templates – linking features to semantics,” Proceedings of IEEE International Conference on Image Processing, vol. 3, pp. 531-535, 1998.

[Chan01] Chang, S.-F., Sikora, T., Purl, A., “Overview of MPEG-7 standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 6, pp. 688-695, 2001.

[Chan05] Chang, S.-F., and Vetro, A., “Video adaptation: concepts, technologies, and open issues,” Proceedings of the IEEE, vol. 94, no. 1, pp. 148-158, 2005.

[Chen98] Chen, B., Wang, H.-W., Chien, L.-F., and Lee, L.-S., “A*-admissible key-phrase spotting with sub-syllable level utterance verification,” Proceedings of IEEE International Conference on Spoken Language Processing, 1998.

[Chen03] Cheng, W.-H., Chu, W.-T., and Wu, J.-L., “Semantic Context Detection based on Hierarchical Audio Models,” Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval, pp. 109-115, 2003.

[Chen04] Chen, H.-W., Kuo, J.-H., Chu, W.-T., and Wu, J.-L., “Action Movies Segmentation and Summarization Based on Tempo Analysis,” Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval, pp. 251-258, 2004.

[Chu04] Chu, W.-T., Cheng, W.-H., Wu, J.-L., and Hsu, Y.-J., “A Study of Semantic Context Detection by Using SVM and GMM Approaches,” Proceedings of the IEEE International Conference on Multimedia & Expo, vol. 3, pp. 1591-1594, 2004.

[Chu05-1] Chu, W.-T., and Wu, J.-L., “Explicit Semantic Events Detection and Development of Realistic Applications for Broadcasting Baseball

Videos," submitted to Multimedia Tools and Applications, 2005.

[Chu05-2] Chu, W.-T., Cheng, W.-H., and Wu, J.-L., "Semantic Context Detection Using Audio Event Fusion," to appear in the EURASIP Journal on Applied Signal Processing, 2005.

[Chu05-3] Chu, W.-T., Cheng, W.-H., Hsu, J. Y.-J., and Wu, J.-L., "Towards Semantic Indexing and Retrieval Using Hierarchical Audio Models," ACM Multimedia Systems Journal, vol. 10, no. 6, pp. 570-583, 2005.

[Chu05-4] Chu, W.-T., and Wu, J.-L., "Integration of Rule-based and Model-based Methods for Baseball Event Detection," Proceedings of IEEE International Conference on Multimedia & Expo, pp. 137-140, 2005.

[Chu05-5] Chu, W.-T., Cheng, W.-H., and Wu, J.-L., "Generative and Discriminative Modeling toward Semantic Context Detection in Audio Tracks," Proceedings of the 11th International Multimedia Modelling Conference, pp. 38-45, 2005.

[Chu05-6] Chu, W.-T., and Wu, J.-L., "Detection of Spirited Incidental Music in Movies," Proceedings of Workshop on Computer Music and Audio Technology, 2005.

[Chu06-1] Chu, W.-T., and Wu, J.-L., "Development of Realistic Applications Based on Explicit Event Detection in Broadcasting Baseball Videos," Proceedings of International Multimedia Modeling Conference, pp.12-19, 2006.

[Chu06-2] Chu, W.-T., Wang, C.-W., and Wu, J.-L. "Extraction of baseball trajectory and physics-based validation for single-view baseball video sequences," accepted by IEEE International Conference on Multimedia & Expo, 2006.

[CMU06] The CMU Pronouncing Dictionary, http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[Cove67] Cover, T.M., and Hart, P.E., "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, pp. 21-27, 1967.

[Cowi01] Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J.G., "Emotion recognition in human-computer interaction," IEEE Signal Processing Magazine, Jan., 2001, pp. 32-80.

[CPBL06] Chinese Professional Baseball League, http://www.cpbl.com.tw

[Day05] Day, M.-Y., Lee, C.-W., Wu, S.-H., Ong, C.-S., Hsu, W.-L., "An integrated knowledge-based and machine learning approach for chinese question classification," Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering, pp. 620-625, 2005.

[Dimi02] Dimitrova, N., Zhang, H.-J., Shahraray, B., Huang, T.S., and Zakhor, A., "Applications of video-content analysis and retrieval," IEEE Multimedia, vol. 3, pp. 42-55, 2002.

[Dora02] Dorai, C., and Venkatesh, S., "Media computing: computation media aesthetics," Kluwer Academic Publisher, 2002.

[Duan03] Duan, L.-Y., Xu, M., Chua, T.-S., Tian, Q., and Xu, C.-S. "A mid-level representation framework for semantic sports video analysis," Proceedings of ACM Multimedia Conference, pp. 33-44, 2003.

[Duda01] Duda, R.O., Hart, P.E., Stork, D.G., "Pattern Classification," John Wiley & Sons, 2001.

[Ekin03] Ekin, A., Tekalp, A.M., and Mehrota, R., "Automatic soccer video analysis and summarization," IEEE Transactions on Image Processing, vol. 12, no. 7, 2003, pp. 796-807.

[Fisc95] Fischer, S., Lienhart, R., and Effelsberg, W., "Automatic recognition of film genres," Proceedings of ACM Multimedia, pp. 295-304, 1995.

[Flic95] Flickner, M., Petkovic, D., Steele, D., Yanker, P., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., and Lee, D., "Query by image and video content: the QBIC system," IEEE Computer, vol. 28, no. 9, pp. 23-32, 1995.

[From97] Fromkin, V., and Rodman, R., "An introduction to language," Harcourt Brace, 6th edition, 1997.

[Ghah06] Ghahramani, Z., Software written in Matlab, http://www.gatsby.ucl.ac.uk/~zoubin/software.html

[GHMM] General Hidden Markov Model Library (GHMM), http://www.ghmm.org

[Guez02] Guezic, A., "Tracking pitches for broadcast television," IEEE Computer, vol. 35, no. 3, pp. 38-43, 2002.

[Haer00] Haering, N., Qian, R.J., and Sezan, M.I., "A semantic event-detection approach and its application to detecting hunts in wildlife video," IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, no. 6, 2000.

[Han02] Han, M., Hua, W., Xu, W., and Gong, Y., "An integrated baseball digest system using maximum entropy method," Proceedings of ACM Multimedia Conference, 2002, pp. 347-350.

[Hanj02] Hanjalic, A., "Shot-boundary detection: unraveled and resolved?" IEEE Transactions on Circuits System and Video Technology, vol. 2, pp. 90-105, 2002.

[Hsu02] Hsu, C.-W. and Lin, C.-J., "A comparison of methods for multiclass

support vector machines," IEEE Transactions on Neural Networks, vol. 13, no. 2, pp. 415-425, 2002.

[Hsu06] Hsu, C.-W., Chang, C.-C., and Lin, C.-J., "A practical guide to support vector machine," http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf, 2006.

[HTK] Hidden Markov Model Toolkit (HTK), http://htk.eng.cam.ac.uk/

[Hua02] Hua, W., Han, W., and Gong, Y., "Baseball scene classification using multimedia features," Proceedings of IEEE International Conference on Multimedia & Expo, 2002, pp. 821-824.

[Huan01] Huang, X., Acero, A., and Hon, H.-W., "Spoken language processing: a guide to theory, algorithm, and system development," Prentice Hall, 2001.

[Hyva01] Hyvarinen, A., Karhunen, J., and Oja, E., "Independent Component Analysis," John Wiley & Sons, 2001.

[Jain00] Jain, A.K., Duin, R.P.W., and Mao, J., "Statistical pattern recognition: a review," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4-37, 2000.

[Khot90] Khotanzad, A., and Hong, Y.-H., "Invariant image recognition by Zernike moments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 5, pp. 489-497, 1990.

[Kitt98] Kitter, J., Hatef, M., Duin, R.D.W., and Matas, J., "On combining classifiers," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226-239, 1998.

[Lay06] Lay, J.A., and Guan, L., "Semantic retrieval of multimedia by concept language," IEEE Signal Processing Magazine, vol. 23, no. 2, pp. 115-123, 2006.

[Lee03] Lee, H.Y., Lee, H.K., and Ha, Y.H., "Spatial color descriptor for image retrieval and video segmentation," IEEE Transactions on Multimedia, vol. 5, no. 3, pp. 358-367, 2003.

[Leon04] Leonardi, R., Migliorati, P., and Prandini, M., "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains," IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 5, 2004, pp. 634-643.

[Li00] Li, S.Z., "Content-based classification and retrieval of audio using the nearest feature line method," IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 619-625, 2000.

[Li01] Li, Y., Zhong, T., and Tretter, D., "An overview of video abstraction techniques," Technical Report, HPL-2001-191, Hewlett-Packard

Company, 2001.

[Li04] Li, B., Errico, J.H., Pan, H., and Sezan, I., "Bridging the semantic gap in sports video retrieval and summarization," Journal of Visual Communication and Image Representation, vol. 15, 2004, pp. 393-424.

[Lian04] Liang, C.-H., Kuo, J.-H., Chu, W.-T., and Wu, J.-L., "Semantic Units Detection and Summarization of Baseball Videos," Proceedings of the IEEE International Midwest Symposium on Circuits and Systems, vol. 1, pp. 297-300, 2004.

[Lian05] Liang, C.-H., Chu, W.-T., Kuo, J.-H., Wu, J.-L., and Cheng, W.-H., "Baseball Event Detection Using Game-Specific Feature Sets and Rules," Proceedings of IEEE International Symposium on Circuits and Systems, pp. 3829-3832, 2005.

[LIBS] LIBSVM – A library for support vector machine, http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html, 2001.

[Lien99] Lienhart, R., "Comparison of automatic shot boundary detection algorithms," Proceedings of SPIE Storage and Retrieval for Still Image and Video Databases VII, vol. 3656, pp. 290-301, 1999.

[Lin03] Lin, W.-H., Hauptmann, A., "Meta-classification: Combining multimodal classifiers," Zaiane, O.R., Simoff, S., Djeraba, C. (eds.) Mining Multimedia and Complex Data, Springer, Berlin Heidelberg New York, pp. 217-231, 2003.

[Liu98] Liu, Z., Huang, J., and Wang, Y., "Classification of TV programs based on audio information using hidden Markov model," Proceedings of the IEEE Signal Processing Society Workshop on Multimedia Signal Processing, pp. 27-32, 1998.

[Lu02] Lu, L., Zhang, H.-J., and Jiang, H., "Content analysis for audio classification and segmentation," IEEE Transactions on Speech and Audio Processing, vol. 7, pp. 504-516, 2002.

[Lu03] Lu, L., and Zhang, H.-J., "Automatic extraction of music snippets," Proceedings of the ACM Multimedia Conference, pp. 140-147, 2003.

[Madd06] Maddage, N.C., "Automatic structure detection for popular music," IEEE Multimedia, vol. 13, no. 1, pp. 65-77, 2006.

[Mart02-1] Martinez, J.M., Koenen, R., and Pereira, F., "MPEG-7: the generic multimedia content description standard, part 1," IEEE Multimedia, vol. 9, no. 2, pp. 78-87, 2002.

[Mart02-2] Martinez, J.M. "Standards - MPEG-7 overview of MPEG-7 description tools, part 2," IEEE Multimedia, vol. 9, no. 3, pp. 83-93, 2002.

[MLB06] Major League Baseball, http://www.mlb.com

[Moha99] Mohan, R., Smith, J.R., and Li, C.-S., “Adapting multimedia content for universal access,” IEEE Transactions on Multimedia, vol. 1, no. 1, pp. 104-114, 1999.

[Mona00] Monaco, J., “How to read a film: the world of movies, media, and multimedia: language, history, theory” Oxford University Press, 2000.

[Monc03] Moncrieff, S., Venkatesh, S., and Dorai, C., “Horror film genre typing and scene labeling via audio analysis,” Proceedings of the IEEE International Conference on Multimedia and Expo, vol. 2, pp. 193-196, 2003.

[Mulh03] Mulhem, P., Kankanhalli, M.S., Yi, J., and Hassan, H., “Pivot vector space approach for audio-video mixing,” IEEE Multimedia, vol. 10, no. 2, pp. 28-40, 2003.

[Murp05] Murphy, K., Hidden Markov model (HMM) Toolbox for Matlab, http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[Naph98] Naphade, M.R., Kristjansson, T., Frey, B., Huang, T.S., “Probabilistic multimedia objects (multijects): a novel approach to video indexing and retrieval in multimedia system,” Proceedings of the IEEE International Conference on Image Processing, vol. 3, pp. 536-540, 1998.

[Naph01] Naphade, M.R., “A probabilistic framework for mapping audio-visual features to high-level semantics in terms of concepts and context,” PhD dissertation, University of Illinois at Urbana-Champaign, 2001.

[Naph02] Naphade, M.R., and Huang, T.S., “Extracting semantics from audiovisual content: the final frontier in multimedia retrieval,” IEEE Transactions on Neural Network, vol. 13, no. 4, pp. 793-810, 2002.

[Nepa01] Nepal, S., Srinivasan, U., and Reynolds, G., “Automatic detection of goal segments in basketball videos,” Proceedings of ACM Multimedia Conference, pp. 261-269, 2001.

[Ngo05] Ngo, C.-W., Ma, Y.-F., and Zhang, H.-J., “Video summarization and scene detection by graph modeling,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 2, pp. 296-305, 2005.

[Pfei96] Pfeiffer, S., Lienhart, R., Fischer, S., and Effelsberg, W., “Abstracting digital movies automatically,” Journal of Visual Communication and Image Representation, vol. 4, pp. 345-353, 1996.

[Plat00] Platt, J.C., Cristianini, N., and Shawe-Taylor, J., “Large margin DAGs for multiclass classification,” Advances in Neural Information Processing Systems, Cambridge, MA: MIT Press, vol. 12, pp. 547-553, 2000.

[Rabi89] Rabiner, L.R., “A tutorial on hidden Markov models and selected

applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, 1989.

[Rabi93] Rabiner, L., and Juang, B.-H., "Fundamentals of speech recognition," Prentice Hall, 1993.

[Reim06] Reimers, U.H. , "DVB-the family of international standards for digital video broadcasting," Proceedings of IEEE, vol. 94, no. 1, pp. 173-182, 2006.

[Rui98] Rui, Y., Huang, T.S., Ortega, M., and Mehrotra, S., "Relevance feedback: a power tool in interactive content-based image retrieval," IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 5, pp. 644-655, 1998.

[Rui99] Rui, Y., Huang, T.S., and Chang, S.-F., "Image retrieval: current techniques, promising directions and open issues," Journal of Visual Communication and Image Representation, vol. 10, no. 4, pp. 39-62, 1999.

[Rui00] Rui, Y., Gupta, A., and Acero, A., "Automatically extracting highlights for tv baseball programs," Proceedings of ACM Multimedia Conference, 2000, pp. 105-115.

[Seth03] Sethy, A., Narayanan, S., "Split-lexicon based hierarchical recognition of speech using syllable and word level acoustic units," Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 772-775, 2003.

[Shih03] Shih, H.-C., and Huang, C.-L., "A semantic network modeling for understanding baseball video," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 820-823, 2003.

[Smeu00] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., and Jain, R., "Content-based image retrieval at the end of early year," IEEE Transactions of Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1349-1380, 2000.

[Smit96] Smith, J.R., and Chang, S.-F., "Visualseek: a fully automated content-based image query system," Proceedings of ACM Multimedia, pp. 87-98, 1996.

[Smit03] Smith, J.R., Naphade, M., and Natsev, A., "Multimedia semantic indexing using model vectors," Proceedings of ICME, vol. 2, pp. 445-448, 2003.

[Snoe06] Snoek, C.G.M., Worring, M., and Smeulders, A.W.M., "Early versus late fusion in semantic video analysis," Proceedings of ACM

Multimedia, pp. 399-402, 2005.

[Soun06] SoundIdeas Sound Effects Library, http://www.sound-ideas.com/

[Stol97] Stolfo, S., Prodromidis, A., Tselepis, S., Lee, W., Fan, D., Chan, P., "JAM: Java agents for meta-learning over distributed databases," Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 74-81, 1997.

[Theo04] Theobalt, C., Albrecht, I., Haber, J., Magnor, M., Seidel, H.-P., "Pitching a baseball: tracking high-speed motion with multi-exposure images," Proceedings of ACM SIGGRAPH, pp. 540-547, 2004.

[Tjon04] Tjondronegoro, D., Chen, Y.-P. P., and Pham, B., "Integrating highlights for more complete sports video summarization," IEEE Multimedia, Oct.-Dec., 2004, pp. 22-37.

[TREC06] TREC Video Retrieval Evaluation, http://www-nlpir.nist.gov/projects/trecvid/

[Tsen04] Tseng, B.L., Lin, C.-Y., and Smith, J.R., "Using MPEG-7 and MPEG-21 for personalizing video," IEEE Multimedia, vol. 11, no. 1, pp. 42-52, 2004.

[Tzan02] Tzanetakis, G., and Cook, P., "Musical genre classification of audio signals," IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293-302, 2002.

[Vapn98] Vapnik, V.N., "Statistical Learning Theory," Wiley, New York, 1998.

[Vetr05] Vetro, A., and Trimmerer, C., "Digital item adaptation: overview of standardization and research activities," IEEE Transactions on Multimedia, vol. 7, no. 3, pp. 418-426, 2005.

[Vide06] Videoland Sports Channel, http://sport.videoland.com.tw/

[Wact96] Wactlar, H., Kanade, T., Smith, M, and Stevens, S., "Intelligent access to digital video: the Informedia project," IEEE Computer, vol. 29, no. 5, pp. 46-52, 1996.

[Wang00] Wang, Y., Liu, Z., Huang, J.C., "Multimedia content analysis using both audio and visual cues," IEEE Signal Processing Magazine, vol. 17, no. 6, pp. 12-36, 2000.

[Wang04-1] Wang, L., Lew, M., and Xu, G., "Offense based temporal segmentation for event detection in soccer video," Proceedings of ACM International Workshop on Multimedia Information Retrieval, pp. 259-266, 2004.

[Wang04-2] Wang, J., Xu, C., Chng, E., Wan, K., and Tian, Q., "Automatic replay generation for soccer video broadcasting," Proceedings of ACM Multimedia Conference, pp. 32-39, 2004.

[Wang04-3] Wang, J., Xu, C., Chng, E., and Tian, Q., "Sports highlight detection

from keyword Sequences using HMM," Proceedings of IEEE International Conference on Multimedia and Expo, pp. 599-602, 2004.

[Welc04] Welch, G., Bishop, G., "An introduction to the Kalman filter," Technical Report no. TR 95-041, University of North Carolina at Capel Hill, 2004.

[Well05] Welling, M. "Support vector machines," Lecture notes in http://www.ics.uci.edu/~welling/teaching/KernelsICS273B/Kernels.html, 2005.

[Xie04] Xie, L., Xu, P., Chang, S.-F., Divakaran, A., and Sun, H., "Structure analysis of soccer video with domain knowledge and hidden Markov models," Pattern Recognition Letters, vol. 25, no. 7, 2004, pp. 767-775.

[Xion03] Xiong, Z., Radhakrishnan, R., Divakaran, A., and Huang, T.S., "Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 632-635, 2003.

[Xu03] Xu, G., Ma, Y.-F., Zhang, H.-J., and Yang, S., "An HMM based semantic analysis framework for sports game event detection," Proceedings of IEEE International Conference on Image Processing, vol. 1, 2003, pp. 25-28.

[Xu04] Xu, H., and Chua, T.-S., "The fusion of audio-visual features and external knowledge for event detection in team sports video," Proceedings of ACM International Workshop on Multimedia Information Retrieval, 2004, pp. 127-134.

[Yeo95] Yeo, B.L., and Liu, B., "Rapid scene change detection on compressed video," IEEE Transactions on Circuits System and Video Technology, vol. 6, pp. 533-544, 1995.

[Yu03] Yu, X., Xu, C., Leong, H.W., Tian, Q., Tang, Q., and Wan, K.W., "Trajectory-based ball detection and tracking with applications to semantic analysis of broadcasting soccer video," Proceedings of ACM Multimedia Conference, 2003, pp. 11-20.

[Yu05] Yu, X., and Farin, D., "Current and emerging topics in sports video processing," Proceedings of IEEE International Conference on Multimedia & Expo, pp. 526-529, 2005.

[Zett99] Zettl, H., "Sight sound motion: applied media aesthetics," Belmont, CA: Wadsworth Pbulishing, 1999.

[Zhan95] Zhang, H.-J., Low, C.Y., Smoliar, S.W., and Wu, J.H., "Video parsing, retrieval and browsing: an integrated and content-based solution," Proceedings of ACM Multimedia, pp. 15-24, 1995.

[Zhan98] Zhang, T. and Kuo, C.-C. J., “Hierarchical system for content-based audio classification and retrieval,” Proceedings of SPIE Multimedia Storage Archive and System, vol. 3, no. 3572, pp. 398-409, 1998.

[Zhan02] Zhang, D., and Chang, S.-F., “Event detection in baseball video using superimposed caption information,” Proceedings of ACM Multimedia Conference, 2002, pp. 315-318.

[Zhon04] Zhong, D., and Chang, S.-F., “Real-time view recognition and event detection for sports video,” Journal of Visual Communication and Image Representation, vol. 15, 2004, pp. 330-347.

[Zilc01] Zilca, R.D., “Text-independent speaker verification using covariance modeling,” IEEE Signal Processing Letters, vol. 8, no. 4, pp. 97-99, 2001.

Curriculum Vitae

朱威達 (Wei-Ta Chu)

wtchu@cmlab.csie.ntu.edu.tw
Communications and Multimedia Laboratory,
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan

Education

Ph.D. (2006) Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan

M.S.(2002) Department of Computer Science and Information Engineering
National Chi Nan University, Nantou, Taiwan

B.S. (2000) Department of Computer Science and Information Engineering
National Chi Nan University, Nantou, Taiwan

Publications

- **Journal Papers**

1. **W.-T. Chu**, C.-W. Wang, and J.-L. Wu, "Ball Trajectory Extraction and Physics-Based Validation in Sports Videos," submitted to *IEEE Transactions on Circuits and Systems for Video Technology*, 2006.
2. **W.-T. Chu**, J.-Y. Hu, C.-F. Chou, and J.-L. Wu, "On the Cooperation of Content Analysis Techniques and Multimedia Communications," submitted to *IEEE Communications Magazine*, 2006.
3. **W.-T. Chu** and J.-L. Wu, "Explicit Semantic Events Detection and Development of Realistic Applications for Broadcasting Baseball Videos," revised for *Multimedia Tools and Applications*, 2006.
4. **W.-T. Chu**, W.-H. Cheng, and J.-L. Wu, "Semantic Context Detection Using Audio Event Fusion," to appear in *EURASIP Journal on Applied Signal Processing*, 2006.

5. **W.-T. Chu**, W.-H. Cheng, J. Y.-J. Hsu, and J.-L. Wu, "Towards Semantic Indexing and Retrieval Using Hierarchical Audio Models," *ACM Multimedia Systems Journal, vol. 10, no. 6,* pp. 570-583, 2005.
6. W.-H. Cheng, **W.-T. Chu**, and J.-L. Wu, "A Visual Attention based Region-of-Interest Determination Framework for Video Sequences," *IEICE Transactions on Information and Systems Journal, vol. E-88D, no. 7,* pp. 1578-1586, 2005.
7. **W.-T. Chu**, W.-H. Cheng, S.-F. He, C.-W. Wang, and J.-L. Wu, "A Unified Framework Using Spatial Color Descriptor and Motion-based Post Refinement for Shot Boundary Detection," *GESTS International Transaction on Computer Science and Engineering, vol. 2, no. 1,* pp. 133-143. 2005.
8. **W.-T. Chu**, and H.-Y. Chen, "Towards better Retrieval and Presentation by Exploring Cross-Media Correlations," *ACM Multimedia Systems Journal, vol. 10, no. 3,* pp. 183-198, 2005.

- **International Conferences**

1. **W.-T. Chu**, C.-W. Wang, and J.-L. Wu, "Extraction of Baseball Trajectory and Physics-Based Validation for Single-View Baseball Video Sequences," accepted by *IEEE International Conference on Multimedia & Expo*, 2006.
2. **W.-T. Chu** and J.-L. Wu, "Development of Realistic Applications Based on Explicit Event Detection in Broadcasting Baseball Videos," *Proceedings of International Multimedia Modelling Conference*, pp. 12-19, 2006.
3. Y.-H. Chen, J.-H. Kuo, **W.-T. Chu**, and J.-L. Wu, "Movie Emotional Event Detection Based on Music Mood and Video Tempo," *Proceedings of IEEE International Conference on Consumer Electronics*, pp. 151-152, 2006.
4. J.-C. Chen, J.-H. Yeh, **W.-T. Chu**, J.-H. Kuo, and J.-L. Wu, "Improvement of Commerical Boundary Detection Using Audiovisual Features," *The 6th Pacific-Rim Conference on Multimedia*, 2005. (*Lecture Notes in Computer Science, vol. 3767,* pp. 776-786, 2005)
5. **W.-T. Chu** and J.-L. Wu, "Integration of Rule-based and Model-based Methods for Baseball Event Detection," *Proceedings of IEEE International Conference on Multimedia & Expo*, pp. 137-140, 2005.
6. C.-H. Liang, **W.-T. Chu**, J.-H. Kuo, J.-L. Wu, and Wen-Huang Cheng, "Baseball Event Detection Using Game-Specific Feature Sets and Rules," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 3829-3832, 2005.

7. W.-H. Cheng, **W.-T. Chu**, J.-H. Kuo, and J.-L. Wu, "Automatic Video Region-of-Interest Determination Based on User Attention Model," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 3219-3222, 2005.

8. **W.-T. Chu**, W.-H. Cheng, and J.-L. Wu, "Generative and Discriminative Modeling toward Semantic Context Detection in Audio Tracks," *Proceedings of the 11th International Multimedia Modelling Conference*, pp. 38-45, 2005.

9. **W.-T. Chu**, W.-H. Cheng, S.-F. He, C.-W. Wang, and J.-L. Wu, "A Unified Framework Using Spatial Color Descriptor and Motion-based Post Refinement for Shot Boundary Detection," *The 5th Pacific-Rim Conference on Multimedia*, 2004. (*Lecture Notes in Computer Science, vol. 3333,* pp. 558-565, 2004.)

10. H.-W. Chen, J.-H. Kuo, **W.-T. Chu**, and J.-L. Wu, "Action Movies Segmentation and Summarization Based on Tempo Analysis," *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 251-258, 2004.

11. C.-H. Liang, J.-H. Kuo, **W.-T. Chu**, and J.-L. Wu, "Semantic Units Detection and Summarization of Baseball Videos," *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems, vol. 1*, pp. 297-300, 2004.

12. **W.-T. Chu**, W.-H. Cheng, J.-L. Wu, and Y.-J. Hsu, "A Study of Semantic Context Detection by Using SVM and GMM Approaches," *Proceedings of the IEEE International Conference on Multimedia & Expo, vol. 3,* pp. 1591-1594, 2004.

13. W.-H. Cheng, **W.-T. Chu**, and J.-L. Wu, "Semantic Context Detection based on Hierarchical Audio Models," *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 109-115, 2003.

14. C.-C. Ho, **W.-T. Chu**, C.-H. Huang, and J.-L. Wu, "User-Oriented Approach in Spatial and Temporal Domain Video Coding," *Proceedings of the 4th IEEE Pacific-Rim Conference on Multimedia*, 2003.

15. **W.-T. Chu**, and H.-Y. Chen, "Multiple Granularity Access to Navigated Hypermedia Documents Using Temporal Meta-Information," *The 3rd IEEE Pacific-Rim Conference on Multimedia*, 2002. (*Lecture Notes in Computer Science, vol. 2532,* pp. 1227-1234, 2002.)

16. **W.-T. Chu**, and H.-Y. Chen, "Cross-Media Correlation: A Case Study for Navigated Hypermedia Documents," *Proceedings of the ACM Multimedia Conference*, pp. 57-66, 2002.

17. **W.-T. Chu**, K.-T. Hsu, and H.-Y. Chen, "Design of an Alignment System for Synchronized Speech-Text Presentation," *Proceedings of Distributed Multimedia Systems*, pp. 86-93, 2001.

18. 陳恆佑, 宋如瑜, **朱威達**, 吳獻良, "多媒體教學系統在華語文課程上的應用," *5th Global Chinese Conference on Computers in Education*, pp. 1060-1066, 2001.

- **Local Conferences**

1. **W.-T. Chu** and J.-L. Wu, "Explicit Baseball Event Detection by Combining Visual and Speech Information," submitted to *19th Computer Vision, Graphics, and Image Processing Conference*, 2006.
2. **W.-T. Chu** and J.-L. Wu, "Detection of Spirited Incidental Music in Movies," *Proceedings of Workshop on Computer Music and Audio Technology*, 2005.
3. W.-H. Cheng, **W.-T. Chu**, and J.-L. Wu, "A Visual Focus Detection Framework for Video Sequences," *Proceedings of Workshop on Consumer Electronics and Signal Processing*, 2004. **(Best student paper award)**
4. K.-Y. Liu, H.-L. Wu, M.-W. Lai, **W.-T. Chu**, B.-H. Wu, and H.-Y. Chen, "Exploring Interactive Multimedia Technologies for Web-based ESL Learning," *Proceedings of the Sixth International Conference on Multimedia Language Education, ROCMELIA*, pp. 181-191, 2002.
5. 賴茂濰, **朱威達**, 陳恆佑, "網路式英語聽力訓練系統," *民生電子研討會*, 2001.

- **Demonstration**

1. J.-C. Chen, **W.-T. Chu**, J.-H. Kuo, C.-Y. Weng, and J.-L. Wu, "Audiovisual slideshow: Present Your Journey by Photos," submitted to *ACM Multimedia Conference*, 2006.
2. K.-Y. Liu, N. Huang, B.-H. Wu, **W.-T. Chu**, and H.-Y. Chen, "The WSML System: Web-based Synchronization Multimedia Lecture System," *Proceedings of the ACM Multimedia Conference*, pp. 662-663, 2002.

- **Others**

1. J.-L. Wu and **W.-T. Chu**, "Audio and Video Semantic Analyses and Its Applications," *International Workshop on Task-Relevant Ubiquitous Media Access*, 2005.
2. Y.-S. Tung, **W.-T. Chu**, W.-H. Cheng, T.-J. Pan, H.-S. Chen, and J.-L. Wu, "FGS Anchor Sequences for Call for Evidence on Scalable Video Coding Advances and

Related Tools," *ISO/IEC JTC1/SC29/WG11 MPEG2002/M9821*, July 2003.

3. 陳恆佑, **朱威達**, 吳獻良, 洪政欣, "網路影音教材查詢及同步點播," *隔空教育論叢年刊第十二輯*, pp. 109-120, 2000.